

Ke Chen
Carola-Bibiane Schönlieb
Xue-Cheng Tai
Laurent Younes
Editors

Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging

Mathematical Imaging and Vision

Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging

Ke Chen • Carola-Bibiane Schönlieb •
Xue-Cheng Tai • Laurent Younes
Editors


Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging


Mathematical Imaging and Vision


With 553 Figures and 72 Tables

 Springer

Editors

Ke Chen 
Department of Mathematical Sciences
The University of Liverpool
Liverpool, UK

Xue-Cheng Tai 
Hong Kong Center for
Cerebrocardiovascular Health
Engineering (COCHE)
Shatin, Hong Kong, China

Carola-Bibiane Schönlieb 
Department of Applied Mathematics and
Theoretical Physics
University of Cambridge
Cambridge, UK

Laurent Younes 
Department of Applied Mathematics and Statistics
Johns Hopkins University
Baltimore, MD, USA

ISBN 978-3-030-98660-5

ISBN 978-3-030-98661-2 (eBook)

<https://doi.org/10.1007/978-3-030-98661-2>

© Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland.

Preface

The rapid development of new imaging hardware, the advance in medical imaging, the advent of multi-sensor data fusion and multimodal imaging, as well as the advances in computer vision have sparked numerous research endeavours leading to highly sophisticated and rigorous mathematical models and theories. Motivated by the increasing use of variational models, shapes and flows, differential geometry, optimisation theory, numerical analysis, statistical/Bayesian graphical models, machine learning, and deep learning, we have invited contributions from leading researchers and publish this handbook to review and capture the state of the art of research in Computer Vision and Imaging.

This constantly improving technology that generates new demands not readily met by existing mathematical concepts and algorithms provides a compelling justification for such a book to meet the ever-growing challenges in applications and to drive future development. As a consequence, new mathematical models have to be found, analysed and realised in practice. Knowing the precise state-of-the-art developments is key, and hence this book will serve the large community of mathematics, imaging, computer vision, computer sciences, statistics, and, in general, imaging and vision research. Our primary audience are

- Graduate students
- Researchers
- Imaging and vision practitioners
- Applied mathematicians
- Medical imagers
- Engineers
- Computer scientists

Viewing discrete images as data sampled from functional surfaces enables the use of advanced tools from calculus, functions and calculus of variations, and optimisation and provides the basis of high-resolution imaging through variational models. No other framework can provide the comparable accuracy and precision to imaging and vision.

Although our initial emphasis is on the variational methods, which represent the optimal solutions to class of imaging and vision problems, and on effective algorithms, which are necessary for the methods to be translated to practical use in various applications, the editors recognise that the range of effective and efficient methods for solving problems from computer vision and imaging go beyond variational methods and have enlarged our coverage to include mathematical models and algorithms. So, the book title reflects this viewpoint and a big vision for the reference book.

All chapters will have introductions so that the book is readily accessible to graduate students. We have divided the 53 chapters of this book into 3 sections, namely

- (a) Convex and Non-convex Large-Scale Optimisation in Imaging
- (b) Model- and Data-Driven Variational Imaging Approaches
- (c) Shape Spaces and Geometric Flows

to facilitate browsing the content list. However, such a division is artificial because, these days, research becomes increasingly intra-disciplinary as well as inter-disciplinary, and ideas from one topic often directly or indirectly inspire or transpire another topic. This is very exciting.

For newcomers to the field, the book provides a comprehensive and fast track introduction to the core research problems, to save time and get on with tackling new and emerging challenges, rather than running the risk of reproducing/comparing to some old works already done or reinventing same results. For researchers, exposure to the state of the art of research works leads to an overall view of the entire field so as to guide new research directions and avoid pitfalls in moving the field forward and looking into the next 25 years of imaging and information sciences.

The dreadful Covid-19 pandemic starting from 2020 has affected lives of everyone, of course including all researchers. We are still not out of the woods. The editors are very much grateful to the book authors who have endured much hardship during the last 3 years and overcome many difficulties to have completed their chapters on time. We are also indebted to many anonymous reviewers who provided valuable reviews and helpful criticism to improve presentations of our chapters.

The original gathering of all editors was in 2017 when the first three editors co-organised the prestigious Isaac Newton Institute programme titled “*Variational methods and effective algorithms for imaging and vision*” (<https://www.newton.ac.uk/event/vmv/>), partially supported by UK EPSRC GR/EP F005431 and Isaac Newton Institute for Mathematical Sciences. During the programme, Mr Jan Holland from Springer-Nature kindly suggested the idea of a book. We are grateful to his suggestion which sparked the editors’ fruitful collaboration in the last few

years. The large team of publishers who have offered immense help to us include Michael Hermann (Springer), Allan Cohen (Palgrave) and Salmanul Faris Nedum Palli (Springer). We thank them all.

Finally, we wish all readers a happy reading.

The editorial team:

Liverpool, UK
Cambridge, UK
Shatin, Hong Kong
Baltimore, USA
February 2023

Ke Chen (Lead)
Carola-Bibiane Schönlieb
Xue-Cheng Tai
Laurent Younes

Contents

Volume 1

Part I Convex and Non-convex Large-Scale Optimization in Imaging	1
1 Convex Non-convex Variational Models	3
Alessandro Lanza, Serena Morigi, Ivan W. Selesnick, and Fiorella Sgallari	
2 Subsampled First-Order Optimization Methods with Applications in Imaging	61
Stefania Bellavia, Tommaso Bianconcini, Nataša Krejić, and Benedetta Morini	
3 Bregman Methods for Large-Scale Optimization with Applications in Imaging	97
Martin Benning and Erlend Skaldehaug Riis	
4 Fast Iterative Algorithms for Blind Phase Retrieval: A Survey	139
Huibin Chang, Li Yang, and Stefano Marchesini	
5 Modular ADMM-Based Strategies for Optimized Compression, Restoration, and Distributed Representations of Visual Data	175
Yehuda Dar and Alfred M. Bruckstein	
6 Connecting Hamilton-Jacobi Partial Differential Equations with Maximum a Posteriori and Posterior Mean Estimators for Some Non-convex Priors	209
Jérôme Darbon, Gabriel P. Langlois, and Tingwei Meng	
7 Multi-modality Imaging with Structure-Promoting Regularizers	235
Matthias J. Ehrhardt	

8	Diffraction Tomography, Fourier Reconstruction, and Full Waveform Inversion	273
	Florian Faucher, Clemens Kirisits, Michael Quellmalz, Otmar Scherzer, and Eric Setterqvist	
9	Models for Multiplicative Noise Removal	313
	Xiangchu Feng and Xiaolong Zhu	
10	Recent Approaches to Metal Artifact Reduction in X-Ray CT Imaging	347
	Soomin Jeon and Chang-Ock Lee	
11	Domain Decomposition for Non-smooth (in Particular TV) Minimization	379
	Andreas Langer	
12	Fast Numerical Methods for Image Segmentation Models	427
	Noor Badshah	
13	On Variable Splitting and Augmented Lagrangian Method for Total Variation-Related Image Restoration Models	503
	Zhifang Liu, Yuping Duan, Chunlin Wu, and Xue-Cheng Tai	
14	Sparse Regularized CT Reconstruction: An Optimization Perspective	551
	Elena Morotti and Elena Loli Piccolomini	
15	Recent Approaches for Image Colorization	585
	Fabien Pierre and Jean-François Aujol	
16	Numerical Solution for Sparse PDE Constrained Optimization ...	623
	Xiaoliang Song and Bo Yu	
17	Game Theory and Its Applications in Imaging and Vision	677
	Anis Theljani, Abderrahmane Habbal, Moez Kallel, and Ke Chen	
18	First-Order Primal–Dual Methods for Nonsmooth Non-convex Optimization	707
	Tuomo Valkonen	

Volume 2

Part II	Model- and Data-Driven Variational Imaging Approaches	749
19	Learned Iterative Reconstruction	751
	Jonas Adler	

20	An Analysis of Generative Methods for Multiple Image Inpainting	773
	Coloma Ballester, Aurélie Bugeau, Samuel Hurault, Simone Parisotto, and Patricia Vitoria	
21	Analysis of Different Losses for Deep Learning Image Colorization	821
	Coloma Ballester, Hernan Carrillo, Michaël Clément, and Patricia Vitoria	
22	Influence of Color Spaces for Deep Learning Image Colorization	847
	Aurélie Bugeau, Rémi Giraud, and Lara Raad	
23	Variational Model-Based Deep Neural Networks for Image Reconstruction	879
	Yunmei Chen, Xiaojing Ye, and Qingchao Zhang	
24	Bilevel Optimization Methods in Imaging	909
	Juan Carlos De los Reyes and David Villacís	
25	Multi-parameter Approaches in Image Processing	943
	Markus Grasmair and Valeriya Naumova	
26	Generative Adversarial Networks for Robust Cryo-EM Image Denoising	969
	Hanlin Gu, Yin Xian, Ilona Christy Unarta, and Yuan Yao	
27	Variational Models and Their Combinations with Deep Learning in Medical Image Segmentation: A Survey	1001
	Luying Gui, Jun Ma and Xiaoping Yang	
28	Bidirectional Texture Function Modeling	1023
	Michal Haindl	
29	Regularization of Inverse Problems by Neural Networks	1065
	Markus Haltmeier and Linh Nguyen	
30	Shearlets: From Theory to Deep Learning	1095
	Gitta Kutyniok	
31	Learned Regularizers for Inverse Problems	1133
	Sebastian Lunz	
32	Filter Design for Image Decomposition and Applications to Forensics	1155
	Robin Richter, Duy H. Thai, Carsten Gottschlich, and Stephan F. Huckemann	

33 Deep Learning Methods for Limited Data Problems in X-Ray Tomography 1183
 Johannes Schwab

34 MRI Bias Field Estimation and Tissue Segmentation Using Multiplicative Intrinsic Component Optimization and Its Extensions 1203
 Samad Wali, Chunming Li, and Lingyan Zhang

35 Data-Informed Regularization for Inverse and Imaging Problems 1235
 Jonathan Wittmer and Tan Bui-Thanh

36 Randomized Kaczmarz Method for Single Particle X-Ray Image Phase Retrieval 1273
 Yin Xian, Haiguang Liu, Xuecheng Tai, and Yang Wang

37 A Survey on Deep Learning-Based Diffeomorphic Mapping 1289
 Huilin Yang, Junyan Lyu, Roger Tam, and Xiaoying Tang

Volume 3

Part III Shape Spaces and Geometric Flows **1323**

38 Stochastic Shape Analysis 1325
 Alexis Arnaudon, Darryl Holm, and Stefan Sommer

39 Intrinsic Riemannian Metrics on Spaces of Curves: Theory and Computation 1349
 Martin Bauer, Nicolas Charon, Eric Klassen, and Alice Le Brigant

40 An Overview of SaT Segmentation Methodology and Its Applications in Image Processing 1385
 Xiaohao Cai, Raymond Chan, and Tiejong Zeng

41 Recent Development of Medical Shape Analysis via Computational Quasi-conformal Geometry 1413
 Hei-Long Chan and Lok-Ming Lui

42 A Survey of Topology and Geometry-Constrained Segmentation Methods in Weakly Supervised Settings 1437
 Ke Chen, Noémie Debroux, and Carole Le Guyader

43	Recent Developments of Surface Parameterization Methods Using Quasi-conformal Geometry	1483
	Gary P. T. Choi and Lok Ming Lui	
44	Recent Geometric Flows in Multi-orientation Image Processing via a Cartan Connection	1525
	R. Duits, B. M. N. Smets, A. J. Wemmenhove, J. W. Portegies, and E. J. Bekkers	
45	PDE-Constrained Shape Optimization: Toward Product Shape Spaces and Stochastic Models	1585
	Caroline Geiersbach, Estefania Loayza-Romero, and Kathrin Welker	
46	Iterative Methods for Computing Eigenvectors of Nonlinear Operators	1631
	Guy Gilboa	
47	Optimal Transport for Generative Models	1659
	Xianfeng Gu, Na Lei, and Shing-Tung Yau	
48	Image Reconstruction in Dynamic Inverse Problems with Temporal Models	1707
	Andreas Hauptmann, Ozan Öktem, and Carola Schönlieb	
49	Computational Conformal Geometric Methods for Vision	1739
	Na Lei, Feng Luo, Shing-Tung Yau, and Xianfeng Gu	
50	From Optimal Transport to Discrepancy	1791
	Sebastian Neumayer and Gabriele Steidl	
51	Compensated Convex-Based Transforms for Image Processing and Shape Interrogation	1827
	Antonio Orlando, Elaine Crooks, and Kewei Zhang	
52	The Potts Model with Different Piecewise Constant Representations and Fast Algorithms: A Survey	1887
	Xuecheng Tai, Lingfeng Li, and Egil Bae	
53	Shape Spaces: From Geometry to Biological Plausibility	1929
	Nicolas Charon and Laurent Younes	
	Index	1959

About the Editors



Prof. Ke Chen, PhD received his BSc, MSc and PhD degrees in applied mathematics, respectively, from the Dalian University of Technology (China), the University of Manchester (UK) and the University of Plymouth (UK). He is a computational mathematician specialised in developing novel and fast numerical algorithms for various scientific computing (especially imaging) applications. He has been the director of multidisciplinary research at the Centre for Mathematical Imaging Techniques (CMIT) since 2007, and the director of the EPSRC Liverpool Centre of Mathematics in Healthcare (LCMH) since 2015. He heads a large group of computational imagers, tackling novel analysis of real-life images. His group's imaging work in variational modelling and algorithmic development is mostly interdisciplinary, strongly motivated by emerging real-life problems and their challenges: image restoration, image inpainting, tomography, image segmentation and registration.



Carola-Bibiane Schönlieb is Professor of Applied Mathematics at the University of Cambridge. There, she is head of the Cambridge Image Analysis group and co-director of the EPSRC Cambridge Mathematics of Information in Healthcare Hub. Since 2011, she is a fellow of Jesus College Cambridge and since 2016 a fellow of the Alan Turing Institute, London. She also holds the chair of the Committee for Applications and Interdisciplinary Relations (CAIR) of the EMS. Her current research interests focus on variational methods, partial differential equations and machine learning for image analysis, image processing and inverse imaging problems. She has active interdisciplinary collaborations with clinicians, biologists and physicists on biomedical imaging topics, chemical engineers and plant scientists on image sensing, as well as collaborations with artists and art conservators on digital art restoration.

Her research has been acknowledged by scientific prizes, among them the LMS Whitehead Prize 2016, the Philip Leverhulme Prize in 2017, the Calderon Prize 2019, a Royal Society Wolfson fellowship in 2020 and a doctorate honoris causa from the University of Klagenfurt in 2022, and by invitations to give plenary lectures at several renowned applied mathematics conferences, among them the SIAM conference on Imaging Science in 2014, the SIAM conference on Partial Differential Equations in 2015, the SIAM annual meeting in 2017, the Applied Inverse Problems Conference in 2019, the FOCM 2020 and the GAMM 2021.

Carola graduated from the Institute for Mathematics, University of Salzburg (Austria), in 2004. From 2004 to 2005, she held a teaching position in Salzburg. She received her PhD degree from the University of Cambridge (UK) in 2009. After 1 year of postdoctoral activity at the University of Göttingen (Germany), she became a lecturer at Cambridge in 2010, promoted to reader in 2015 and promoted to professor in 2018.



Prof. Xue-Cheng Tai is a chief research scientist and executive programme director at Hong Kong Center for Cerebro-cardiovascular Health Engineering (COCHE), Hong Kong Science Park. He is a professor and head of the Department of Mathematics at Hong Kong Baptist University (China) since 2017. Before 2017, he served as a professor in the Department of Mathematics at Bergen University (Norway). His research interests include numerical PDEs, optimisation techniques, inverse problems and image processing. He has done significant research work in his research areas and published more than 250 top-quality international conference and journal papers. He is the winner of the 8th Feng Kang Prize for scientific computing. Prof Tai has served as organising and programme committee member for a number of international conferences and has been often invited at international conferences. He has served as referee and reviewer for many premier conferences and journals.



Laurent Younes is a professor in the Department Applied Mathematics and Statistics, Johns Hopkins University (USA), which he joined in 2003, after 10 years as a researcher for the CNRS in France. He is a former student of Ecole Normale Supérieure (Paris) and of the University of Paris 11 from which he received his PhD in 1988. His work includes contributions to applied probability, statistics, graphical models, shape analysis and computational medicine. He is a fellow of the IMS and of the AMS.

Contributors

Jonas Adler Department of Mathematics, KTH – Royal Institute of Technology, Stockholm, Sweden

Alexis Arnaudon Department of Mathematics, Imperial College, London, UK
Blue Brain Project, École polytechnique fédéral de Lausanne (EPFL), Geneva, Switzerland

Jean-François Aujol Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, Talence, France

Noor Badshah Department of Basic Sciences, University of Engineering and Technology, Peshawar, Pakistan

Egil Bae Norwegian Defence Research Establishment (FFI), Kjeller, Norway

Coloma Ballester IPCV, DTIC, University Pompeu Fabra, Barcelona, Spain

Martin Bauer Department of Mathematics, Florida State University, Tallahassee, FL, USA

E. J. Bekkers Amsterdam Machine Learning Lab, University of Amsterdam, Amsterdam, The Netherlands

Stefania Bellavia Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze (INdAM-GNCS members), Firenze, Italia

Martin Benning The School of Mathematical Sciences, Queen Mary University of London, London, UK

Tommaso Bianconcini Verizon Connect, Firenze, Italia

Alfred M. Bruckstein Computer Science Department, Technion – Israel Institute of Technology, Haifa, Israel

Aurélie Bugeau LaBRI, CNRS, Université de Bordeaux, Talence, France

Tan Bui-Thanh Department of Aerospace Engineering and Engineering Mechanics, The Oden Institute for Computational Engineering and Sciences, UT Austin, Austin, TX, USA

Xiaohao Cai School of Electronics and Computer Science, University of Southampton, Southampton, UK

Hernan Carrillo LaBRI, CNRS, Bordeaux INP, Université de Bordeaux, Bordeaux, France

Hei-Long Chan Chinese University of Hong Kong, Hong Kong, China

Raymond Chan Department of Mathematics, College of Science, City University of Hong Kong, Kowloon Tong, Hong Kong, China

Huibin Chang School of Mathematical Sciences, Tianjin Normal University, Tianjin, China

Nicolas Charon Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA

Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA

Ke Chen Department of Mathematical Sciences, Centre for Mathematical Imaging Techniques, University of Liverpool, Liverpool, UK

Yunmei Chen Department of Mathematics, University of Florida, Gainesville, FL, USA

Gary P. T. Choi Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Michaël Clément LaBRI, CNRS, Bordeaux INP, Université de Bordeaux, Bordeaux, France

Elaine Crooks Department of Mathematics, Swansea University, Swansea, UK

Yehuda Dar Electrical and Computer Engineering Department, Rice University, Houston, TX, USA

Jérôme Darbon Division of Applied Mathematics, Brown University, Providence, RI, USA

Noémie Debroux Pascal Institute, University of Clermont Auvergne, Clermont-Ferrand, France

Juan Carlos De los Reyes Research Center for Mathematical Modelling (MODEMAT), Escuela Politécnica Nacional, Quito, Ecuador

Yuping Duan Center for Applied Mathematics, Tianjin University, Tianjin, China

R. Duits Applied Differential Geometry, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Matthias J. Ehrhardt Institute for Mathematical Innovation, University of Bath, Bath, UK

-
- Florian Faucher** Faculty of Mathematics, University of Vienna, Vienna, Austria
- Xiangchu Feng** School of Mathematics and Statistics, Xidian University, Xi'an, China
- Caroline Geiersbach** Weierstrass Institute, Berlin, Germany
- Guy Gilboa** Technion – IIT, Haifa, Israel
- Rémi Giraud** Univ. Bordeaux, CNRS, IMS UMR5251, Bordeaux INP, Talence, France
- Carsten Gottschlich** Institute for Mathematical Stochastics, University of Göttingen, Göttingen, Germany
- Markus Grasmair** NTNU, Trondheim, Norway
- Hanlin Gu** Hong Kong University of Science and Technology, Hong Kong, China
- Xianfeng Gu** Stony Brook University, Stony Brook, NY, USA
- Luying Gui** Department of Mathematics, Nanjing University of Science and Technology, Nanjing, China
- Abderrahmane Habbal** Modeling and Data Science, Mohammed VI Polytechnic University Benguerir, Morocco
- Université Côte d'Azur, Inria, Sophia Antipolis, France
- Michal Haindl** Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czechia
- Markus Haltmeier** Department of Mathematics, University of Innsbruck, Innsbruck, Austria
- Andreas Hauptmann** Research Unit of Mathematical Sciences, University of Oulu, Oulu, Finland
- Darryl Holm** Department of Mathematics, Imperial College, London, UK
- Stephan F. Huckemann** Felix-Bernstein-Institute for Mathematical Statistics in the Biosciences, University of Göttingen, Göttingen, Germany
- Samuel Hurault** Bordeaux INP, CNRS, IMB, Université de Bordeaux, Talence, France
- Soomin Jeon** Department of Radiology, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA
- Moez Kallel** Laboratory for Mathematical and Numerical Modeling in Engineering Science (LAM SIN), University of Tunis El Manar, National Engineering School of Tunis, Tunis-Belvédère, Tunisia

Clemens Kirisits Faculty of Mathematics, University of Vienna, Vienna, Austria

Eric Klassen Department of Mathematics, Florida State University, Tallahassee, FL, USA

Nataša Krejić Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia

Gitta Kutyniok Ludwig-Maximilians-Universität München, Mathematisches Institut, München, Germany

Andreas Langer Centre for Mathematical Sciences, Lund University, Lund, Sweden

Gabriel P. Langlois Division of Applied Mathematics, Brown University, Providence, RI, USA

Alessandro Lanza Department of Mathematics, University of Bologna, Bologna, Italy

Alice Le Brigant Department of Applied Mathematics, University Paris, Paris, France

Carole Le Guyader INSA Rouen Normandie, Laboratory of Mathematics, Normandie University, Rouen, France

Chang-Ock Lee Department of Mathematical Sciences, KAIST, Daejeon, Republic of Korea

Na Lei Dalian University of Technology, Dalian, China

Chunming Li School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

Lingfeng Li Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, China

Department of Mathematics, Southern University of Science and Technology, Shenzhen, China

Haiguang Liu Microsoft Research-Asian, Beijing, China

Zhifang Liu School of Mathematical Sciences, Tianjin Normal University, Tianjin, China

Estefania Loayza-Romero Institute for Analysis and Numerics, University of Münster, Münster, Germany

Lok Ming Lui Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China

Sebastian Lunz Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK

Feng Luo Rutgers University, Piscataway, NJ, USA

Junyan Lyu Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China

Jun Ma Department of Mathematics, Nanjing University of Science and Technology, Nanjing, China

Stefano Marchesini SLAC National Laboratory, Menlo Park, CA, USA

Tingwei Meng Division of Applied Mathematics, Brown University, Providence, RI, USA

Serena Morigi Department of Mathematics, University of Bologna, Bologna, Italy

Benedetta Morini Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze (INdAM-GNCS members), Firenze, Italia

Elena Morotti Department of Political and Social Sciences, University of Bologna, Bologna, Italy

Valeriya Naumova Machine Intelligence Department, Simula Consulting and SimulaMet, Oslo, Norway

Sebastian Neumayer Institute of Mathematics, TU Berlin, Berlin, Germany

Linh Nguyen Department of Mathematics, University of Idaho, Moscow, ID, USA

Ozan Öktem Department of Information Technology, Division of Scientific Computing, Uppsala University, Uppsala, Sweden

Antonio Orlando CONICET, Departamento de Bioingeniería, Universidad Nacional de Tucumán, Tucumán, Argentina

Simone Parisotto DAMTP, University of Cambridge, Cambridge, UK

Elena Loli Piccolomini Department of Computer Science and Engineering, University of Bologna, Bologna, Italy

Fabien Pierre LORIA, UMR CNRS 7503, Université de Lorraine, INRIA projet Tangram, Nancy, France

J. W. Portegies Center for Analysis, Scientific Computing and Applications, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Michael Quellmalz Institute of Mathematics, Technical University Berlin, Berlin, Germany

Lara Raad LIGM, CNRS, Univ Gustave Eiffel, Marne-la-Vallée, France

Robin Richter Felix-Bernstein-Institute for Mathematical Statistics in the Biosciences, University of Göttingen, Göttingen, Germany

Erlend Skaldehaug Riis The Department of Applied Mathematics and Theoretical Physics, Cambridge, UK

Otmar Scherzer Faculty of Mathematics, University of Vienna, Vienna, Austria

Carola Schönlieb Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK

Johannes Schwab Department of Mathematics, University of Innsbruck, Innsbruck, Austria

Ivan W. Selesnick Department of Electrical and Computer Engineering, New York University, New York, NY, USA

Eric Setterqvist Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz, Austria

Fiorella Sgallari Department of Mathematics, University of Bologna, Bologna, Italy

B. M. N. Smets Applied Differential Geometry, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Stefan Sommer Department of Computer Science (DIKU), University of Copenhagen, Copenhagen E, Denmark

Xiaoliang Song School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, China

Gabriele Steidl Institute of Mathematics, TU Berlin, Berlin, Germany

Xuecheng Tai Hong Kong Center for Cerebro-cardiovascular Health Engineering (COCHE), Shatin, Hong Kong, China

Roger Tam School of Biomedical Engineering, The University of British Columbia, Vancouver, BC, Canada

Xiaoying Tang Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China

Duy H. Thai Department of Mathematics, Colorado State University, Fort Collins, CO, USA

Anis Theljani Department of Mathematical Sciences, University of Liverpool Mathematical Sciences Building, Liverpool, UK

Iлона Christy Unarta Hong Kong University of Science and Technology, Hong Kong, China

Tuomo Valkonen Center for Mathematical Modeling, Escuela Politécnica Nacional, Quito, Ecuador

Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland

David Villacís Research Center for Mathematical Modelling (MODEMAT), Escuela Politécnica Nacional, Quito, Ecuador

Patricia Vitoria IPCV, DTIC, University Pompeu Fabra, Barcelona, Spain

Samad Wali School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

Yang Wang Hong Kong University of Science and Technology, Hong Kong, SAR, China

Kathrin Welker Faculty of Mechanical Engineering and Civil Engineering, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Hamburg, Germany

A. J. Wemmenhove Applied Differential Geometry, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Jonathan Wittmer Department of Aerospace Engineering and Engineering Mechanics, UT Austin, Austin, TX, USA

Chunlin Wu School of Mathematical Sciences, Nankai University, Tianjin, China

Yin Xian Hong Kong Applied Science and Technology Research Institute (ASTRI), Hong Kong, China

TCL Research Hong Kong, Hong Kong, SAR, China

Huilin Yang Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China

Li Yang School of Mathematical Sciences, Tianjin Normal University, Tianjin, China

Xiaoping Yang Department of Mathematics, Nanjing University, Nanjing, China

Yuan Yao Hong Kong University of Science and Technology, Hong Kong, China

Shing-Tung Yau Harvard University, Cambridge, MA, USA

Xiaojing Ye Department of Mathematics and Statistics, Georgia State University, Atlanta, GA, USA

Laurent Younes Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA

Bo Yu School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, China

Tieyong Zeng Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China

Qingchao Zhang Department of Mathematics, University of Florida, Gainesville, FL, USA

Lingyan Zhang School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

Kewei Zhang School of Mathematical Sciences, University of Nottingham, Nottingham, UK

Xiaolong Zhu School of Mathematics and Statistics, Xidian University, Xi'an, China

Part I
Convex and Non-convex Large-Scale
Optimization in Imaging



Convex Non-convex Variational Models

1

Alessandro Lanza, Serena Morigi, Ivan W. Selesnick,
and Fiorella Sgallari

Contents

Introduction	4
Convex or Non-convex: Main Idea and Related Works	10
Sparsity-Inducing Separable Regularizers	11
CNC Models with Sparsity-Inducing <i>Separable</i> Regularizers	16
Sparsity-Inducing Non-separable Regularizers	22
CNC Models with Sparsity-Inducing <i>Non-separable</i> Regularizers	24
Construction of Matrix B	25
A Simple CNC Example	27
Path of Solution Components	29
Forward-Backward Minimization Algorithms	30
FB Strategy for Separable CNC Models	32
FB Strategy for Non-separable CNC Models	35
Efficient Solution of the Backward Steps by ADMM	36
Numerical Examples	41
Examples Using CNC Separable Models	46
Examples Using CNC Non-separable Models	49
Conclusion	57
References	57

Abstract

An important class of computational techniques to solve inverse problems in image processing relies on a variational approach: the optimal output is obtained by finding a minimizer of an energy function or “model” composed of two terms,

A. Lanza · S. Morigi · F. Sgallari (✉)

Department of Mathematics, University of Bologna, Bologna, Italy

e-mail: alessandro.lanza2@unibo.it; serena.morigi@unibo.it; fiorella.sgallari@unibo.it

I. W. Selesnick

Department of Electrical and Computer Engineering, New York University, New York, NY, USA

e-mail: selesi@nyu.edu

the data-fidelity term, and the regularization term. Much research has focused on models where both terms are convex, which leads to convex optimization problems. However, there is evidence that non-convex regularization can improve significantly the output quality for images characterized by some sparsity property. This fostered recent research toward the investigation of optimization problems with non-convex terms. Non-convex models are notoriously difficult to handle as classical optimization algorithms can get trapped at unwanted local minimizers. To avoid the intrinsic difficulties related to non-convex optimization, the convex non-convex (CNC) strategy has been proposed, which allows the use of non-convex regularization while maintaining convexity of the total cost function. This work focuses on a general class of parameterized non-convex sparsity-inducing separable and non-separable regularizers and their associated CNC variational models. Convexity conditions for the total cost functions and related theoretical properties are discussed, together with suitable algorithms for their minimization based on a general forward-backward (FB) splitting strategy. Experiments on the two classes of considered separable and non-separable CNC variational models show their superior performance than the purely convex counterparts when applied to the discrete inverse problem of restoring sparsity-characterized images corrupted by blur and noise.

Keywords

Convex non-convex optimization · Sparsity regularization · Image restoration · Alternating direction method of multipliers · Forward backward algorithm

Introduction

A wide class of linear systems derived from the discretization of linear ill-posed inverse problems in data processing is characterized by high dimensionality, ill-conditioned matrices, and noise-corrupted data. In this class of discrete inverse problems, a noisy indirect observation $b \in \mathbb{R}^m$ of an original unknown image $x \in \mathbb{R}^n$ is modeled as

$$b = Ax, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ accounts for the data-acquisition system. For instance, A can be a convolution matrix modeling optical blurring, a wavelet or Fourier transform matrix in image synthesis, a radon transform matrix in X-ray computerized tomography, a sampling matrix in compressed sensing, a binary selection matrix in image inpainting, or the identity matrix in image denoising and segmentation.

When $m < n$, the linear system (1) is underdetermined and among the infinity of solutions, it is common to seek an approximate solution with minimal norm, that is, one solves the constrained optimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|_2^2 \quad \text{subject to} \quad b = Ax, \quad (2)$$

where $\|v\|_2$ denotes the ℓ_2 norm of vector v .

On the other hand, when $m > n$, the linear system (1) is overdetermined; in general there is no solution, and it is common to seek for the least squares solution, that is, the solution which minimizes the residual norm; in formula,

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|_2^2. \quad (3)$$

Even in the most favorable case that $m = n$, so that the linear system (1) can admit a unique solution, ill-conditioning of matrix A typically makes the problem very difficult from a numerical point of view.

Indeed, for many image processing applications of practical interest, problems in form (1) are *ill-posed* linear inverse problems. The term *ill-posed* was coined in the early twentieth century by Hadamard who defined a linear problem to be well-posed if it satisfies the following three requirements:

- Existence: The problem must have a solution.
- Uniqueness: The problem must have only one solution.
- Stability: The solution must depend continuously on the data.

If the problem violates one or more of these requirements, it is said to be ill-posed (Hansen 1997).

A violation of the stability condition implies that arbitrarily small perturbations of the data can produce arbitrarily large perturbations in the solution. Noise is a typical unavoidable perturbation component in the digital data acquisition process which, coupled with ill-conditioning of matrix A , makes inverse problems in imaging typically ill-posed.

In this work, we assume that the noise is additive white Gaussian (AWG), so that the observed noisy image $b \in \mathbb{R}^m$ is related to the underlying true image $x \in \mathbb{R}^n$ by means of the following degradation model

$$b = Ax + \eta, \quad (4)$$

with $\eta \in \mathbb{R}^m$ the realization of an m -dimensional random vector having Gaussian distribution with zero mean and scalar covariance matrix. In many practical cases, the matrix A is so ill-conditioned (if not numerically singular) that recovering x given b and A by means of a naive (not regularized) least-squares procedure leads to meaningless results. Some sort of regularization is required. The key aspect is to reformulate the problem such that the solution to the new problem is less sensitive to the perturbations. We say that we stabilize or regularize the problem.

Regularization strategies in traditional variational methods are usually problem-dependent and take advantage of a priori information specific to any particular imaging application. In this paper, we focus on those applications which involve

sparsity in the solution, or in its representation, or in a function of the solution. For instance, images of stars from a telescope are sparse themselves, while images of humans are sparse under the wavelet transform. Sparsity plays an important role in image processing and machine learning. How to build appropriate sparse-based models, how to numerically find solutions of the sparse-based models, and how to derive theoretical guarantees of the correctness of the solutions are essential for the success of sparsity in a wide range of applications (Bruckstein et al. 2009).

We focus on regularized variational methods where an approximate solution $x^* \in \mathbb{R}^n$ of the inverse problem (4) is sought among the (global) minimizers of a cost function $\mathcal{J}: \mathbb{R}^n \rightarrow \mathbb{R}$ which takes the following form

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \mathcal{J}(x), \quad \mathcal{J}(x) = \frac{1}{2} \|Ax - b\|_2^2 + \mu \Psi(x). \quad (5)$$

The quadratic term in (5) is the so-called L_2 *fidelity term*, which forces closeness of solution(s) x^* to data b according to the linear acquisition model (4) and to the assumed noise Gaussian distribution. The term $\Psi(x)$ in (5) represents the sparsity-inducing *regularization term* and encodes some sparsity priors on the unknown sought image. Finally, the positive scalar μ , referred to as the *regularization parameter* of variational model (5), is a free parameter which allows to control the trade-off between data fidelity and regularization.

In this work, we are particularly interested in sparsity-promoting regularization terms $\Psi: \mathbb{R}^n \rightarrow \mathbb{R}$ having the following general form

$$\Psi(x) := \Phi(x, y), \quad y := G(z) \quad z := Lx, \quad (6)$$

with

- $L \in \mathbb{R}^{r \times n}$ the regularization matrix
- $G: \mathbb{R}^r \rightarrow \mathbb{R}^s$ a possibly nonlinear vector-valued function with $g_i: \mathbb{R}^r \rightarrow \mathbb{R}$, $i = 1, \dots, s$, representing its scalar-valued components
- $y \in \mathbb{R}^s$ the features vector to be sparsified
- $\Phi: \mathbb{R}^n \times \mathbb{R}^s \rightarrow \mathbb{R}$ a sparsity-promoting penalty function (Selesnick and Bayram 2014; Selesnick et al. 2015; Lanza et al. 2016a)

It is important for the purposes of this work to introduce a partition of the class of sparsity-promoting regularizers Ψ defined in (6) into two sub-classes based on *separable* and *non-separable* penalty functions Φ .

Definition 1 (Separable and non-separable sparsity-promoting regularizers).

A sparsity-inducing regularizer Ψ of the form in (6) is referred to as *separable* (with respect to the feature vector y to be sparsified) if the penalty function Φ only depends on y and is additively separable with respect to the y components; in formula,

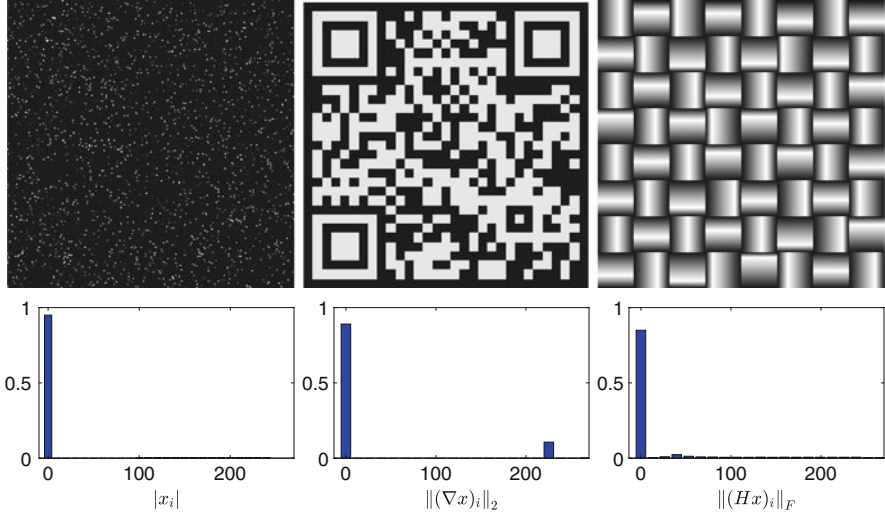


Fig. 1 Prototypical example images characterized by different sparse feature vectors (first row) and their associated normalized histograms (second row)

$$\Phi(y) = \sum_{i=1}^s \phi_i(y_i), \quad \text{with } \phi_i : \mathbb{R} \rightarrow \mathbb{R}, \quad (7)$$

otherwise, it is named *non-separable*.

Examples of image feature vectors $y = G(Lx)$ which can be characterized by a sparsity property in typical application scenarios are, e.g., the vectorized image itself (for predominantly zero images), the vector of image gradient magnitudes (for piecewise constant images), the vector of image Hessian Frobenious norms (for piecewise affine images), and the vector of coefficients of the image in a transformed domain (e.g., Fourier, wavelet, ...).

Examples of predominantly zero, piecewise constant, and piecewise affine images are depicted in the first row of Fig. 1. They are characterized, from left to right, by a sparse vector y of components $y_i = |x_i|$, $y_i = \|(\nabla x)_i\|_2$ and $y_i = \|(Hx)_i\|_F$, $i = 1, \dots, n$, respectively, where $(\nabla x)_i \in \mathbb{R}^2$ and $(Hx)_i \in \mathbb{R}^{2 \times 2}$ represent the gradient and the Hessian matrix of image x at pixel i , respectively. In the second row of Fig. 1, the reported normalized histograms of the corresponding y vector values clearly highlight their sparsity.

Although the three images above represent almost ideal prototypes, also many images from real-life applications commonly exhibit sparsity features. In Fig. 2, we show three realistic images characterized by increasing level of sparsity of the gradient magnitudes, together with their associated histograms. This indicates the practical importance of sparse-regularized variational models which, in many application scenarios, hold the potential for very high quality results.

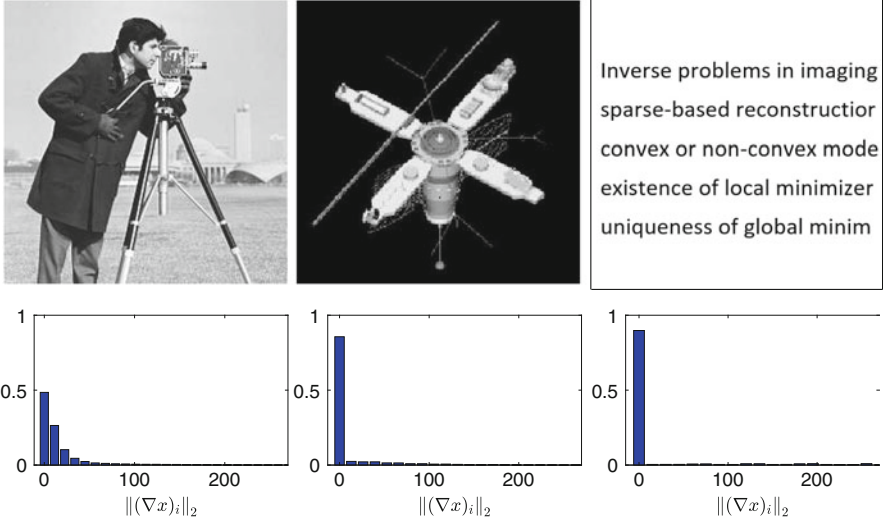


Fig. 2 Realistic images characterized, from left to right, by increasing level of sparsity of the gradient magnitudes (first row) and their associated normalized histograms (second row)

Some interesting models of the form (5)–(6) are characterized by the following well-known matrices A and L :

- **TV- L_2 Restoration:** In image restoration, the popular Total Variation (TV)- L_2 (Rudin et al. 1992) calls for a matrix A characterizing the image blur, or $A = I_n$ for image denoising. For what concerns the linear operator L , it is defined as $L := (D_h^T, D_v^T)^T \in \mathbb{R}^{2n \times n}$ with $D_h, D_v \in \mathbb{R}^n$ finite difference matrices discretizing the first-order horizontal and vertical partial derivatives, respectively, $g_i(z) := \|(z_i, z_{i+n})\|_2$ or $g_i(z) = \|(z_i, z_{i+n})\|_1$, $i = 1, \dots, n$, for isotropic and anisotropic TV regularization, respectively, and Φ the ℓ_1 norm function; in formulas

$$\text{TV}(x) = \|G(Lx)\|_1 = \sum_{i=1}^n |g_i(Lx)| = \begin{cases} \sum_{i=1}^n \sqrt{(D_h x)_i^2 + (D_v x)_i^2} & \text{(isotropic)} \\ \sum_{i=1}^n (|(D_h x)_i| + |(D_v x)_i|) & \text{(anisotropic)} \end{cases} \quad (8)$$

- **Sparse Reconstruction (Analysis):** A full rank, $L := W$ with W an orthogonal basis or an overcomplete dictionary, which satisfies the tight frame condition, i.e., $L^T L = \delta I_n$, $\delta > 0$, Parekh and Selesnick (2015).

- **Sparse Reconstruction (Synthesis):** $A := W^{-1}$, $L = I_n$, and G the identity operator.

The main difficulties in solving variational models of the considered form in (5)–(6) stems from the facts that the involved optimization domain is typically of very high dimension (the number of pixels in the image), the linear operator A can be ill-conditioned or even singular, and, more importantly, the regularization term Ψ is preferably a non-convex non-smooth function in order to effectively promote sparsity of vector y . Summarizing, (5)–(6) is a very challenging large-scale optimization problem. The real challenge comes from possible non-convexity of the problem, which yields all the well-known associated intricacies, namely, the existence of local minimizers and the problematic convergence of minimization algorithms.

A very interesting approach proposed in literature to address this issue is the so-called CNC strategy. It consists in constructing and then minimizing convex cost functions containing non-convex (sparsity-promoting) regularization terms. This can be obtained by using regularizers parameterized such that their degree of non-convexity can be tuned. By suitably setting the parameters of the regularizer, one can thus obtain a convex variational model containing a non-convex regularizer which holds the potential to induce sparsity of the solution more effectively than any convex regularizer. As it will be shown in this work, suitably parameterized non-separable regularizers of the form in (6) allow to apply the CNC strategy to the solution of any linear inverse problem in imaging, thus overcoming the intrinsic limitations of separable regularizers.

The chapter contents will be organized as follows. In section “[Convex or Non-convex: Main Idea and Related Works](#),” we outline the main ideas at the basis of the CNC strategy and shortly review the most related approaches. In section “[Sparsity-Inducing Separable Regularizers](#),” we present separable non-convex parameterized regularizers, and then in section “[CNC Models with Sparsity-Inducing Separable Regularizers](#),” we illustrate the associated CNC models and the related convexity condition results. In section “[Sparsity-Inducing Non-separable Regularizers](#),” we present non-separable non-convex parameterized regularizers, while their integration into suitable CNC models is described in section “[CNC Models with Sparsity-Inducing Non-separable Regularizers](#),” together with the construction of the related matrix B which leads to convexity of the total cost function. An illustrative example of CNC separable and non-separable models is given in section “[A Simple CNC Example](#).” In section “[Forward-Backward Minimization Algorithms](#),” we outline the optimization algorithms for solving the illustrated classes of CNC variational models, based on the FB splitting strategy and the Alternating Direction Method of Multipliers (ADMM) for the related subproblems. Finally, in section “[Numerical Examples](#),” we evaluate experimentally the performance of the two CNC classes when applied to the linear ill-posed inverse problem of restoring images corrupted by blur and noise.

Convex or Non-convex: Main Idea and Related Works

Convexity is a sufficient condition for all local minima to be global minima. If \mathcal{J} is non-convex, it may have many local minima which are not global minima. This means that classical convex optimization algorithms applied to a non-convex cost function \mathcal{J} will almost certainly get trapped at a local minimum that is of higher cost than the global minimum. Moreover, which local minimum is reached will depend strongly on the starting point of the algorithm.

However, non-convex non-smooth optimization problems arise more and more frequently in image processing, neural network training, and machine learning, where suitable non-convex regularizers have shown superior performance with respect to their convex counterparts (Nikolova 2011; Bruckstein et al. 2009). In the literature, for example, the most natural sparsity-inducing penalty is the ℓ_0 pseudo-norm, which, however, leads to NP-hard and non-convex optimization problems.

Literature on non-convex optimization dates back to the 1950s. An important class of non-convex optimization problems that has been extensively studied in the past is related to the specific set of non-convex cost functions that can be defined as the difference of convex functions, or DC functions for short; we refer to the seminal papers Tuy (1995) and Hartman (1959) and the more recent work Yuille and Rangarajan (2003) for more details on DC functions and optimization. Other important approaches to optimization in the non-convex regime are represented, e.g., by simulated annealing, see Geman and Geman (1984); genetic algorithms, see Jensen and Nielsen (1992); the Mean Field Annealing by Geiger and Girosi, which provides a deterministic version of simulated annealing (Geiger and Girosi 1991); and the Graduated Non-Convexity (GNC) strategy introduced in Blake and Zisserman (1987) by Blake and Zisserman.

The basic idea of the popular GNC algorithmic strategy is to construct a modified, parameterized cost function \mathcal{J}_λ , governed by a control parameter $\lambda \in [0, 1]$, chosen so that $\mathcal{J}_0 = \mathcal{J}$, the true cost function, and $\mathcal{J}_1 = \mathcal{J}_c$, a convex approximation to \mathcal{J} . Then GNC computes a solution to the non-convex problem by starting from its convex approximation \mathcal{J}_c , which must have a global minimum, and gradually changing λ (i.e., gradually increasing the amount of non-convexity) until the original non-convex function \mathcal{J} is recovered. The solution obtained at each iteration is used as initial guess for the subsequent iteration. In the construction of a suitable convex surrogate function \mathcal{J}_c , the authors in Blake and Zisserman (1987) introduced the concept of “balancing” the positive second derivatives in the first term (fidelity) against the negative second derivatives in the regularization term. This represents the seminal idea behind the CNC strategy, namely, designing non-convex parameterized penalty terms which allow to maintain convexity of the total cost function.

This simple concept, later called the CNC strategy (Lanza et al. 2015), has been applied by Nikolova (1998) in the context of denoising of binary images and then extended to many other sparse-regularized variational problems (Bayram

2016; Selesnick and Bayram 2014; Lanza et al. 2017), including 1D and 2D total variation denoising (Lanza et al. 2016a; Malek-Mohammadi et al. 2016; Zou et al. 2019; Du and Liu 2018), transform-based denoising (Parekh and Selesnick 2015; Ding and Selesnick 2015), low-rank matrix estimation (Parekh and Selesnick 2016), decomposition and segmentation of images and scalar fields over surfaces (Chan et al. 2017; Huska et al. 2019a,b), as well as machine fault detection (Cai et al. 2018; Wang et al. 2019).

The flexibility and effectiveness of the CNC approach depends on the construction of non-trivial separable and non-separable convex functions. It turns out that Moreau envelopes and infimal convolutions are useful for this purpose (Selesnick 2017a,b; Carlsson 2016; Soubies et al. 2015). Based on convex analysis, families of non-convex non-separable penalty functions have been proposed in Selesnick (2017a) that do maintain convexity of the cost functional \mathcal{J} for any matrix A , but only in the special case where both G and L in (6) are identity operators. More recently, a convex approach was applied in Lanza et al. (2019) where a general CNC framework is proposed for constructing non-separable non-convex regularizers starting from any convex regularizer, any matrix A and L , and quite general functions G . In particular, an infimal convolution is subtracted from a convex regularizer, such as the ℓ_1 -norm, leading to a resulting non-convex regularizer.

Non-convex penalties of various functional forms have been proposed too for overcoming limitations of the ℓ_1 norm by using penalties that promote sparsity more strongly (Castella and Pesquet 2015; Candés et al. 2008; Nikolova 2011; Nikolova et al. 2010; Chartrand 2014; Chouzenoux et al. 2013; Portilla and Mancera 2007; Shen et al. 2016). However, these methods do not aim to maintain convexity of the cost function to be minimized. Moreover, for what concerns non-separable sparsity-inducing penalties in (6), pioneering work has been conducted in Tipping (2001) and Wipf et al. (2011); however, also such penalties were not designed to maintain cost function convexity.

We finally note that infimal convolution (related to the Moreau envelope) has been used to define generalized TV regularizers (Setzer et al. 2011; Chambolle and Lions 1997; Burger et al. 2016; Becker and Combettes 2014). However, the aims and methodologies of these past works are quite different from those considered here. In fact, in these works, the ℓ_1 norm is replaced by an infimal convolution; the resulting regularizer is convex.

Sparsity-Inducing Separable Regularizers

In this section, we first recall some definitions which will be useful for the rest of the work, and, in particular, we report some results from convex analysis. We then review some popular sparsity-inducing separable regularizers and discuss their properties.

In this work, we denote by \mathbb{R}_+ and \mathbb{R}_{++} the sets of nonnegative and positive real numbers, respectively, by I_n the identity matrix of order n , by 0_n the n -dimensional null vector, by $\text{null}(M)$ the null space of matrix M , and by $\Gamma_0(\mathbb{R}^n)$ the set of proper lower semicontinuous convex functions from \mathbb{R}^n to $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$.

Definition 2 (infimal convolution). Let $f, g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$. The infimal convolution of f and g is defined by

$$(f \square g)(x) = \inf_{v \in \mathbb{R}^n} \{f(v) + g(x - v)\}. \quad (9)$$

and it is said to be exact and denoted by $f \boxdot g$ if the infimum above is attained for any $x \in \mathbb{R}^n$, namely, $(f \boxdot g)(x) = \min_{v \in \mathbb{R}^n} \{f(v) + g(x - v)\}$, for any $x \in \mathbb{R}^n$.

Definition 3 (Moreau envelope). Let $f \in \Gamma_0(\mathbb{R}^n)$ and let $a \in \mathbb{R}_{++}$. The Moreau envelope of f with parameter a is defined by

$$\text{env}_f^a(x) = \left(f \boxdot \frac{a}{2} \|\cdot\|_2^2 \right)(x) = \min_{v \in \mathbb{R}^n} \left\{ f(v) + \frac{a}{2} \|x - v\|_2^2 \right\}. \quad (10)$$

Definition 4 (proximity operator). Let $f \in \Gamma_0(\mathbb{R}^n)$ and let $a \in \mathbb{R}_{++}$. The proximity operator of f with parameter a is defined by

$$\text{prox}_f^a(x) = \arg \min_{v \in \mathbb{R}^n} \left\{ f(v) + \frac{a}{2} \|x - v\|_2^2 \right\}. \quad (11)$$

We notice that, for any $f \in \Gamma_0(\mathbb{R}^n)$, $a \in \mathbb{R}_{++}$, the cost function $f(v) + \frac{a}{2} \|x - v\|_2^2$ in (10)–(11) is strongly convex in v ; hence it admits a unique (global) minimizer.

Definition 5 (Huber function). The Huber function $h_a : \mathbb{R} \rightarrow \mathbb{R}_+$ with parameter $a \in \mathbb{R}_{++}$ is defined by

$$h_a(t) = \text{env}_{|\cdot|}^a(t) = \min_{v \in \mathbb{R}} \left\{ |v| + \frac{a}{2} (t - v)^2 \right\} = \begin{cases} \frac{a}{2} t^2 & \text{for } |t| \in [0, 1/a], \\ |t| - \frac{1}{2a} & \text{for } |t| \in]1/a, +\infty[. \end{cases} \quad (12)$$

Definition 6 (minimax concave penalty function). The minimax concave (MC) penalty function $\phi_{\text{MC}} : \mathbb{R} \rightarrow \mathbb{R}_+$ with parameter $a \in \mathbb{R}_{++}$ is defined by

$$\phi_{\text{MC}}(t; a) = |t| - h_a(t) = \begin{cases} -\frac{a}{2} t^2 + |t| & \text{for } |t| \in [0, 1/a], \\ \frac{1}{2a} & \text{for } |t| \in]1/a, +\infty[. \end{cases} \quad (13)$$

Proposition 1 (Moreau envelope gradient). *Let $f \in \Gamma_0(\mathbb{R}^n)$ and let $a \in \mathbb{R}_{++}$. Then, the Moreau envelope of f with parameter a is a differentiable function with gradient given by*

$$\nabla \left(\text{env}_f^a \right) (x) = a \left(x - \text{prox}_f^a(x) \right). \quad (14)$$

Proposition 2. *Let $h_a : \mathbb{R} \rightarrow \mathbb{R}$ be the Huber function defined in (12). Then, for any value of the parameter $a \in \mathbb{R}_{++}$ the function*

$$f_a(z) := h_a(\|x\|_2), \quad x \in \mathbb{R}^n, \quad (15)$$

is continuously differentiable and its gradient is given by

$$\nabla f_a(x) = \min \left\{ a, \frac{1}{\|x\|_2} \right\} x. \quad (16)$$

Proof. Recalling the Huber function definition in (12), the function f_a in (15) takes the explicit form

$$f_a(x) = \begin{cases} \frac{a}{2} \sum_{i=1}^n x_i^2 & \text{for } \|x\|_2 \in [0, 1/a], \\ \sqrt{\sum_{i=1}^n x_i^2} - \frac{1}{2a} & \text{for } \|x\|_2 \in]1/a, +\infty[. \end{cases} \quad (17)$$

The two pieces of function f_a in (17) are clearly both continuously differentiable on their domain with gradients given by

$$\nabla f_a(x) = \begin{cases} ax & \text{for } \|x\|_2 \in [0, 1/a], \\ \frac{1}{\|x\|_2} x & \text{for } \|x\|_2 \in]1/a, +\infty[. \end{cases} \quad (18)$$

It follows easily from (18) that, for any $a \in \mathbb{R}_{++}$, the gradient function $\nabla f_a(x)$ is continuous also at points x on the spherical surface $\|x\|_2 = 1/a$ separating its two pieces. Finally, the compact form of ∇f_a given in (16) comes straightforwardly from (18). \square

Among separable sparsity-promoting regularizers (see Definition 1), the most natural choice is represented by the ℓ_0 pseudo-norm of the features vector y to sparsify, namely, $\Phi(y) = \|y\|_0 = \#\{i : y_i \neq 0\}$, as it directly measures the sparsity of y by counting the number of non-zero elements in it (see the dashed magenta line in Fig. 3a). However, ℓ_0 regularization leads to non-convex NP-hard optimization problems. Intrinsic difficulties involved in using the ℓ_0 pseudo-norm can be overcome by using the ℓ_1 norm, namely, $\Phi(y) = \|y\|_1 = \sum_{i=1}^S |y_i|$ (see

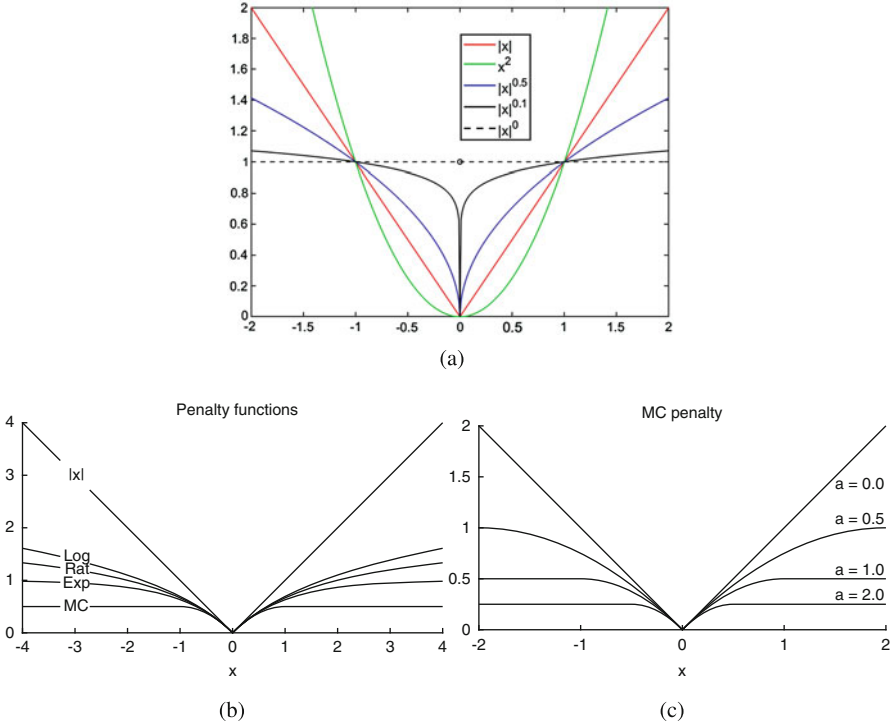


Fig. 3 Sparsity-inducing scalar penalties: (a) ℓ_p penalty for some different p values, (b) some parameterized non-convex penalties satisfying assumptions 1–5 (see Table 1) and the MC penalty (see definition in (13)) all with concavity parameter $a = 1$, and (c) MC penalty for some different values of the concavity parameter a

the solid red curve in Fig. 3a). In fact, this choice very likely leads to a convex sparsity-inducing regularizer and, hence, to a convex variational model which can be solved numerically by standard convex optimization algorithms. However, it is well known that the ℓ_1 norm penalty function tends to underestimate high-amplitude components of the vector to which it is applied, in our case $y = G(Lx)$. More generally, it is well known that non-convex penalty functions hold the potential for inducing sparsity more effectively than convex penalty functions. A natural non-convex separable alternative to the ℓ_1 norm is the ℓ_p quasi-norm penalty (Sidky et al. 2014), $\Phi(y) = \frac{1}{p} \|y\|_p^p = \frac{1}{p} \sum_{i=1}^s |y_i|^p$, $0 < p < 1$; see the solid blue and black curves in Fig. 3a, corresponding to $p = 0.5$ and $p = 0.1$, respectively. However, such a non-convex family of penalties can not be used to the purpose of constructing CNC variational models. In fact, since the infimum of the second-order derivative of the ℓ_p penalty is equal to $-\infty$ for any $p \in]0, 1[$, it is not possible to obtain a total convex model even when coupling the regularizer with a strongly convex quadratic fidelity term.

To the aim of constructing CNC models with separable sparsity-promoting regularizers characterized by tunable degree of non-convexity, one can usefully consider the class of parameterized scalar penalty functions $\phi(t; a) : \mathbb{R} \rightarrow \mathbb{R}$ which, for any value of the parameter $a \in \mathbb{R}_+$, satisfy the following assumptions:

1. $\phi(t; a) \in C^0(\mathbb{R}) \cap C^2(\mathbb{R} \setminus \{0\})$
2. $\phi(t; a) = \phi(-t; a) \quad \forall t \in \mathbb{R}_{++}$
3. $\phi'(t; a) \geq 0 \quad \forall t \in \mathbb{R}_{++}$
4. $\phi''(t; a) \leq 0 \quad \forall t \in \mathbb{R}_{++}$
5. $\phi(0; a) = 0, \quad \inf_{t \in \mathbb{R}_{++}} \phi''(t; a) = -a$

We denoted by $\phi'(t; a)$ and $\phi''(t; a)$ the first-order and second-order derivatives of ϕ with respect to the variable t , respectively. Assumptions 1–5 above are quite standard and encompass a wide class of continuous but non-smooth non-convex sparsity-promoting penalty functions (Geman and Geman 1984). The parameter a , referred to as the penalty concavity parameter, is directly related to the degree of non-convexity of the penalty function, as defined in assumption 5.

In Table 1, we report the definitions of four widely used sparsity-promoting parameterized scalar penalty functions, referred to as ϕ_{\log} , ϕ_{rat} , ϕ_{atan} , and ϕ_{exp} , which satisfy all the assumptions 1–5 and have been considered, e.g., in Selesnick and Bayram (2014), Chen and Selesnick (2014), and Lanza et al. (2015, 2016a). In particular, the penalty ϕ_{atan} has been proposed in Selesnick and Bayram (2014) as the maximally sparsity-inducing function among those characterized by a first-order derivative of inverse quadratic polynomial type.

In order to mimic in a more faithful manner not only the asymptotically constant behavior of the ℓ_0 pseudo-norm, a class of piecewise defined truncated penalties has been introduced in literature. One of the most popular and effective representatives of this class is the so-called minimax concave (MC) penalty function, formally defined in (13) and also reported in the last row of Table 1. In the rest of this work, we will use the MC penalty within all the illustrated separable CNC variational models.

Table 1 Four popular non-convex, sparsity-promoting, parameterized scalar penalty functions $\phi(t; a) : \mathbb{R} \rightarrow \mathbb{R}_+$ satisfying assumptions 1–5 and, in the last row, the MC penalty function

$\phi_{\log}(t; a)$	$= \frac{\log(1 + at)}{a}$
$\phi_{\text{rat}}(t; a)$	$= \frac{t}{1 + at/2}$
$\phi_{\text{atan}}(t; a)$	$= \frac{\text{atan}\left(\frac{1+2at}{\sqrt{3}}\right) - \frac{\pi}{6}}{a\sqrt{3}/2}$
$\phi_{\text{exp}}(t; a)$	$= \frac{1 - e^{-at}}{a}$
$\phi_{\text{MC}}(t; a)$	$= \begin{cases} -\frac{a}{2}t^2 + t & \text{for } t \in [0, 1/a] \\ \frac{1}{2a} & \text{for } t \in]1/a, +\infty[\end{cases}$

To give a visual insight of the considered parameterized penalty functions, in Fig. 3b we depict the graphs of some of the penalties in Table 1, all with concavity parameter $a = 1$, whereas in Fig. 3c we illustrate the MC penalty for some different values of the concavity parameter a .

CNC Models with Sparsity-Inducing *Separable* Regularizers

This section is concerned with the formulation of CNC variational models with *separable* sparsity-promoting regularization terms; see Definition 1. The general form of such models reads

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \mathcal{J}_S(x; a), \quad (19)$$

$$\mathcal{J}_S(x; a) = \frac{1}{2} \|Ax - b\|_2^2 + \mu \Psi_S(x; a), \quad \Psi_S(x; a) = \sum_{i=1}^s \phi_{\text{MC}}(g_i(Lx); a_i), \quad (20)$$

where, we recall, $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{r \times n}$ are the coefficient matrices of two bounded linear operators, $g_i : \mathbb{R}^r \rightarrow \mathbb{R}$, $i = 1, \dots, s$ are the components of a possibly nonlinear vector-valued function $G : \mathbb{R}^r \rightarrow \mathbb{R}^s$, $\mu \in \mathbb{R}_{++}$ is the regularization parameter, $\phi_{\text{MC}} : \mathbb{R} \rightarrow \mathbb{R}_+$ is the non-convex MC penalty function defined in (13), and where we introduced the vector $a := (a_1, \dots, a_s)^T \in \mathbb{R}_{++}^s$ containing the concavity parameters of all the s instances of the MC penalty in the regularizer Ψ_S . We refer to (19)–(20) as the class of CNC separable (least-squares) models, abbreviated CNC-S- L_2 models.

In order to refer to models (19)–(20) as CNC, we clearly need to derive and then impose convexity conditions for the objective function \mathcal{J}_S . More precisely, we seek sufficient conditions on the operators A , L , and G and on the parameters μ and a_i , $i = 1, \dots, s$, to ensure that the function \mathcal{J}_S in (20) is convex (strongly convex) on its entire domain $x \in \mathbb{R}^n$. It is worth noting that, in practice, the operators A , L , and G are commonly prescribed by the specific application at hand. In fact, operator A typically comes from a (more or less accurate) modeling of the image acquisition process, whereas operators L and G are related to the expected properties of sparsity of the sought solution. This implies that the derived convexity conditions can be regarded as constraints on the free parameters μ and a_i of model (19)–(20).

In Lemma 1, we give some useful reformulations of the separable regularizer Ψ_S defined in (20); then in Theorem 1, we derive conditions for convexity of \mathcal{J}_S .

Lemma 1. *The separable regularizer Ψ_S in (20) can be rewritten as*

$$\Psi_S(x; a) = \|G(Lx)\|_1 - \mathcal{H}_S(x; a), \quad (21)$$

where the function \mathcal{H}_S in (21) takes the following equivalent forms:

$$\mathcal{H}_S(x; a) = \sum_{i=1}^s h_{a_i}(g_i(Lx)) \quad (22)$$

$$= \left(\|\cdot\|_1 \boxplus \frac{1}{2} \|W \cdot\|_2^2 \right) (G(Lx)) \quad (23)$$

$$= \text{env}_{\|\cdot\|_{W^{-1}, \|\cdot\|_1}}^1 (WG(Lx)), \quad (24)$$

with h_{a_i} the Huber function defined in (12) and $W \in \mathbb{R}^{s \times s}$ the matrix defined by

$$W := \text{diag}(\sqrt{a_1}, \dots, \sqrt{a_s}). \quad (25)$$

In the special case that $a_i = \bar{a} \quad \forall i = 1, \dots, s$, $\bar{a} \in \mathbb{R}_{++}$, then (23) and (24) reduce to

$$\mathcal{H}_S(x; a) = \text{env}_{\|\cdot\|_1}^{\bar{a}} (G(Lx)). \quad (26)$$

Proof. First, recalling the MC penalty definition in (13), Ψ_S in (20) can be rewritten as

$$\Psi_S(x; a) = \sum_{i=1}^s (|g_i(Lx)| - h_{a_i}(g_i(Lx))) = \|G(Lx)\|_1 - \underbrace{\sum_{i=1}^s h_{a_i}(g_i(Lx))}_{\mathcal{H}_S(x; a)}, \quad (27)$$

which proves (21)–(22). Then, based on the Huber function definition in (12), the function $\mathcal{H}_S(x; a)$ in (27) can be manipulated as follows:

$$\begin{aligned} \mathcal{H}_S(x; a) &= \sum_{i=1}^s \text{env}_{|\cdot|}^{a_i}(g_i(Lx)) \\ &= \sum_{i=1}^s \min_{v_i \in \mathbb{R}} \left\{ |v_i| + \frac{a_i}{2} (g_i(Lx) - v_i)^2 \right\} \\ &= \min_{v \in \mathbb{R}^s} \sum_{i=1}^s \left(|v_i| + \frac{a_i}{2} (g_i(Lx) - v_i)^2 \right) \\ &= \min_{v \in \mathbb{R}^s} \left\{ \sum_{i=1}^s |v_i| + \frac{1}{2} \sum_{i=1}^s \left(\sqrt{a_i} (g_i(Lx) - v_i) \right)^2 \right\} \\ &= \min_{v \in \mathbb{R}^s} \left\{ \|v\|_1 + \frac{1}{2} \|W(G(Lx) - v)\|_2^2 \right\} \end{aligned} \quad (28)$$

$$= \left(\|\cdot\|_1 \boxplus \frac{1}{2} \|W \cdot\|_2^2 \right) (G(Lx)), \quad (29)$$

with matrix W defined in (25). The last equality (29), which proves (23), comes straightforwardly from the definition of infimal convolution in (9).

Starting from (28), and noting that by assumption the square diagonal matrix W in (25) is invertible (in fact, $a_i \in \mathbb{R}_{++} \forall i = 1, \dots, s$), we can write

$$\begin{aligned} \mathcal{H}_S(x; a) &= \min_{v \in \mathbb{R}^s} \left\{ \|v\|_1 + \frac{1}{2} \|WG(Lx) - Wv\|_2^2 \right\} \\ &= \min_{z \in \mathbb{R}^s} \left\{ \|W^{-1}z\|_1 + \frac{1}{2} \|WG(Lx) - z\|_2^2 \right\} \\ &= \text{env}_{\|W^{-1} \cdot\|_1}^1 (WG(Lx)), \end{aligned}$$

which completes the proof of (24). Statement (26) follows easily. \square

In the following result, we define the set of sub-vectors $\{z^{(i)}\}_{i=1}^s$, $z^{(i)} \in \mathbb{R}^{r_i}$, as a *partition* of vector $z \in \mathbb{R}^r$ if $z^{(i)} = P^{(i)}z$, with $P^{(i)} \in \mathbb{R}^{r_i \times r}$ binary selection matrices satisfying $\left((P^{(1)})^T, (P^{(2)})^T, \dots, (P^{(s)})^T \right)^T = P$, with $P \in \mathbb{R}^{r \times r}$ a permutation matrix, so that $\left((z^{(1)})^T, (z^{(2)})^T, \dots, (z^{(s)})^T \right)^T = Pz$, a permuted version of z .

Theorem 1. *If the components $g_i : \mathbb{R}^r \rightarrow \mathbb{R}$ of function G are all lower semicontinuous functions, then for any matrices A, L and any value of parameters $\mu \in \mathbb{R}_{++}$, $a \in \mathbb{R}_{++}^s$, the objective function $\mathcal{J}_S : \mathbb{R}^n \rightarrow \mathbb{R}$ defined in (20) is lower semicontinuous and bounded from below by zero.*

Moreover, if any g_i is either linear or a lower semicontinuous convex and nonnegative function, then a sufficient condition for \mathcal{J}_S to be convex (strongly convex) is that the function

$$\mathcal{J}_1(x) := \|Ax\|_2^2 - \mu \|WG(Lx)\|_2^2 \text{ is convex (strongly convex),} \quad (30)$$

with matrix W defined in (25).

In particular, in the special cases that G is the identity operator or a function defined by

$$G(z) = \left(\|z^{(1)}\|_2, \dots, \|z^{(s)}\|_2 \right)^T, \quad \text{with } \{z^{(i)}\}_{i=1}^s \text{ partition of } z \in \mathbb{R}^r, \quad (31)$$

then it follows from (30) that \mathcal{J}_S is convex (strongly convex) if

$$Q := A^T A - \mu L^T W^2 L \geq 0 \ (\succ 0). \quad (32)$$

Finally, in case that $a_i = \tilde{a} \ \forall i = 1, \dots, s$, (32) reduces to

$$Q = A^T A - \mu \tilde{a} L^T L \geq 0 \ (\succ 0), \quad (33)$$

that is,

$$\tilde{a} = \tau_c \frac{\rho_{A,L}}{\mu}, \quad \tau_c \in [0, 1] \quad \left(\tau_c \in [0, 1[\right), \quad \rho_{A,L} := \frac{\sigma_{A,\min}^2}{\sigma_{L,\max}^2}, \quad (34)$$

with $\sigma_{A,\min}$ and $\sigma_{L,\max}$ denoting the minimum singular value of matrix A and the maximum singular value of matrix L , respectively.

Proof. Since the MC penalty function defined in (13) is continuous and bounded from below by zero, if functions g_i are all lower semicontinuous, then the regularizer Ψ_S and, hence, the total objective function \mathcal{J}_S in (20) are both lower semicontinuous and bounded from below by zero.

In order to derive convexity conditions for \mathcal{J}_S , we first introduce the function $q_a : \mathbb{R} \rightarrow \mathbb{R}_+$ defined by

$$q_a(t) := \frac{a}{2} t^2 + |t| - h_a(t) = \begin{cases} |t| & \text{for } |t| \in [0, 1/a], \\ \frac{a}{2} t^2 + \frac{1}{2a} & \text{for } |t| \in]1/a, +\infty[, \end{cases} \quad (35)$$

where the second equality in (35) comes from the Huber function definition in (12). It is easy to prove that, for any value of the parameter $a \in \mathbb{R}_{++}$, the function q_a in (35) is convex on \mathbb{R} , continuously differentiable on $\mathbb{R} \setminus \{0\}$, and monotonically increasing on \mathbb{R}_+ .

Based on results in Lemma 1, in particular (21)–(22), and on definition of the Huber function in (12), the expression of function \mathcal{J}_S in (20) can be manipulated and equivalently rewritten as follows:

$$\begin{aligned} \mathcal{J}_S(x; a) &= \frac{1}{2} \|Ax - b\|_2^2 + \mu \left(\|G(Lx)\|_1 - \sum_{i=1}^s h_{a_i}(g_i(Lx)) \right) \\ &= \frac{1}{2} \|Ax - b\|_2^2 + \mu \sum_{i=1}^s \left[|g_i(Lx)| - h_{a_i}(g_i(Lx)) \right] \\ &= \frac{1}{2} \|Ax - b\|_2^2 + \mu \sum_{i=1}^s \left[|g_i(Lx)| - h_{a_i}(g_i(Lx)) \right] \end{aligned}$$

$$\begin{aligned}
& \left. + \frac{a_i}{2} (g_i(Lx))^2 - \frac{a_i}{2} (g_i(Lx))^2 \right] \\
&= \frac{1}{2} \|Ax - b\|_2^2 - \frac{\mu}{2} \sum_{i=1}^s a_i (g_i(Lx))^2 + \mu \sum_{i=1}^s q_{a_i} (g_i(Lx)) \\
&= \frac{1}{2} \left(\|Ax - b\|_2^2 - \mu \|WG(Lx)\|_2^2 \right) + \mu \sum_{i=1}^s q_{a_i} (g_i(Lx)) \\
&= \frac{1}{2} \mathcal{J}_1(x) + \underbrace{(1/2)\|b\|_2^2 - b^T Ax}_{\mathcal{J}_2(x)} + \underbrace{\mu \sum_{i=1}^s q_{a_i} (g_i(Lx))}_{\mathcal{J}_3(x)}, \quad (36)
\end{aligned}$$

with function $\mathcal{J}_1(x)$ defined in (30). Function $\mathcal{J}_2(x)$ in (36) is affine; hence it clearly does not affect convexity of the total objective function \mathcal{J}_S . Recalling that, given two convex functions $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ and $f_2 : \mathbb{R} \rightarrow \mathbb{R}$, if f_1 is linear or f_2 is monotonically increasing, then the composite function $f_2 \circ f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, function $\mathcal{J}_3(x)$ in (36) is convex. In fact, since the functions q_{a_i} are all convex on \mathbb{R} and monotonically increasing on \mathbb{R}_+ and, by assumption in the theorem statement, all functions g_i are either linear or lower semicontinuous, convex, and nonnegative, each term of the summation defining \mathcal{J}_3 in (36) is a convex function of x . Finally, since $\mu \in \mathbb{R}_{++}$, it follows that a sufficient condition for \mathcal{J}_S to be convex (strongly convex) is that the term \mathcal{J}_1 in (30) is convex (strongly convex). This proves (30).

If G is the identity operator or G has the form in (31), then we have

$$\mathcal{J}_1(x) = x^T \left(A^T A - \mu L^T W^2 L \right) x, \quad (37)$$

from which convexity condition (32) follows easily.

Finally, condition (33) comes straightforwardly from (32) after recalling the definition of matrix W in (25) and the equivalent condition (34) on \tilde{a} has been proved in Lanza et al. (2017). \square

In order to apply in practice the CNC strategy with separable regularizers, one has to compute the value of the scalar $\rho_{A,L}$ defined in (34), depending on the minimum singular value of the measurement matrix A , $\sigma_{A,\min}$, and on the maximum singular value of the regularization matrix L , $\sigma_{L,\max}$. In many important imaging applications, the values of $\sigma_{A,\min}$ and $\sigma_{L,\max}$ can be obtained by explicit formulas. In a general case where no explicit expressions for $\sigma_{A,\min}$ and $\sigma_{L,\max}$ are available, efficient numerical procedures can be used for their accurate estimation.

The parameter τ_c in (34) is referred to as the *convexity coefficient* of the separable CNC variational model in (19)–(20), as it allows to tune the degree of convexity of the model cost function \mathcal{J}_S . In particular, we notice that for τ_c approaching 0 from above, the separable regularizer Ψ_S tends toward the standard

convex ℓ_1 norm-based sparsity-promoting regularizer $\|G(Lx)\|_1$, whereas for τ_c approaching 1 from below, the regularizer Ψ_S tends to be maximally non-convex (hence, potentially, maximally sparsity-promoting) under the CNC constraint that the total cost function \mathcal{J}_S must be convex.

In Corollary 1 below, we highlight some important properties of the introduced class of separable CNC variational models which hold when the null spaces of the measurement matrix A and the regularization matrix L have trivial intersection. In fact, this is an important case, as it almost always occurs in practical applications.

Corollary 1. *Under the same settings of Theorem 1 with G the identity operator or a function of the form in (31), in case that $\text{null}(A) \cap \text{null}(L) = \{0_n\}$ we have:*

- C1. *Convexity condition (32) can be satisfied (with strict or weak inequality) only if matrix A has full column rank.*
- C2. *If A has full column rank, and condition (32) is satisfied with strict inequality, then the function \mathcal{J}_S in (20) is strongly convex; hence it admits a unique global minimizer.*
- C3. *If A has full column rank, and condition (32) is satisfied with weak inequality, then the function \mathcal{J}_S in (20) is convex and coercive; hence it admits a compact convex set of global minimizers.*

Proof. We prove C1 by contradiction. Let us assume that A has not full column rank, such that $A^T A$ has at least one null eigenvalue. Let v be an eigenvector associated with a null eigenvalue of $A^T A$, and let us consider the restriction $Z(t)$ of the quadratic function $x^T Q x$ – with Q the matrix defined in (32) – to the line tv , $t \in \mathbb{R}$:

$$Z(t) = tv^T Q tv = \cancel{tv^T A^T A tv} - \mu tv^T L^T W^T W L tv = -t^2 \mu \|W L v\|_2^2. \quad (38)$$

Under the considered assumption that $\text{null}(A) \cap \text{null}(L) = \{0_n\}$, Lv is different from the null vector. Then, recalling that W is a positive definite diagonal matrix and that $\mu \in \mathbb{R}_{++}$, we have $\mu \|W L v\|_2^2 > 0$; hence $Z(t)$ is a quadratic concave function. This proves C1. C2 does not need a proof. For what concerns C3, first we notice that when A has full column rank, the quadratic fidelity term in (20) is coercive. Moreover, since the MC penalty defined in (13) is bounded below (by zero) for any $a \in \mathbb{R}_{++}$, then the regularizer Ψ_S in (20) is also bounded below (by zero). This implies that the total function \mathcal{J}_S in (20) is coercive and C3 follows easily. \square

It is an important consequence of statement C1 in Corollary 1 that if the measurement matrix $A \in \mathbb{R}^{m \times n}$ in the considered imaging application is wide, namely, $m < n$ (this is the case of many important applications, ranging from image inpainting to compressed sensing), then the CNC strategy with separable sparsity-inducing regularizers can not be used. This strongly motivated the introduction of

CNC models with non-separable regularizers, which will be illustrated in the next two sections.

Sparsity-Inducing Non-separable Regularizers

As pointed out in previous section, when the measurement matrix A is not full column rank, then a CNC formulation is not possible using a separable sparsity-promoting regularizer. However, in Lanza et al. (2019) and Selesnick et al. (2020), a general strategy to construct parameterized sparsity-promoting non-convex non-separable regularizers has been proposed which allows to tackle also the case of A not being full column rank. This is of great importance, as it enables us to apply the CNC approach to practically any linear inverse problem in imaging.

In accordance with Lanza et al. (2019) and Selesnick et al. (2020), we present a general strategy for constructing non-separable sparsity-promoting regularizers Ψ_{NS} starting from any convex sparsity-promoting regularizer \mathcal{R} and then subtracting its generalized Moreau envelope. In particular, we consider regularizers \mathcal{R} of the form

$$\mathcal{R}(x) := \Theta(y), \quad y = G(Lx), \quad (39)$$

where, coherently with the definitions given in previous sections, $L \in \mathbb{R}^{r \times n}$, $G : \mathbb{R}^r \rightarrow \mathbb{R}^s$ is a possibly nonlinear function, $y \in \mathbb{R}^s$ represents the image features vector to be sparsified, and $\Theta : \mathbb{R}^s \rightarrow \mathbb{R}$ is some function promoting sparsity of its argument. Following Lanza et al. (2019), the introduced regularizer and the matrix $B \in \mathbb{R}^{q \times n}$ – the meaning of which will be clarified later – must satisfy the following assumptions:

- B1. $\mathcal{R} \in \Gamma_0(\mathbb{R}^n)$, bounded below by 0 with $\mathcal{R}(0) = 0$.
- B2. $\Theta(G(\cdot))$ is proper, lower semicontinuous, and coercive.
- B3. B has full row rank and satisfies $\text{null}(B) \cap \text{null}(L) = \{0_n\}$.

The non-separable sparsity-promoting regularizer Ψ_{NS} is defined as follows:

$$\Psi_{\text{NS}}(x; B) := \mathcal{R}(x) - \mathcal{H}_{\text{NS}}(x; B), \quad (40)$$

with

$$\mathcal{H}_{\text{NS}}(x; B) := \left(\mathcal{R} \square \frac{1}{2} \|B \cdot\|_2^2 \right) (x; B) = \min_{v \in \mathbb{R}^n} \left\{ \mathcal{R}(v) + \frac{1}{2} \|B(x - v)\|_2^2 \right\}, \quad (41)$$

where B is a matrix-valued parameter which plays the same role of the parameter vector a in the class of separable regularizers illustrated in section “CNC Models with Sparsity-Inducing *Separable* Regularizers”. Indeed, the introduced class of non-separable regularizers in (40)–(41) can be regarded as a sort of generalization

of the class of separable regularizers defined in (20) and equivalently reformulated in (21), (22), (23), (24). The square diagonal matrix W in (25), containing the square root of the parameter vector a on the main diagonal, is replaced in (40)–(41) by a more general (not necessarily square and diagonal) parameter matrix B , and the term $\|G(Lx)\|_1$ in (21) is substituted by a more general convex function $\mathcal{R}(x) = \Theta(G(Lx))$, according to definitions (39)–(40).

We notice that the introduced regularizer in (40)–(41) can not be written as a function of only the vector to be sparsified $y = G(Lx)$, hence, coherently with Definition 1, is non-separable and takes the general form $\Psi_{\text{NS}}(x; B) = \Phi(x, y)$.

We also note that if $C^T C = B^T B$, then $\mathcal{H}_{\text{NS}}(x; B) = \mathcal{H}_{\text{NS}}(x; C)$ for all $x \in \mathbb{R}^n$. That is, the function $\mathcal{H}_{\text{NS}}(x; B)$ depends only on $B^T B$ and not B itself. Therefore, without loss of generality, we may assume B has full row rank. In fact, if a given matrix B does not have full row rank, then there is another matrix C with full row rank such that $C^T C = B^T B$ which yields the same function $\mathcal{H}_{\text{NS}}(x; B)$.

In the sequel, we outline some properties of function $\mathcal{H}_{\text{NS}}(x; B)$, proved in Lanza et al. (2019).

Proposition 3. *The function $\mathcal{H}_{\text{NS}}(x; B)$ in (41) exhibits the following properties:*

1. *For any matrix B , $\mathcal{H}_{\text{NS}}(x; B)$ is proper, continuous, and convex and satisfies*

$$0 \leq \mathcal{H}_{\text{NS}}(x; B) \leq \mathcal{R}(x), \quad \forall x \in \mathbb{R}^n, \quad (42)$$

$$\mathcal{H}_{\text{NS}}(x; B) \leq \mathcal{H}_{\text{NS}}(x; \alpha I_n), \quad \forall x \in \mathbb{R}^n, \forall \alpha \geq \|B\|_2. \quad (43)$$

2. *For any full row rank matrix B , $\mathcal{H}_{\text{NS}}(x; B)$ is a differentiable function, with gradient given by*

$$\nabla \mathcal{H}_{\text{NS}}(x; B) = B^T B \left(x - \arg \min_{v \in \mathbb{R}^n} \left\{ \frac{1}{2} \|B(x - v)\|_2^2 + \mathcal{R}(v) \right\} \right). \quad (44)$$

Moreover, $\mathcal{H}_{\text{NS}}(x; B)$ can be expressed in terms of a Moreau envelope as

$$\mathcal{H}_{\text{NS}}(x; B) = \left(\text{env}_{d \circ B^+}^1 \circ B \right) (x), \quad (45)$$

where $d: \mathbb{R}^n \rightarrow \mathbb{R}$ is the convex function

$$d(x) = \min_{w \in \text{null}(B)} \mathcal{R}(x - w). \quad (46)$$

By the way of illustration, in Fig. 4 we show a simple example of non-separable non-convex regularizer $\Psi_{\text{NS}}(x; B)$ (third column) obtained – in accordance with the definition in (40)–(41) – by subtracting from the convex regularizer $\mathcal{R}(x) = \|x\|_1$ (first column) its generalized Moreau envelope $\mathcal{H}_{\text{NS}}(x; B)$ (second column), for a

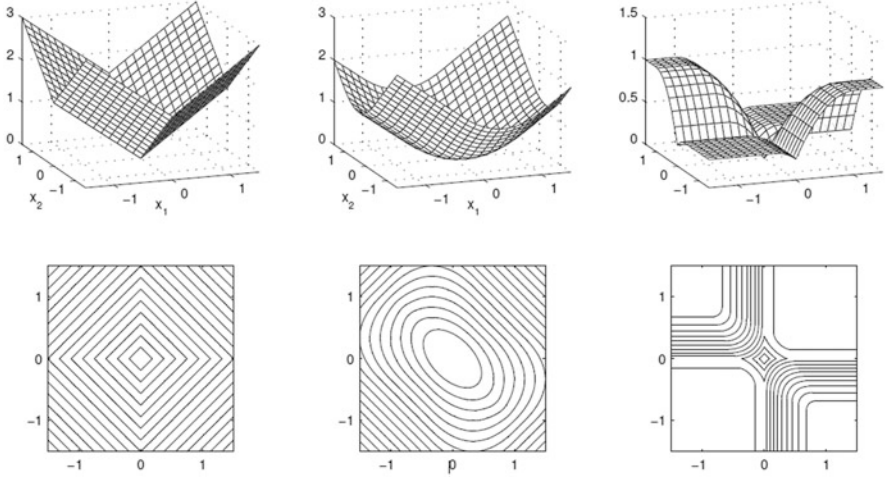


Fig. 4 Example of construction of a non-separable regularizer of the form in (40)–(41) with parameter matrix B defined in (47): $\mathcal{R}(x) = \|x\|_1$ (first column), $\mathcal{H}_{\text{NS}}(x; B)$ (second column), and $\Psi_{\text{NS}}(x; B) = \mathcal{R}(x) - \mathcal{H}_{\text{NS}}(x; B)$ (third column); the associated contour plots are shown in the bottom row

vector $x \in \mathbb{R}^2$, and a (rectangular) parameter matrix $B \in \mathbb{R}^{3 \times 2}$ defined as

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}. \quad (47)$$

CNC Models with Sparsity-Inducing *Non-separable* Regularizers

This section is concerned with the formulation of CNC variational models containing *non-separable* sparsity-promoting regularizers (see Definition 1) having the form introduced in (40)–(41). The considered non-separable CNC models thus read

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \mathcal{J}_{\text{NS}}(x; B), \quad (48)$$

$$\mathcal{J}_{\text{NS}}(x; B) := \frac{1}{2} \|Ax - b\|_2^2 + \mu \Psi_{\text{NS}}(x; B), \quad \Psi_{\text{NS}}(x; B) := \mathcal{R}(x) - \mathcal{H}_{\text{NS}}(x; B), \quad (49)$$

with function \mathcal{H}_{NS} defined in (41) and the matrix B and the regularizer \mathcal{R} satisfying assumptions B1–B3 outlined in the previous section. We refer to (48)–(49) as the class of CNC non-separable (least-squares) models, abbreviated CNC-NS-L₂.

In Theorem 2, we give conditions on the parameter matrix B of the regularizer Ψ_{NS} in order to guarantee convexity (strong convexity) of the total cost function \mathcal{J}_{NS} in (48)–(49); then in Corollary 2 we discuss existence and uniqueness of its minimizer(s), that is, of the solution(s) x^* of the introduced class of CNC-NS- L_2 variational models.

Theorem 2. *Let \mathcal{R} and B satisfy assumptions B1–B3, and let Ψ_{NS} be the function defined in (49) with \mathcal{H}_{NS} given in (41). Then, the function \mathcal{J}_{NS} in (49) is proper, lower semicontinuous, and bounded below by zero. Moreover, a sufficient condition for \mathcal{J}_{NS} to be convex (strongly convex) is that the matrix of parameters B satisfies*

$$Q := A^T A - \mu B^T B \succeq 0 \ (\succ 0). \quad (50)$$

Corollary 2. *Under the same assumptions of Theorem 2, if function \mathcal{J}_{NS} in (49) is strongly convex, then it admits a unique global minimizer. If, instead, \mathcal{J}_{NS} is only convex, with Q weakly satisfying (50), and $\text{null}(A) \cap \text{null}(L) = \{0_n\}$, then \mathcal{J}_{NS} is coercive; hence it admits compact convex set of global minimizers.*

The proofs of Theorem 2 and Corollary 2 are reported in Lanza et al. (2019).

Remark 1. All the previous derivations are valid for any function $\Theta : \mathbb{R}^r \rightarrow \mathbb{R}$ in the definition of the convex regularizer \mathcal{R} in (39), provided that assumptions B1–B3 are satisfied. However, since $\mathcal{R} = \Theta(G(L \cdot))$ must be a convex regularizer inducing (as effectively as possible) sparsity of the features vector $y = G(Lx)$, then it is very reasonable to consider convex, sparsity-promoting, additively separable functions Θ of the form

$$\Theta(y) = \sum_{i=1}^s \theta(y_i), \quad (51)$$

with $\theta : \mathbb{R} \rightarrow \mathbb{R}_+$ even, continuous, convex, monotonically increasing on \mathbb{R}_+ and such that $\theta(0) = 0$. In particular, one of the best (and most natural) choices is to consider $\Theta = \|\cdot\|_1$, corresponding to $\theta = |\cdot|$. If one aims at avoiding non-differentiability (which is not the case in this work), a good alternative is to consider as θ the Huber function in place of the absolute value function.

Construction of Matrix B

Convexity condition (50) for the cost function \mathcal{J}_{NS} in (49) sets an inequality constraint on $B^T B$, hence on the matrix B of free parameters in the non-separable regularizer Ψ_{NS} . In the sequel, we illustrate a few simple strategies for choosing B .

The first and simplest strategy consists in setting $B = \sqrt{\gamma/\mu} A$, that is,

$$B^T B = \frac{\gamma}{\mu} A^T A, \quad \gamma \in [0, 1], \quad (52)$$

which clearly fulfills condition (50). We notice that, analogously to τ_c in (34) for the CNC separable models, the scalar parameter γ in (52) controls the degree of non-convexity of the non-separable regularization term Ψ_{NS} , hence the degree of convexity of the total objective \mathcal{J}_{NS} : the greater the γ , the more non-convex the Ψ_{NS} and, hence, the less convex the \mathcal{J}_{NS} . In particular, for γ approaching 0 from above, B tends to the null matrix, and hence, the non-separable regularizer Ψ_{NS} tends to the convex regularizer \mathcal{R} . On the other side, for γ approaching 1 from below, the regularizer Ψ_{NS} tends to be maximally non-convex (hence, potentially, maximally sparsity-promoting) under the CNC constraint that the total cost function \mathcal{J}_{NS} must be convex.

A more sophisticated and flexible strategy for constructing a matrix $B^T B$ satisfying convexity condition (50) can be derived by considering the eigenvalue decomposition of the symmetric, positive semidefinite matrix $A^T A$,

$$A^T A = V E V^T, \quad E, V \in \mathbb{R}^{n \times n}, \quad E = \text{diag}(e_1, \dots, e_n), \quad V^T V = V V^T = I_n, \quad (53)$$

with $e_i, i = 1, \dots, n$, indicating the real non-negative eigenvalues of $A^T A$. We set

$$B^T B = \frac{1}{\mu} V \Gamma E V^T, \quad \Gamma := \text{diag}(\gamma_1, \dots, \gamma_n), \quad \gamma_i \in [0, 1] \forall i \in \{1, \dots, n\}, \quad (54)$$

so that, replacing (54) into convexity condition (50), we have

$$Q = V (E - \Gamma E) V^T \geq 0 (> 0) \iff E (I_n - \Gamma) \geq 0 (> 0), \quad (55)$$

which is clearly satisfied given the definition of matrix Γ in (54). We notice that when one chooses $\gamma_1 = \gamma_2 = \dots = \gamma_n = \gamma \in [0, 1]$, then (54) reduces to (52), that is, strategy (52) is included in the more general strategy (54).

Finally, in Park and Burrus (1987) another method for prescribing the matrix $B^T B$, hence B , for the specific purpose of image processing with TV regularization has been proposed. In particular, the diagonal matrix Γ in (54) is set to represent a two-dimensional dc-notch filter (a type of band stop filter) defined by $\Gamma := I - H$, where H is a two-dimensional low-pass filter with a dc-gain of unity and $H \leq I$. A simple choice for H is $H = H_0^T H_0$ with H_0 a moving-average filter having square support. Hence, H is a row-column separable two-dimensional filter given by convolution with a triangle sequence (Park and Burrus 1987).

A Simple CNC Example

In this section, we provide some visual insights on the properties of the considered non-convex separable and non-separable sparsity-promoting regularizers, Ψ_S and Ψ_{NS} , respectively defined in (20) and (40). To this aim, we consider the three two-dimensional variational models defined by minimizing the cost functions

$$\mathcal{J}_R(x) := \frac{1}{2} \|Ax - b\|_2^2 + \mu \mathcal{R}(x), \quad \mathcal{R}(x) = \|Lx\|_1, \quad (56)$$

$$\mathcal{J}_S(x; a) := \frac{1}{2} \|Ax - b\|_2^2 + \mu \Psi_S(x; a), \quad \Psi_S(x; a) = \mathcal{R}(x) - \mathcal{H}_S(x; a), \quad (57)$$

$$\mathcal{J}_{NS}(x; B) := \frac{1}{2} \|Ax - b\|_2^2 + \mu \Psi_{NS}(x; B), \quad \Psi_{NS}(x; B) = \mathcal{R}(x) - \mathcal{H}_{NS}(x; B), \quad (58)$$

where (56) represents the model containing the baseline convex ℓ_1 norm-based sparsity-inducing regularizer, the functions \mathcal{H}_S in (57) and \mathcal{H}_{NS} in (58) are defined in (24) and (41), respectively, and we set

$$\mu = 1.5, \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0.4 & 1.5 \\ -1.0 & 0.8 \end{bmatrix}, \quad L = \begin{bmatrix} -2.0 & -1.0 \\ 0.5 & -2.5 \end{bmatrix}. \quad (59)$$

Moreover, according to the convexity conditions in (34) and (52), for the CNC separable and non-separable models in (57) and (58), we choose

$$a_1 = a_2 = \bar{a} = \tau_c \frac{\rho_{A,L}}{\mu}, \quad \tau_c = 0.99, \quad (60)$$

$$B = \sqrt{\frac{\gamma}{\mu}} A, \quad \gamma = 0.99, \quad (61)$$

respectively, so that both the CNC models are pushed toward their convexity limit.

In Fig. 5, we show the regularizer \mathcal{R} and total cost function \mathcal{J}_R of the baseline convex model (56), in Fig. 6 the regularizer Ψ_S and total cost function \mathcal{J}_S of the separable CNC model (57), and in Fig. 7 the regularizer Ψ_{NS} and total cost function \mathcal{J}_{NS} of the non-separable CNC model (58). All function graphs are accompanied, in the bottom row, by their associated contour plots. The solid red and blue lines in the contour plot figures represent the hyperplanes Y_1 and Y_2 , respectively, with $Y_i := \{x \in \mathbb{R}^2 : L_i x = 0\}$, $i \in \{1, 2\}$, and L_i the i -th row of matrix L .

From the left columns of Figs. 5, 6, and 7, it can be noticed that the baseline regularizer $\mathcal{R}(x)$ is clearly convex, but not strictly convex, whereas the separable and non-separable regularizers $\Psi_S(x; a)$ and $\Psi_{NS}(x; B)$ are non-convex. In fact, according to their definitions in (57) and (58), they are obtained by subtracting

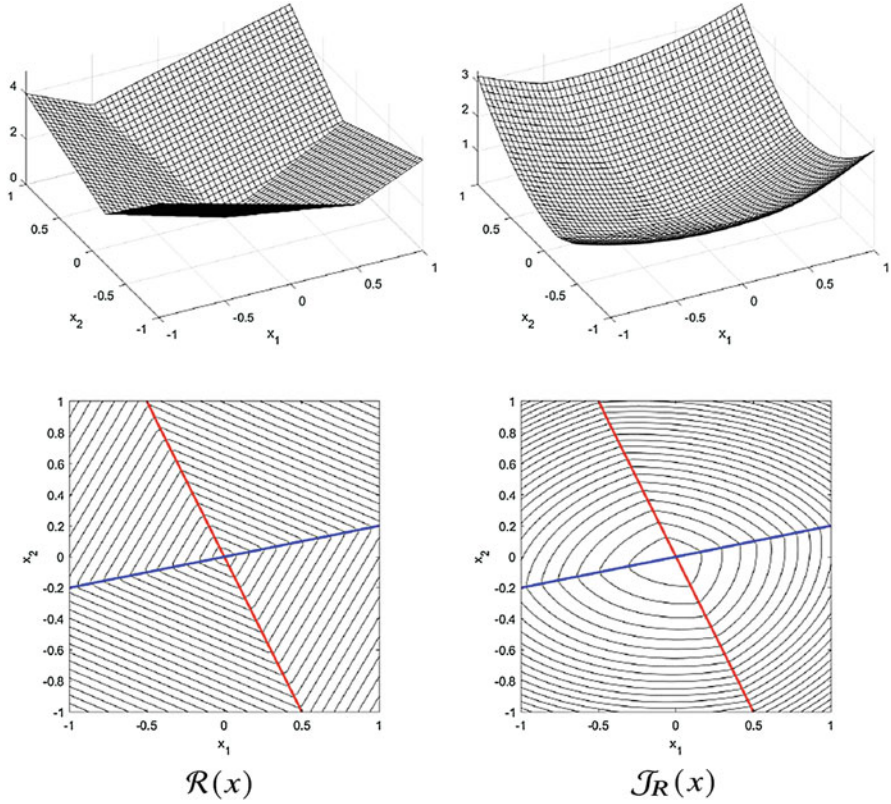


Fig. 5 Graphs of functions $\mathcal{R}(x)$ and $\mathcal{J}_R(x)$ in (56) with associated contour plots

from $\mathcal{R}(x)$ the convex terms $\mathcal{H}_S(x; a)$ and $\mathcal{H}_{NS}(x; B)$, respectively. The non-convex regularizers $\Psi_S(x; a)$ and $\Psi_{NS}(x; B)$ thus hold the potential for promoting sparsity of the vector $Lx = (L_1x, L_2x)^T$ more effectively than the convex regularizer $\mathcal{R}(x)$.

The plots in the right columns of Figs. 5, 6, and 7 confirm, first, that the total cost function $\mathcal{J}_R(x)$ is clearly convex and then, more interestingly, that the cost functions $\mathcal{J}_S(x; a)$ and $\mathcal{J}_{NS}(x; B)$ of the separable and non-separable CNC models in (57) and (58) are also both convex, as prescribed by the CNC rationale and as expected due to our settings $\tau_c = \gamma = 0.99 < 1$.

As a final interesting experiment, we push both the separable and non-separable CNC models in (57), (58) outside their guaranteed convexity regimes, as defined by sufficient conditions (34), (52), respectively. More precisely, we set $\tau_c, \gamma > 1$ in (60), (61), thus obtaining the total cost functions $\mathcal{J}_S(x; a)$, $\mathcal{J}_{NS}(x; B)$ depicted in Fig. 8. It can be noticed from the graphs in the top row and, more clearly, from the associated contour plots in the bottom row that both the cost functions are non-convex, as expected from theory.

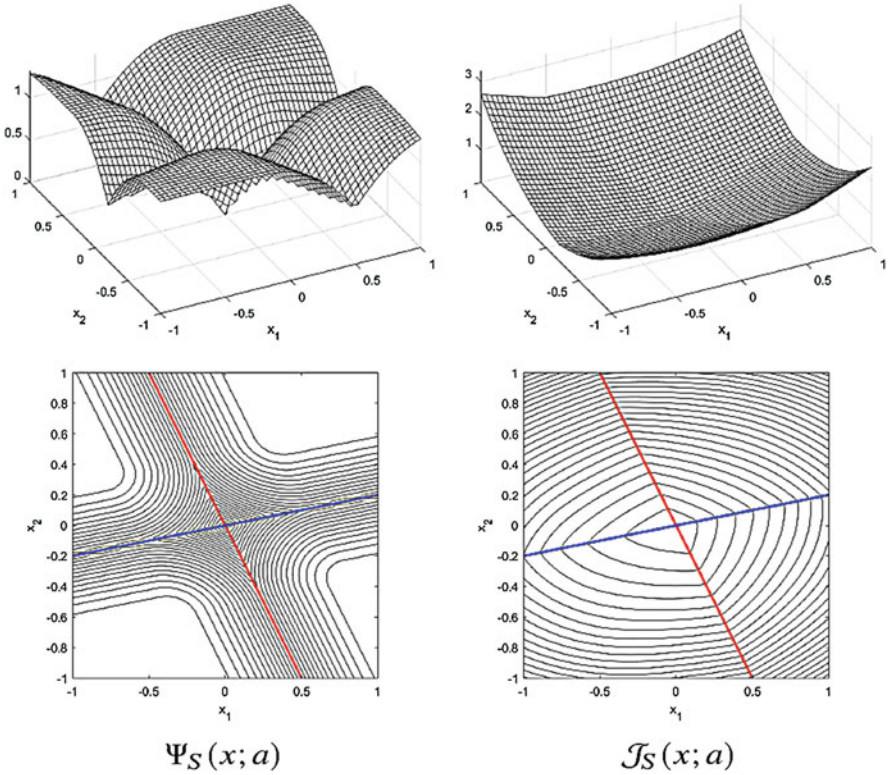


Fig. 6 Graphs of functions $\Psi_S(x; a)$ and $\mathcal{J}_S(x; a)$ in (57) with associated contour plots

Path of Solution Components

The different behavior of standard ℓ_1 norm convex regularization versus its associated non-convex non-separable regularization can be illustrated by observing the solution path as the regularization parameter μ varies. In particular, we denote by x_{L_1} the solution of the minimization problem (56) with $L = I$ and by x_{NS} the solution of its associated non-separable CNC model (58). When μ is sufficiently large, both the solutions x_{L_1} and x_{NS} will be the all-zero vector. When μ is sufficiently close to zero, the solution using either regularizations will approximate the unconstrained least-squares solution. However, as μ varies between these two extremes, the solutions obtained using the two regularization methods will sweep different paths. This is illustrated in Fig. 2.1 in Hastie et al. (2015) which concerns an example of least-squares problem with ℓ_1 norm regularization where matrix A is of size 50×5 . This example is reproduced in Fig. 9. As in Hastie et al. (2015), the solution path is shown as a function of the fraction: the ℓ_1 norm of x_{L_1} divided by the ℓ_1 norm of the unconstrained (unregularized) least-squares solution x_{LS} ; this fraction varies between zero and one. Repeating the same example using non-

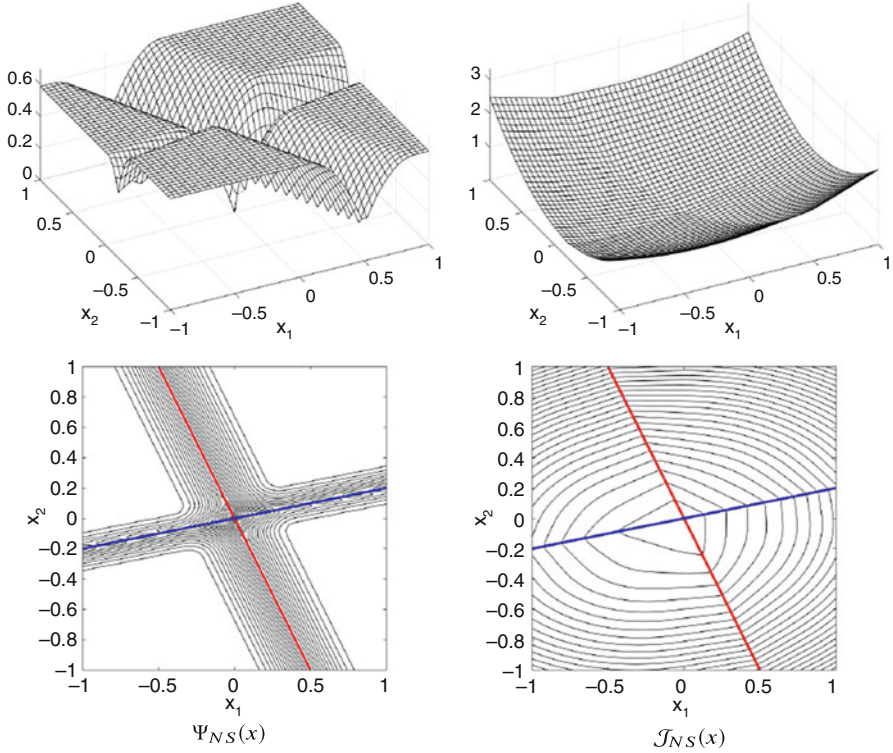


Fig. 7 Graphs of functions $\Psi_{NS}(x; B)$ and $\mathcal{J}_{NS}(x; B)$ in (58) with associated contour plots

separable non-convex regularization in (58) instead of ℓ_1 norm regularization, we obtain a different solution path for x_{NS} , as shown in Fig. 9. It can be seen that the x_{NS} solution is more sparse than the ℓ_1 norm solution x_{L_1} for most of the solution component path. The x_{NS} solution starts to have two non-zero components when the x_{L_1} solution already has three non-zero components. It can also be seen that along most of the solution path, non-zero components of the x_{NS} solution are greater in absolute value than those of the x_{L_1} solution. The solution paths show that components of the x_{NS} solution become non-zero later (along this axis) than components of the x_{L_1} solution.

Forward-Backward Minimization Algorithms

In this section, we introduce optimization algorithms for the numerical solution of the illustrated separable and non-separable CNC variational models, based on the iterative FB strategy within the general framework of splitting, commonly used when the objective function is the sum of two convex but not necessarily differentiable functions. This iterative method, proposed in Beck and Teboulle

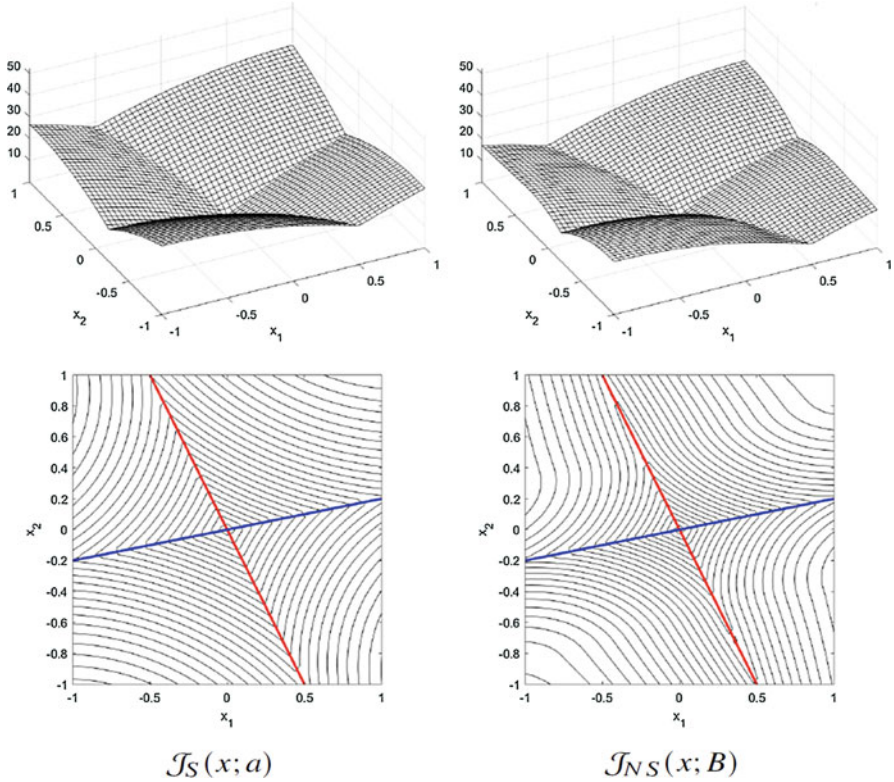


Fig. 8 Graphs of the total cost functions $\mathcal{J}_S(x; a)$, $\mathcal{J}_{NS}(x; B)$ and associated contour plots for the separable and non-separable variational models in (57), (58) pushed beyond their convexity limit, that is, for $\tau_c, \gamma > 1$

(2009), has attracted extensive interests due to its simplicity and several important advantages. It is well-known that this method uses little storage, readily exploits the separable structure of the minimization problem, and is easily implemented to practical applications. It relies on a forward gradient step (an explicit step) followed by a backward proximal step (an implicit step).

In the separable case (section “[FB Strategy for Separable CNC Models](#)”), it reduces to a standard proximal gradient or subgradient splitting minimization algorithm. In the non-separable case (section “[FB Strategy for Non-separable CNC Models](#)”), a more general form of the FB algorithm aimed to solve monotone inclusion problems is used. The solution of the minimization problems in the backward steps of the FB applied to both the separable and non-separable cases relies on a very efficient ADMM strategy (section “[Efficient Solution of the Backward Steps by ADMM](#)”).

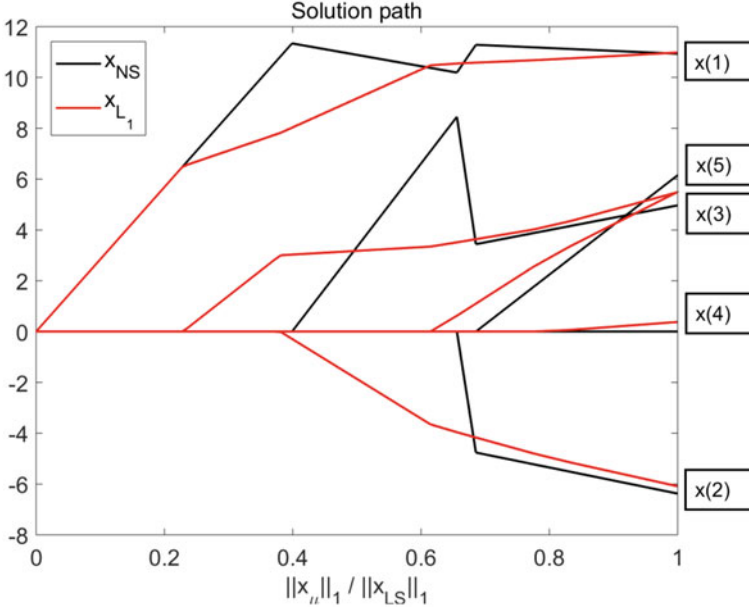


Fig. 9 Path of the five solution components for the regularized least squares example in Hastie et al. (2015); $\|x_\mu\| / \|x_{LS}\|$ is the red colored path for $x_\mu = x_{L_1}$ and the black colored path for $x_\mu = x_{NS}$, for increasing values of the regularization parameter μ

FB Strategy for Separable CNC Models

Based on Lemma 1, in particular expression (21) for the separable sparsity-promoting regularizer Ψ_S , with function \mathcal{H}_S in the forms (22) and (24), the class of considered separable CNC variational models defined in (19)–(20) can be equivalently rewritten in the following equivalent form:

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \mathcal{J}_S(x; a), \quad (62)$$

$$\mathcal{J}_S(x; a) = \frac{1}{2} \|Ax - b\|_2^2 - \mu \sum_{i=1}^s h_{a_i}(g_i(Lx)) + \mu \|G(Lx)\|_1 \quad (63)$$

$$= \underbrace{\frac{1}{2} \|Ax - b\|_2^2 - \mu \operatorname{env}_{\|W^{-1}\|_1}^1(WG(Lx))}_{\mathcal{J}_1(x; a)} + \underbrace{\mu \|G(Lx)\|_1}_{\mathcal{J}_2(x)}. \quad (64)$$

Based on results in Theorem 1, first we notice that if convexity condition (30) is satisfied – which is the case of interest for us – both the total objective \mathcal{J}_S and the two terms \mathcal{J}_1 and \mathcal{J}_2 in (64) are proper, lower semicontinuous, and convex

functions. Then, the term \mathcal{J}_2 is in general – i.e., for the great majority of reasonable functions G – a non-differentiable function, whereas \mathcal{J}_1 can be differentiable or non-differentiable depending on G . Indeed, some popular regularizers are defined in terms of G functions yielding differentiability of \mathcal{J}_1 , as it will be illustrated in Proposition 4.

Hence, we propose to compute approximate solutions x^* of the CNC separable model in (62), (63), and (64) by means of the FB iterative scheme outlined in Proposition 5. The forward step consists of a subgradient – or gradient, depending on G – descent step of the term \mathcal{J}_1 , whereas the backward step is a proximal step of \mathcal{J}_2 . In Proposition 4, we preliminarily derive the expression of the subgradient – or gradient – of the function \mathcal{J}_1 .

Proposition 4. *Let $\mathcal{J}_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ be the function defined in (64), and let the convexity conditions (30) for \mathcal{J}_S be satisfied. Then, in the general case of a possibly non-differentiable function G , the subdifferential $\partial\mathcal{J}_1 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ takes the form*

$$\begin{aligned} \partial\mathcal{J}_1(x; a) = & A^T(Ax - b) \\ & - \mu L^T \partial G(Lx) W \left(WG(Lx) - \text{prox}_{\|W^{-1}\cdot\|_1} (WG(Lx)) \right), \end{aligned} \quad (65)$$

with $\partial\mathcal{J}_1$ and ∂G replaced by $\nabla\mathcal{J}_1$ and ∇G if G is differentiable.

In the special case that G is a non-differentiable function of the form in (31) with the partition of vector $z = Lx$ defined by a permutation matrix $P = \left((P^{(1)})^T, \dots, (P^{(s)})^T \right)^T \in \mathbb{R}^{r \times r}$, $P^{(i)} \in \mathbb{R}^{r_i \times r}$, $i = 1, \dots, s$, then the function \mathcal{J}_1 in (64) is differentiable with gradient $\nabla\mathcal{J}_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by

$$\nabla\mathcal{J}_1(x; a) = A^T(Ax - b) - \mu L^T P^T C(Lx) P Lx, \quad C = \text{diag} \left(C^{(1)}, \dots, C^{(s)} \right), \quad (66)$$

where $C : \mathbb{R}^r \rightarrow \mathbb{R}^{r \times r}$ is a block-diagonal matrix-valued function with scalar diagonal blocks defined by

$$C^{(i)}(z) = \min \left\{ a_i, \frac{1}{\|P^{(i)}z\|_2} \right\} I_{r_i}, \quad i = 1, \dots, s. \quad (67)$$

Proof. The quadratic term in \mathcal{J}_1 – namely, the data fidelity term – is clearly differentiable with gradient given by $A^T(Ax - b)$. Recalling that the Moreau envelope is a differentiable function (see Proposition 1), the second term in \mathcal{J}_1 is differentiable if the function G is differentiable. In fact, in this case the term is composition of differentiable functions. If G is non-differentiable, then the term can be non-differentiable or, for some special G , also differentiable.

In the general case of a possibly non-differentiable function G , expression (65) for the subdifferential of \mathcal{J}_1 comes from applying the chain rule of differentiation to

the calculus of the subdifferential of function \mathcal{J}_1 in the form (64) and from recalling the expression of the gradient of the Moreau envelope function given in (14).

To demonstrate (66)–(67), first we notice that if G has the form in (31), we can write:

$$\mathcal{H}_S(x; a) = \sum_{i=1}^s h_{a_i}(g_i(Lx)) = \sum_{i=1}^s h_{a_i}(\|z^{(i)}\|_2) = \sum_{i=1}^s f_{a_i}(P^{(i)}z), \quad z = Lx,$$

with f_a the function defined in (17). Hence, we have

$$\mathcal{H}_S(x; a) = H(Lx; a), \quad \text{with } H(z; a) := \sum_{i=1}^s f_{a_i}(P^{(i)}z). \quad (68)$$

It follows from Proposition 2 that the function $H(z; a)$ above is differentiable (sum of differentiable functions) with gradient given by

$$\begin{aligned} \nabla_z H(z) &= \sum_{i=1}^s \left[(P^{(i)})^T \nabla_z f_{a_i}(P^{(i)}z) \right] \\ &= \sum_{i=1}^s \left((P^{(i)})^T \min \left\{ a_i, 1/\|P^{(i)}z\|_2 \right\} P^{(i)}z \right) \\ &= \left(\sum_{i=1}^s \left((P^{(i)})^T \min \left\{ a_i, 1/\|P^{(i)}z\|_2 \right\} P^{(i)} \right) \right) z \\ &= P^T C(z) P z, \end{aligned}$$

with C the diagonal matrix-valued function defined in (66)–(67). The function $\mathcal{H}_S(x; a)$ in (68) is thus differentiable with gradient given by

$$\nabla_x \mathcal{H}_S(x; a) = L^T \nabla_z H(Lx; a) = L^T P^T C(Lx) P L x.$$

Recalling the definition of function \mathcal{J}_1 in (63), it is thus clear that it is a differentiable function with gradient given in (66)–(67). \square

Proposition 5. *Let $\mathcal{J}_S(x; a) : \mathbb{R}^n \rightarrow \mathbb{R}$ be the function defined in (62), (63), and (64), with parameters $a \in \mathbb{R}_{++}^s$ satisfying convexity condition in (30). Then, a global minimizer x^* of \mathcal{J}_S can be obtained as the limit point of the sequence of iterates $\{x^{(k)}\}_{k=1}^\infty$ generated by the following FB iterative scheme:*

for $k = 0, 1, 2, \dots$

$$\omega^{(k)} \in \partial \mathcal{J}_1(x^{(k)})$$

$$w^{(k)} = x^{(k)} - \lambda^{(k)} \omega^{(k)}$$

$$x^{(k+1)} = \operatorname{prox}_{\mathcal{J}_2}^{1/\lambda^{(k)}}(w^{(k)}) = \arg \min_{x \in \mathbb{R}^n} \left\{ \|G(Lx)\|_1 + \frac{1}{2\lambda^{(k)}\mu} \|x - w^{(k)}\|_2^2 \right\}$$

end

where the variable stepsizes $\lambda^{(k)}$ are chosen according to the strategy in Bello Cruz (2017) if \mathcal{J}_1 is non-differentiable, or $\lambda^{(k)} = \lambda \in]0, 2/\rho[$ with ρ the Lipschitz constant of the gradient of \mathcal{J}_1 , if \mathcal{J}_1 is differentiable.

For a generic non-differentiable G function, (62), (63), and (64) is a non-smooth convex optimization problem with an objective function which is the sum of two non-differentiable convex functions, \mathcal{J}_1 and \mathcal{J}_2 . In this case, the proximal FB splitting iteration in Proposition 5 – in particular, the computation of $\omega^{(k)}$ in the forward step – relies on the subdifferential (65). For the convergence of this particular FB case, we refer the reader to Bello Cruz (2017).

In case that G is a differentiable function (e.g., G is the identity function) or a non-differentiable function of the special form in (31), the proximal FB splitting iteration in Proposition 5 uses the gradient given in (66). Therefore, the convergence follows the classical results in Beck and Teboulle (2009).

FB Strategy for Non-separable CNC Models

Even though the proposed class of non-separable regularization functions Ψ_{NS} in (49) does not have a simple explicit formula, a global minimizer of the total sparse-regularized cost function \mathcal{J}_{NS} in (49) can be readily calculated using proximal algorithms.

As described in Lanza et al. (2019), in order to compute the solution x^* of the minimization problem in (48)–(49) by using proximal algorithms, it is useful to rewrite it as an equivalent saddle-point problem:

$$\{x^*, v^*\} = \arg \min_{x \in \mathbb{R}^n} \max_{v \in \mathbb{R}^n} \mathcal{F}(x, v; B), \quad (69)$$

$$\mathcal{F}(x, v; B) = \frac{1}{2} \|Ax - b\|_2^2 + \mu \mathcal{R}(x) - \mu \mathcal{R}(v) - \frac{\mu}{2} \|B(x - v)\|_2^2, \quad (70)$$

where, we recall, the regularization function $\mathcal{R}(x) = \Theta(G(Lx))$ and the parameter matrix B satisfy assumptions B1–B3 outlined at the beginning of section “Sparsity-Inducing Non-separable Regularizers”.

The solution of the saddle-point problem above can be calculated using a general form of the FB algorithm (Theorem 25.8 in Bauschke and Combettes 2011). This form of the FB algorithm is formulated to solve the general class of monotone inclusion problems, of which the saddle-point problem (69)–(70) is a special case.

The algorithm, which we will refer to as Primal-Dual FB (PDFB) (as in Lanza et al. (2019)), is outlined in Proposition 6. It involves operators A , A^T , B , and B^T and the proximity operator of the regularization term \mathcal{R} .

Proposition 6. *Let $\mathcal{F}(x, v; B) : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ be the function defined in (70) with the parameters matrix B set as in (53)–(54). Then, a saddle-point $\{x^*, v^*\}$ of \mathcal{F} can be obtained as the limit point of the sequence of iterates $\{x^{(k)}, v^{(k)}\}_{k=1}^{\infty}$ generated by the following PDFB iterative scheme:*

$$\text{set } \rho = \max_i \left\{ \frac{1 - 2\gamma_i + 2\gamma_i^2}{1 - \gamma_i} e_i \right\}$$

$$\text{set } \lambda \in]0, 2/\rho [$$

$$\text{for } k = 0, 1, 2, \dots$$

$$w^{(k)} = x^{(k)} - \lambda \left[A^T (Ax^{(k)} - b) + \mu B^T B (v^{(k)} - x^{(k)}) \right]$$

$$u^{(k)} = v^{(k)} - \lambda \mu B^T B (v^{(k)} - x^{(k)})$$

$$x^{(k+1)} = \arg \min_{x \in \mathbb{R}^n} \left\{ \mathcal{R}(x) + \frac{1}{2\lambda\mu} \|x - w^{(k)}\|_2^2 \right\}$$

$$v^{(k+1)} = \arg \min_{v \in \mathbb{R}^n} \left\{ \mathcal{R}(v) + \frac{1}{2\lambda\mu} \|v - u^{(k)}\|_2^2 \right\}$$

end

where e_i and γ_i are defined in (53)–(54) and k is the iteration counter.

Efficient Solution of the Backward Steps by ADMM

The backward steps in the FB and PDFB algorithms outlined in Propositions 5 and 6 for the numerical solution of the separable and non-separable CNC variational models illustrated in sections “CNC Models with Sparsity-Inducing Separable Regularizers” and “CNC Models with Sparsity-Inducing Non-separable Regularizers”, respectively, all consist of solving the same class of minimization problems, which, in the general case, does not admit a closed-form solution. More precisely, the computations of $x^{(k+1)}$ in the FB algorithm in Proposition 5 and of $x^{(k+1)}$ and $v^{(k+1)}$ in the PDFB algorithm in Proposition 6 all correspond to calculating the proximal operator of a regularization function $\mathcal{R} : \mathbb{R}^n \rightarrow \mathbb{R}$ of the form $\mathcal{R} = \Upsilon(G(L \cdot))$ with proximity parameter $\alpha := 1/(\lambda\mu) \in \mathbb{R}_{++}$ at a point $p \in \mathbb{R}^n$ (equal to $w^{(k)}$ for $x^{(k+1)}$ and to $u^{(k)}$ for $v^{(k+1)}$). We have thus to solve the following minimization problem:

$$\begin{aligned}
t^* &= \text{prox}_{\mathcal{R}}^{\alpha}(p) = \arg \min_{t \in \mathbb{R}^n} \left\{ \mathcal{R}(t) + \frac{\alpha}{2} \|t - p\|_2^2 \right\} \\
&= \arg \min_{t \in \mathbb{R}^n} \left\{ \Upsilon(G(Lt)) + \frac{\alpha}{2} \|t - p\|_2^2 \right\}. \tag{71}
\end{aligned}$$

For both the FB and PDFB cases, the matrix L and the function G – hence, the image features vector $y = G(L \cdot)$ to be sparsified – are defined as in section “**Introduction**”, whereas the function $\Upsilon : \mathbb{R}^s \rightarrow \mathbb{R}$ is to the ℓ_1 norm function $\|\cdot\|_1$ for FB and the function Θ for PDFB. In both cases, it follows from the considered convexity assumptions/conditions that the regularizer $\mathcal{R} = \Upsilon(G(L \cdot))$ is convex; hence the cost function in (71) is strongly convex and admits a unique (global) minimizer t^* .

As it will be later discussed, in most cases of practical interest the function $\Upsilon(G(\cdot))$ is easily proximable, that is, its proximity operator admits a closed form expression or can be calculated very efficiently. Hence, we suggest to solve the minimization problem in (71) by means of the following ADMM-based approach.

First, we rewrite (71) in the equivalent linearly constrained form:

$$\{t^*, z^*\} = \arg \min_{t, z} \left\{ \Upsilon(G(z)) + \frac{\alpha}{2} \|t - p\|_2^2 \right\} \text{ s.t. : } z = Lt, \tag{72}$$

where $z \in \mathbb{R}^r$ is an auxiliary variable (the notation has been chosen for coherence with definition in (6)). Then, we introduce the augmented Lagrangian function,

$$\mathcal{L}(t, z, \rho) = \Upsilon(G(z)) + \frac{\alpha}{2} \|t - p\|_2^2 - \langle \rho, z - Lt \rangle + \frac{\beta}{2} \|z - Lt\|_2^2, \tag{73}$$

where $\beta > 0$ is a scalar penalty parameter and $\rho \in \mathbb{R}^r$ is the dual variable, i.e., the vector of Lagrange multipliers associated with the set of r linear constraints in (72). Solving (72) is tantamount to seek for the saddle point of the augmented Lagrangian function in (73). The saddle-point problem reads as follows:

$$\{t^*, z^*\} = \arg \min_{t, z} \max_{\rho} \mathcal{L}(t, z, \rho). \tag{74}$$

Upon suitable initialization, and for any $j = 0, 1, 2, \dots$, the j -th iteration of the ADMM applied to solving the saddle-point problem (74) with the augmented Lagrangian function \mathcal{L} defined in (73) reads as follows:

$$\begin{aligned}
t^{(j+1)} &= \arg \min_{t \in \mathbb{R}^n} \mathcal{L}(t, z^{(j)}, \rho^{(j)}) \\
&= \left(\epsilon I_n + L^T L \right)^{-1} \left(\epsilon p + L^T \left(z^{(j)} - \frac{1}{\beta} \rho^{(j)} \right) \right), \quad \epsilon = \frac{\alpha}{\beta}, \tag{75}
\end{aligned}$$

$$\begin{aligned}
z^{(j+1)} &= \arg \min_{z \in \mathbb{R}^r} \mathcal{L}(t^{(j+1)}, z, \rho^{(j)}) = \arg \min_{z \in \mathbb{R}^r} \left\{ \Upsilon(G(z)) + \frac{\beta}{2} \|z - q^{(j)}\|_2^2 \right\} \\
&= \text{prox}_{\Upsilon(G(\cdot))}^{\beta} \left(q^{(j)} \right), \quad q^{(j)} = L t^{(j+1)} + \frac{1}{\beta} \rho^{(j)}, \tag{76}
\end{aligned}$$

$$\rho^{(j+1)} = \rho^{(j)} - \beta \left(z^{(j+1)} - L t^{(j+1)} \right). \tag{77}$$

The ADMM scheme outlined above has guaranteed convergence and, in most cases of practical interest, allows to compute very efficiently the solution t^* of (71).

In the general case, the computational cost of the ADMM iteration (75), (76), and (77) is dominated by the solution of the two subproblems for the primal variables t and z , as the cost for updating the dual variable $\rho \in \mathbb{R}^r$ by (77) is linear in r , hence in the number of pixels n (we do not consider the cost of multiplication by matrix L since the term $L t^{(j+1)}$ in (77) must have been previously computed for solving (76)).

The subproblem for t in (75) consists in solving an $n \times n$ linear system with symmetric positive definite (hence, invertible) coefficient matrix $\epsilon I_n + L^T L$. For ADMM implementations with iteration-independent penalty parameter β , the matrix is constant along the ADMM iterations, and for FB (or PDFB) implementations with iteration-independent stepsize λ , it is also constant along the (outer) FB (or PDFB) iterations. The linear system can thus be solved by direct methods, namely, Cholesky factorization carried out once for all before starting iterations and solution of (75) by forward and backward substitution, or by iterative methods. In particular, when L is a sparse matrix, the (suitably preconditioned) conjugate gradient method equipped with some variable stopping tolerance strategy represents a good (i.e., efficient) choice. If L is a diagonal matrix, or some unitary matrix (e.g., the 2D discrete Fourier or cosine transform matrix, so as to sparsify the sought image coefficients in the Fourier or cosine basis), or the matrix of some overcomplete dictionary satisfying the tight frame condition $L^T L = \delta I_n$, $\delta \in \mathbb{R}_{++}$, then the coefficient matrix is diagonal and (75) can be solved very efficiently. Finally, in the special but practically very important case where $L = \left(L_1^T, \dots, L_c^T \right)^T$ with $L_i \in \mathbb{R}^{n \times n}$ convolution matrices, $i = 1, \dots, c$, the linear system can also be solved very efficiently by fast 2D discrete transforms. In particular, by assuming periodic, symmetric, or anti-symmetric boundary conditions for the unknown image t , the linear system in (75) can be solved by using the fast 2D discrete Fourier, cosine, or sine transforms, respectively, all characterized by $O(n \log_2(n))$ computational complexity. This is the case of the TV regularizer (isotropic and anisotropic), the Hessian-based regularizers and, more in general, of the whole important class of widely used regularizers aimed to sparsify some (discretized) differential quantity of the sought image.

Based on Remark 1 in section “CNC Models with Sparsity-Inducing *Non-separable Regularizers*”, for both the FB and PDFB cases the subproblem for z in (76) can be written as

$$\hat{z} = \arg \min_{z \in \mathbb{R}^r} \left\{ \sum_{i=1}^s \nu(g_i(z)) + \frac{\beta}{2} \|z - q\|_2^2 \right\}, \quad (78)$$

where, to simplify notations, we dropped the iteration index superscripts (namely, $\hat{z} = z^{(j+1)}$ and $q = q^{(j)}$) and where the function $\nu : \mathbb{R} \rightarrow \mathbb{R}_+$ is defined by $\nu = |\cdot|$ for FB and by $\nu = \theta$ for PDFB. Then, for the important case of sparsified image feature vectors $y = G(Lx)$ characterized by the function G being the identity operator or a function of the form in (31), the r -dimensional minimization problem in (78) is separable into the following s independent (and lower-dimensional) sub-problems:

$$\hat{z}^{(i)} = \arg \min_{z^{(i)} \in \mathbb{R}^{r_i}} \left\{ \nu(\|z^{(i)}\|_2) + \frac{\beta}{2} \|z^{(i)} - q^{(i)}\|_2^2 \right\} \quad (79)$$

$$= \text{prox}_{\nu(\|\cdot\|_2)}^{\beta} \left(q^{(i)} \right), \quad i = 1, \dots, s, \quad (80)$$

where, in accordance with (31), the set of (sub-)vectors $\{z^{(i)}\}_{i=1}^s$, $z^{(i)} \in \mathbb{R}^{r_i}$, $\sum_{i=1}^s r_i = r$, represents a partition of vector $z \in \mathbb{R}^r$, i.e., $((z^{(1)})^T, \dots, (z^{(s)})^T)^T = Pz$, with $P \in \mathbb{R}^{r \times r}$ a permutation matrix. Clearly, the (sub-)vectors $\hat{z}^{(i)}$, $q^{(i)} \in \mathbb{R}^{r_i}$ in (79)–(80) are defined according to an analogous partition of vectors \hat{z} , $q \in \mathbb{R}^r$ in (78). The s minimization problems in (79) may have different dimensionality – in fact, in the considered general case, the integers r_i are not assumed to be equal – but they all have the same structure corresponding to the proximal map of the composite function $\nu(\|\cdot\|_2)$, as outlined in (80). Based on results in Proposition 7 below, under quite general and very reasonable, i.e., very likely to be satisfied in practice, assumptions on function ν , each sub-problem in (79)–(80) reduces to a 1-d strongly convex box-constrained (well-posed) minimization problem which, for most popular ν functions, admits a closed-form solution. In particular, based on (83)–(84), if ν is the absolute value function, then (79)–(80) reduces to

$$z^{(i)} = \begin{cases} 0_{r_i} & \text{if } \|q^{(i)}\|_2 = 0, \\ \max \left\{ 1 - \frac{1}{\beta \|q^{(i)}\|_2}, 0 \right\} q & \text{if } \|q^{(i)}\|_2 > 0, \end{cases} \quad i = 1, 2, \dots, s. \quad (81)$$

Recalling that, based on definition (31), the vectors $q^{(i)}$ form a partition of $q \in \mathbb{R}^r$ and, hence, $s \leq r$, the computation in (81) – including calculation of all the ℓ_2 norm terms $\|q^{(i)}\|_2$ – has linear complexity in the dimension r of the codomain of matrix $L \in \mathbb{R}^{r \times n}$, hence in the number of pixels n .

Proposition 7. *For any proper, lower semicontinuous, convex, and monotonically increasing function $\nu : \mathbb{R}_+ \rightarrow \overline{\mathbb{R}}$, the composite function $\nu(\|\cdot\|_2) : \mathbb{R}^r \rightarrow \overline{\mathbb{R}}$*

is proper, lower semicontinuous, and convex and its proximal map with proximity parameter $\beta \in \mathbb{R}_{++}$ evaluated at point $q \in \mathbb{R}^r$ is given by

$$\text{prox}_{v(\|\cdot\|_2)}^\beta(q) = \arg \min_{z \in \mathbb{R}^r} \left\{ v(\|z\|_2) + \frac{\beta}{2} \|z - q\|_2^2 \right\} \quad (82)$$

$$= \begin{cases} 0_r & \text{if } \|q\|_2 = 0, \\ \hat{\xi} \frac{q}{\|q\|_2} & \text{if } \|q\|_2 > 0, \end{cases} \quad \hat{\xi} = \arg \min_{\xi \in [0, \|q\|_2]} \left\{ v(\xi) + \frac{\beta}{2} (\xi - \|q\|_2)^2 \right\}. \quad (83)$$

In particular, if v is the identity function, then $\hat{\xi}$ in (83) is given by

$$\hat{\xi} = \max \left\{ \|q\|_2 - \frac{1}{\beta}, 0 \right\}. \quad (84)$$

Proof. First, all the stated properties of composite function $v(\|\cdot\|_2)$ come easily from the assumed properties of functions v and from the ℓ_2 norm function $\|\cdot\|_2$ being continuous and convex on the entire domain \mathbb{R}^r .

Then, convexity of $v(\|\cdot\|_2)$ yields strong convexity of the cost function in (82) which, hence, admits a unique (global) minimizer $\hat{z} \in \mathbb{R}^r$. If $\|q\|_2 = 0$ or, equivalently, q is the null vector, then the cost function in (82) reduces to $v(\|z\|_2) + (\beta/2)\|z\|_2^2$, which is a monotonically increasing function of $\|z\|_2$. The solution of (82) in this case is thus $\hat{z} = 0_r$. If q is not the null vector, i.e., $\|q\|_2 > 0$, then it is easy to prove (see the initial part of the proof of Proposition 1 in Sidky et al. 2014) that, under the considered assumptions on function v , the solution of (82) must belong to the closed segment of extremes 0_r and q . By thus considering the restriction of the cost function in (82) to that segment, parameterized by $z = \xi q / \|q\|_2$, $\xi \in [0, \|q\|_2]$, one easily obtains the 1-D constrained minimization problem in (83). Finally, the closed-form solution in (84) obtained when v is the identity function represents the quite popular multidimensional soft-thresholding operator. Its derivation can be found, e.g., in the proof of Proposition 1 in Sidky et al. (2014). \square

It is worth noticing that in the special case where the regularization matrix L is the identity matrix (e.g., when one wants to sparsify the image itself as it is expected to be predominantly zero-valued, or in general in the synthesis-based sparse reconstruction framework), then the backward step in (71) reduces to

$$t^* = \arg \min_{t \in \mathbb{R}^n} \left\{ \mathcal{Y}(G(t)) + \frac{\alpha}{2} \|t - p\|_2^2 \right\} = \text{prox}_{\mathcal{Y}(G(\cdot))}^\alpha(p). \quad (85)$$

Hence, ADMM is not required since problem (85) consists in computing only one proximal map of the same type as in the ADMM sub-problem for z in (76), which in its turn reduces to solving the s lower-dimensional problems in (79)–(80) by, e.g., (81).

Finally, we notice that a suitable warm-start strategy can be used in both the FB and PDFB approaches in order to further speedup the backward step computation by ADMM. More precisely, at each (outer) iteration of the FB and PDFB algorithms, the (inner) iterative ADMM scheme in (75), (76), and (77) is initialized with the results of previous (outer) iteration, in terms of both the primal variables t, z and the dual variable ρ . This allows to significantly decrease the number of ADMM iterations.

Numerical Examples

In this section, we test the non-convex separable and non-separable sparsity-promoting regularization terms introduced in sections “[Sparsity-Inducing Separable Regularizers](#)” and “[Sparsity-Inducing Non-separable Regularizers](#)”, respectively. More precisely, we are interested in evaluating experimentally the performance of the two classes of separable CNC-S- L_2 and non-separable CNC-NS- L_2 variational models illustrated in sections “[CNC Models with Sparsity-Inducing Separable Regularizers](#)” and “[CNC Models with Sparsity-Inducing Non-separable Regularizers](#)”, respectively, when applied to the linear discrete inverse problem of restoring images corrupted by blur and AWG noise. More broadly, the goal of this numerical session is to investigate experimentally if and how convex variational models containing non-convex sparsity-inducing regularizers, i.e., the class of CNC models, can improve upon standard convex models containing convex sparsity-promoting regularizers in case the sought solution really exhibits some sparsity property.

At this aim, we consider the three gray-scale test images SPD0, SPD1, SPD2 shown in Fig. 1 and reported again in the first row of Fig. 10. They all have resolution 256×256 pixels and, we recall, they are characterized, from left to right, by strong sparsity of the three feature vectors

$$y^{(j)} \in \mathbb{R}^n, \quad \text{with } y_i^{(0)} = |x_i|, \quad y_i^{(1)} = \|(\nabla x)_i\|_2, \quad y_i^{(2)} = \|(Hx)_i\|_F, \quad (86)$$

$i = 1, \dots, n$, respectively, where $(\nabla x)_i \in \mathbb{R}^2$ and $(Hx)_i \in \mathbb{R}^{2 \times 2}$ denote the discrete gradient and Hessian matrix of image x at pixel i . In a nutshell, the SPD0, SPD1, and SPD2 images are representatives of the three important classes of predominantly zero, piecewise constant, and piecewise affine images, respectively. For each test image, in the second, third, and fourth row of Fig. 11 we also show the three associated *binary sparsity masks* $M^{(0)}$, $M^{(1)}$, $M^{(2)}$, respectively, with 0-value pixels in black and 1-value pixels in yellow. Such masks, defined by

$$M_i^{(j)} = \begin{cases} 0 & \text{if } y_i^{(j)} = 0 \\ 1 & \text{if } y_i^{(j)} \neq 0 \end{cases}, \quad j = 0, 1, 2, \quad i = 1, \dots, n, \quad (87)$$

provide an immediate idea of the level of sparsity of each image in terms of the three feature vectors considered in (86). In Table 2, we report, for each image, the

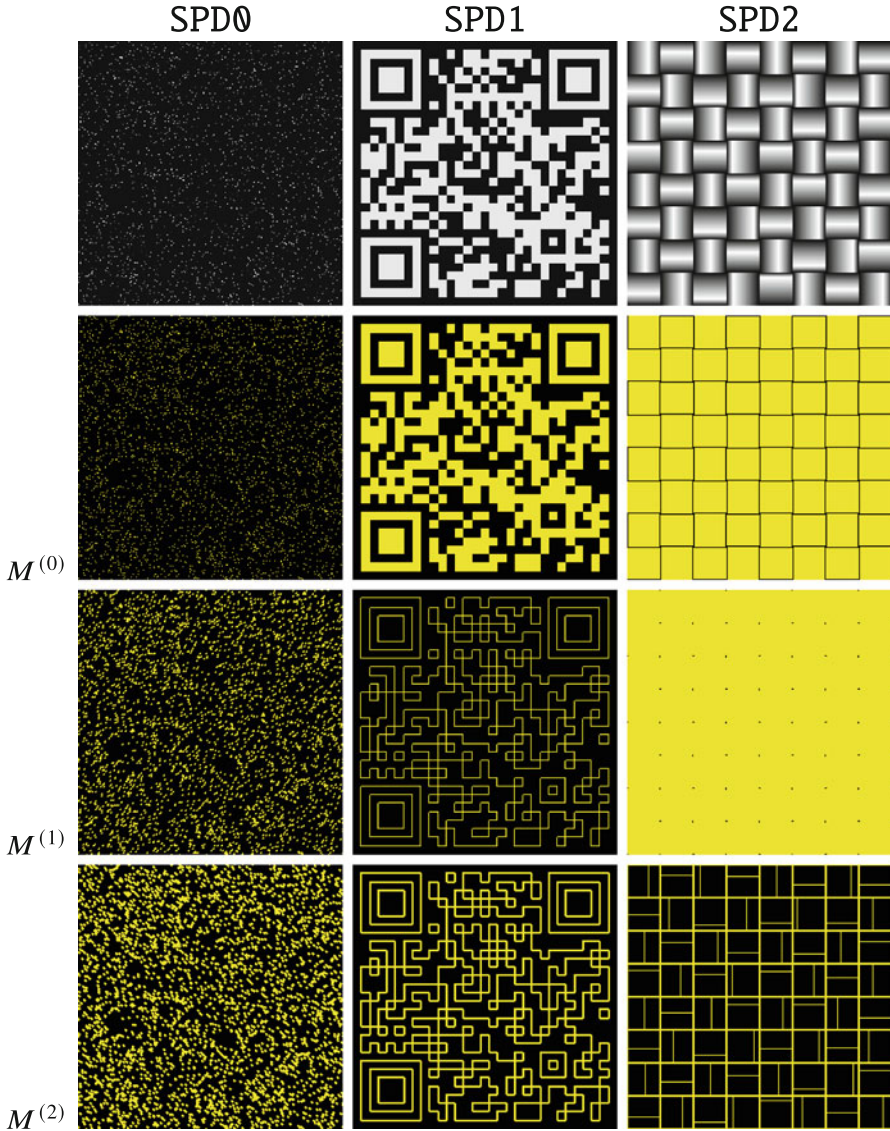


Fig. 10 The three test images SPD0, SPD1, SPD2 (first row) and their associated binary sparsity masks $M^{(j)}$, $j = 0, 1, 2$ (second-fourth rows) defined in (87) in terms of the feature vectors $y^{(j)}$, $j = 0, 1, 2$, given in (86)

total number of pixels n and the three total numbers of 0-value pixels of the binary sparsity masks defined by $\zeta^{(j)} := n - \sum_{i=1}^n M_i^{(j)}$, $j = 0, 1, 2$. As expected, the SPD0, SPD1, and SPD2 images exhibit the highest level of sparsity, i.e., the largest number of 0-value pixels, for the features vectors $y^{(0)}$, $y^{(1)}$, and $y^{(2)}$, respectively.

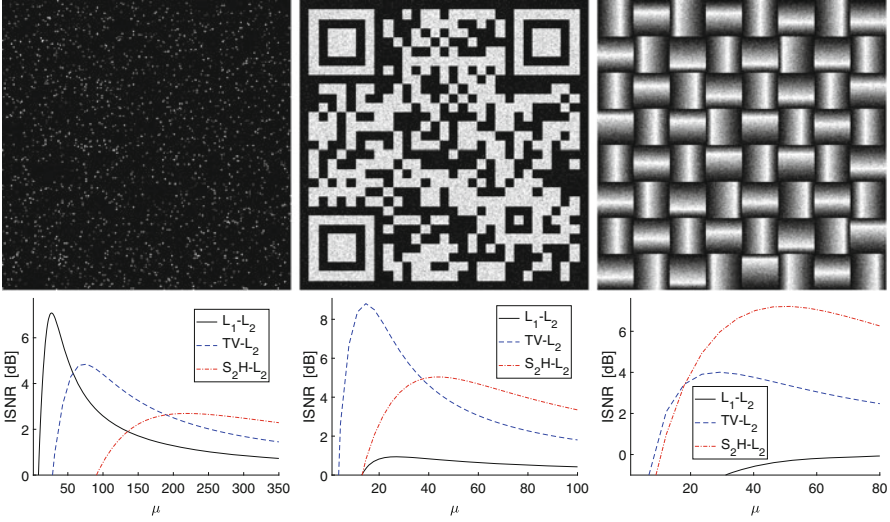


Fig. 11 The three test images SPD0, SPD1, SPD2 corrupted by AWG noise of standard deviation σ yielding $BSNR(b, \bar{x}) = 15$ (first row) and the associated ISNR graphs for the three purely convex baseline models L_1-L_2 , $TV-L_2$, S_2H-L_2 defined in (89), (90), and (91) (second row)

Table 2 Sparsity levels of the three test images SPD0, SPD1, SPD2 shown in the first row of Fig. 10 in terms of the features vectors $y^{(0)}$, $y^{(1)}$, $y^{(2)}$ defined in (86)

	SPD0	SPD1	SPD2
n	65536	65536	65536
$\zeta^{(0)}$	62255	35178	4096
$\zeta^{(1)}$	56172	58367	128
$\zeta^{(2)}$	48144	51463	55680

In accordance with the sparsity properties of the three considered test images, to evaluate the performance of the proposed CNC separable and non-separable models, we will compare them with the corresponding purely convex (i.e., with convex regularizers) models, namely, the minimal L_1 norm model (89), referred to as L_1-L_2 model, the isotropic $TV-L_2$ model (90), and the S_2H-L_2 model (91) containing the S_2H regularizer which induces sparsity of the image Hessian Shatten 2-norm (Lefkimmatis et al. 2013). More precisely, we consider the following three variational models:

$$x^* = \arg \min_{x \in \mathbb{R}^n} \mathcal{J}^{(j)}(x), \quad j = 0, 1, 2, \tag{88}$$

with cost functions defined by

$$\mathbf{L}_1 - \mathbf{L}_2 : \quad \mathcal{J}^{(0)}(x) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n |x_i|}_{L_1(x)}, \tag{89}$$

$$\mathbf{TV} - \mathbf{L}_2 : \quad \mathcal{J}^{(1)}(x) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n \|(\nabla x)_i\|_2}_{\mathbf{TV}(x)}, \quad (90)$$

$$\mathbf{S}_2\mathbf{H} - \mathbf{L}_2 : \quad \mathcal{J}^{(2)}(x) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n \|(Hx)_i\|_F}_{\mathbf{S}_2\mathbf{H}(x)}. \quad (91)$$

We thus assume that the above three models are representative of the class of purely convex models, and we compare their performance with those of the proposed associated separable CNC-S-L₂ and non-separable CNC-NS-L₂ models which, we recall, are also convex but contain non-convex regularizers.

It is worth to point out that the three models in (89), (90), and (91) can be represented in a unified form according to definition (6) of the considered class of sparsity-promoting regularizers:

$$\mathcal{J}^{(j)}(x) = \frac{1}{2} \|Ax - b\|_2^2 + \mu \|y^{(j)}\|_1, \quad y^{(j)} = G^{(j)}(L^{(j)}x), \quad j = 0, 1, 2. \quad (92)$$

In particular, the nonlinear vector-valued functions $G^{(j)} : \mathbb{R}^{r_j} \rightarrow \mathbb{R}^n$ read

$$G^{(j)}(z) = \left(g_1^{(j)}(z), \dots, g_n^{(j)}(z) \right)^T, \quad z \in \mathbb{R}^{r_j}, \quad r_j = (j+1)n, \quad j = 0, 1, 2, \quad (93)$$

with components defined by

$$g_i^{(0)}(z) = |z_i|, \quad g_i^{(1)}(z) = \left\| (z_i, z_{i+n}) \right\|_2, \quad g_i^{(2)}(z) = \left\| (z_i, z_{i+n}, z_{i+2n}) \right\|_2, \quad (94)$$

$i = 1, \dots, n$, whereas the linear operators $L^{(j)} \in \mathbb{R}^{r_j \times n}$ are

$$L^{(0)} = I_n, \quad L^{(1)} = \left(D_h^T, D_v^T \right)^T, \quad L^{(2)} = \left(D_{hh}^T, D_{vv}^T, \sqrt{2}D_{hv}^T \right)^T, \quad (95)$$

with $D_h, D_v, D_{hh}, D_{vv}, D_{hv} \in \mathbb{R}^{n \times n}$ finite difference operators discretizing the first-order horizontal and vertical and the second-order horizontal, vertical, and mixed horizontal-vertical partial derivatives, respectively. The discrete gradient and Hessian operators in (90) and (91) are thus defined in terms of these matrices as follows:

$$(\nabla x)_i = \begin{bmatrix} (D_h x)_i \\ (D_v x)_i \end{bmatrix}, \quad (Hx)_i = \begin{bmatrix} (D_{hh} x)_i & (D_{hv} x)_i \\ (D_{hv} x)_i & (D_{vv} x)_i \end{bmatrix}, \quad i = 1, \dots, n. \quad (96)$$

Finally, for what concerns the actual discretization of the gradient and Hessian operators, in all the experiments matrices $D_h, D_v, D_{hh}, D_{vv}, D_{hv}$ are the 2-D convolution matrices (with periodic boundary conditions) associated with the following point-spread functions:

$$D_h \rightarrow (+1, \mathbf{-1}), \quad D_v \rightarrow \begin{pmatrix} +1 \\ \mathbf{-1} \end{pmatrix},$$

$$D_{hh} \rightarrow (+1, \mathbf{-2}, +1), \quad D_{vv} \rightarrow \begin{pmatrix} +1 \\ \mathbf{-2} \\ +1 \end{pmatrix}, \quad D_{hv} \rightarrow \begin{pmatrix} +1 & -1 \\ -1 & \mathbf{+1} \end{pmatrix},$$

with boldface cells indicating the center of application of the PSF.

For all numerical examples, the experimental setting is as follows. The original test image \bar{x} is synthetically degraded according to the measurement model (4). First, \bar{x} is corrupted by space-invariant Gaussian blur under the assumption of periodic boundary conditions. The acquisition matrix $A \in \mathbb{R}^{n \times n}$, referred to as blurring matrix in this case, is thus block-circulant with circulant blocks and is constructed starting from the Gaussian convolution kernel, or point-spread function, generated by the Matlab command `fspecial('gaussian',band,sigma)`. The parameters `band` and `sigma` determine the bandwidth and the values of each circulant block in A , respectively. In particular, `band` represents the side length (in pixels) of the square support of the kernel, whereas `sigma` is the standard deviation of the circular, zero-mean, bivariate Gaussian probability density function representing the Gaussian point-spread function in the continuous setting. The blurred image $A\bar{x} \in \mathbb{R}^n$ is then corrupted by AWG noise with standard deviation σ to obtain the observed image $b \in \mathbb{R}^n$. Given A and b , the goal is to determine as accurately as possible estimates x^* of the original uncorrupted image \bar{x} by using variational models containing sparsity-promoting separable and non-separable regularization terms.

Regarding the optimization algorithms, the considered models are numerically solved by using the FB and PDFB splitting algorithms described in section “[Forward-Backward Minimization Algorithms](#)” and applying the illustrated ADMM strategy for the efficient computation of the backward steps. In all the experiments and for all the models, we use the observed corrupted image as the initial iterate, i.e., $x^{(0)} = b$, and we terminate the iterations as soon as two successive iterates satisfy

$$\delta_k^{(x)} := \frac{\|x^{(k)} - x^{(k-1)}\|_2}{\|x^{(k-1)}\|_2} < 10^{-5}. \quad (97)$$

The quality of the observed degraded images b and of the restored images x^* (in comparison with the original uncorrupted image \bar{x}) are measured by means of the Blurred Signal-to-Noise Ratio (BSNR) and the Improved Signal-to-Noise Ratio (ISNR), respectively. They are defined by

$$\text{BSNR}(b, \bar{x}) = \text{SNR}(b, A\bar{x}), \quad \text{ISNR}(x^*, b, \bar{x}) = \text{SNR}(x^*, \bar{x}) - \text{SNR}(b, \bar{x}), \quad (98)$$

with the Signal-to-Noise Ratio (SNR) quality measure of an image I versus a reference image \bar{I} given by

$$\text{SNR}(I, \bar{I}) := 10 \log_{10} \left(\frac{\|\bar{I} - E[\bar{I}]\|_2^2}{\|\bar{I} - I\|_2^2} \right) [dB], \quad (99)$$

where $E[\bar{I}]$ denotes the image with constant intensity equal to the mean value of \bar{I} . The larger the BSNR value, the lower is the intensity, i.e., the standard deviation σ , of the AWG noise corrupting the observation b (hence, the easier is the image restoration problem); the larger the ISNR value, the higher the quality of the restored image x^* obtained by the considered variational model. In all the experiments, after choosing the blurring operator A and computing the blurred image $A\bar{x}$, we set the desired BSNR value of the observation b and then exploit the BSNR definition in (98)–(99) in order to determine the (unique) value of the AWG noise standard deviation σ yielding the selected BSNR value:

$$\sigma = \frac{\|A\bar{x} - E[A\bar{x}]\|_2}{\sqrt{n} 10^{\frac{\text{BSNR}}{20}}}. \quad (100)$$

Examples Using CNC Separable Models

We consider the problem of denoising the three considered test images SPD0, SPD1, SPD2 corrupted only by AWG noise (no blur, i.e. $A = I_n$ in the acquisition model (4) as well as in the baseline convex variational models (89), (90), and (91)) with standard deviation σ yielding $\text{BSNR}(b, \bar{x}) = 15$, as shown in the first row of Fig. 11. The three separable CNC variational models, referred to as CNC-S-L₁-L₂, CNC-S-TV-L₂, and CNC-S-S₂H-L₂, to be compared with the baseline purely convex models L₁-L₂, TV-L₂ and S₂H-L₂ defined in (89), (90), and (91), read as follows:

$$x^* = \arg \min_{x \in \mathbb{R}^n} \mathcal{J}_S^{(j)}(x; a), \quad j = 0, 1, 2, \quad (101)$$

with cost functions defined by

CNC – S – L₁ – L₂ :

$$\mathcal{J}_S^{(0)}(x; a) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n \phi_{\text{MC}}(|x_i|; a)}_{\text{S-L}_1(x;a)}, \quad (102)$$

CNC – S – TV – L₂ :

$$\mathcal{J}_S^{(1)}(x; a) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n \phi_{\text{MC}}(\|(\nabla x)_i\|_2; a)}_{\text{S-TV}(x;a)}, \quad (103)$$

CNC – S – S₂H – L₂ :

$$\mathcal{J}_S^{(2)}(x; a) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \sum_{i=1}^n \phi_{\text{MC}}(\|(Hx)_i\|_F; a)}_{\text{S-S}_2\text{H}(x;a)}, \quad (104)$$

where ϕ_{MC} is the scalar MC penalty function defined in (13) and where we are assuming a space-invariant, i.e., constant for all pixel locations, concavity parameter $a \in \mathbb{R}_{++}$ for ϕ_{MC} . It follows from Theorem 1, in particular, condition (33), that sufficient conditions for the three cost functions above to be convex (strongly convex) are the following:

$$\mathcal{Q}^{(j)} = I_n - \mu a \left(L^{(j)}\right)^T L^{(j)} \succeq 0 \ (\succ 0), \quad j = 0, 1, 2, \quad (105)$$

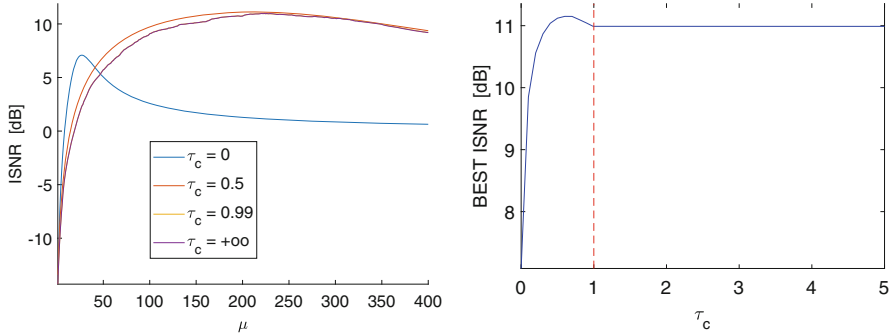
where we used the fact that $A = I_n$ for the considered case of image denoising and where the regularization matrices $L^{(j)}$ are defined in (95). According to the statement of Theorem 1, the sufficient conditions in (105) can be equivalently and usefully rewritten as follows:

$$a = \tau_c \frac{1}{\mu \kappa^{(j)}}, \quad \tau_c \in [0, 1] \ (\tau_c \in [0, 1[), \quad j = 0, 1, 2, \quad (106)$$

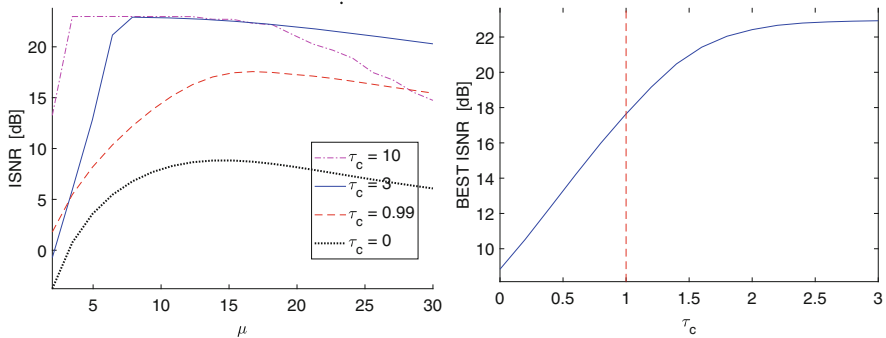
with the scalar coefficients $\kappa^{(j)}$ given by

$$k^{(j)} = \sigma_{\max}^2 \left(L^{(j)}\right), \quad j = 0, 1, 2 \implies k^{(0)} = 1, \quad k^{(1)} = 8, \quad k^{(2)} = 64. \quad (107)$$

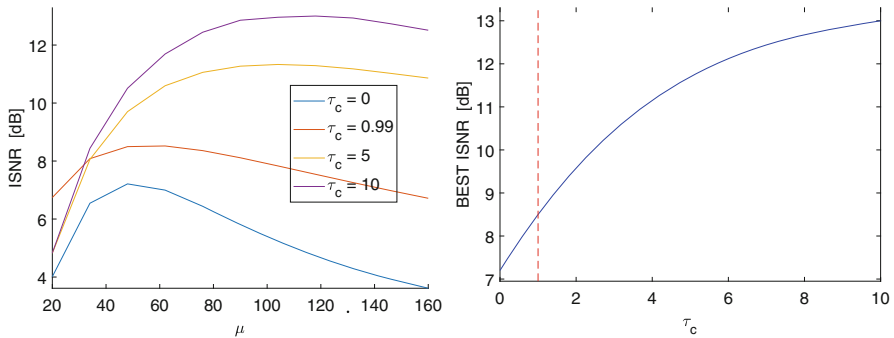
As a preliminary experiment, we evaluate the performance of the three baseline convex models L₁-L₂, TV-L₂, and S₂H-L₂ defined in (89), (90), and (91) when applied to the three corrupted images illustrated in the first row of Fig. 11. The plots



(a) Results of model CNC-S- L_1 - L_2 applied to the noise-corrupted test image SPD0



(b) Results of model CNC-S-TV- L_2 applied to the noise-corrupted test image SPD1



(c) Results of model CNC-S- S_2 H- L_2 applied to the noise-corrupted test image SPD2

Fig. 12 ISNR results of separable CNC models CNC-S- L_1 - L_2 , CNC-S-TV- L_2 , and CNC-S- S_2 H- L_2 defined in (102), (103), and (104) when applied to the noise-corrupted images SPD0, SPD1, and SPD2, respectively. First column: ISNR values as a function of the regularization parameter μ for some different τ_c values. Second column: highest achieved ISNR values as a function of the convexity coefficient τ_c . The dashed vertical red lines, corresponding to $\tau_c = 1$, separate, for each model, the pure convex and CNC regimes ($\tau_c \in [0, 1)$) from the pure non-convex regime ($\tau_c \in]1, +\infty[$).

in Fig. 11 (second row) represent the ISNR values achieved by the three models as a function of the regularization parameter μ for the three corresponding noise-corrupted test images SPD0, SPD1, and SPD2 illustrated in the first row. From a visual inspection, column by column, of Fig. 11 (second row), we observe that, as expected, the best ISNR values are obtained by models L_1 - L_2 , TV- L_2 , and S_2 H- L_2 on images SPD0, SPD1 and SPD2, respectively. This is completely in accordance with the sparsity properties of the three images. The regularizers of models L_1 - L_2 , TV- L_2 , and S_2 H- L_2 are in fact suitable for predominantly zero, piecewise constant, and piecewise affine images, respectively, as they promote sparsity of the intensities and of the first- and second-order intensity derivatives of the restored image.

In the next experiment, we compare the best assessed regularization models in the three convexity regimes: pure convex ($\tau_c = 0$), CNC ($\tau_c \in (0, 1]$), and pure non-convex regime ($\tau_c > 1$). In other words, we now test the three separable CNC models CNC-S- L_1 - L_2 , CNC-S-TV- L_2 , and CNC-S- S_2 H- L_2 defined in (102), (103), and (104) on the corresponding test images for different τ_c values. In Fig. 12, for each test image SPD0 (first row), SPD1 (second row), and SPD2 (third row), we report some interesting ISNR curves for the associated best-performing models CNC-S- L_1 - L_2 , CNC-S-TV- L_2 , and CNC-S- S_2 H- L_2 , respectively. In particular, the plots in the first column represent, for some different τ_c values, the achieved ISNR values as a function of the regularization parameter μ . The curves in the second column depict, for a fine grid of τ_c values, the highest ISNR values achieved by letting μ vary in its entire domain.

In Figs. 13, 14, and 15, we report the best (i.e., with highest associated ISNR value) denoising results obtained by applying models CNC-S- L_1 - L_2 , CNC-S-TV- L_2 , and CNC-S- S_2 H- L_2 to the noise-corrupted test images SPD0, SPD1, and SPD2, respectively, with different τ_c values. In particular, in the first column of Figs. 13, 14, and 15, we show the denoised images, whereas in the second column we report the associated absolute error images.

From ISNR plots reported in the second column of Fig. 12, we can first observe that usefulness of using high τ_c values becomes larger as the order of image derivatives sparsified by the regularizer increases. For model CNC-S- L_1 - L_2 , the best results are obtained in the CNC regime, i.e., for $\tau_c \in]0, 1]$. We recall that in this case the upper limit of the CNC regime ($\tau_c = 1$) corresponds to using $\|x\|_0$ as the regularizer, such that the solution is obtained by a pixel-wise hard thresholding of the noisy observation b . For the CNC-S-TV- L_2 model, the ISNR gain obtained by the CNC regime is remarkable, whereas for the CNC-S- S_2 H- L_2 model, such gain is smaller. In other words, pushing the model in pure non-convex regime ($\tau_c > 1$) is much more appealing for CNC-S- S_2 H- L_2 than for CNC-S-TV- L_2 .

Examples Using CNC Non-separable Models

In this section, we test the performance of the proposed non-separable CNC variational models when applied to image denoising and deblurring problems. In fact, unlike the separable CNC strategy, the non-separable CNC approach can be

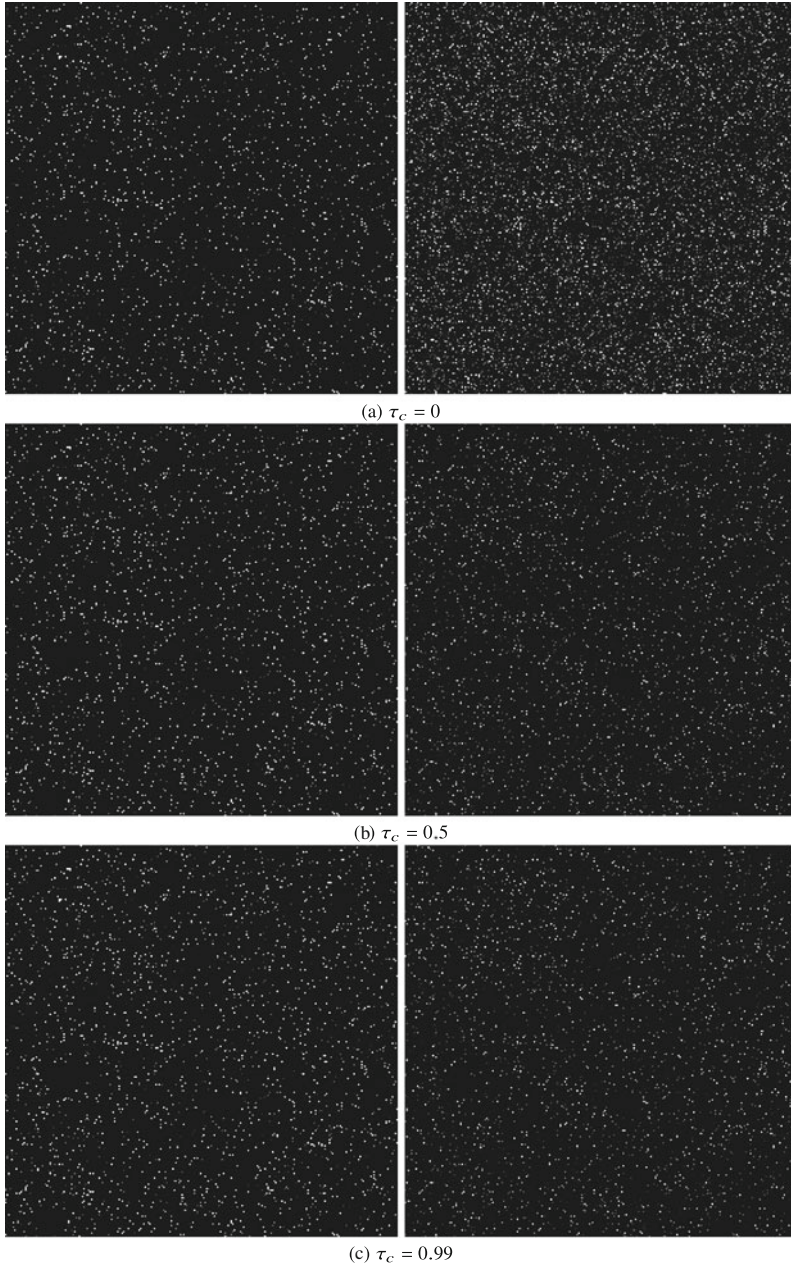


Fig. 13 *Separable CNC models*. Best denoising results obtained by CNC-S-L₁-L₂ on image SPD0 for different τ_c values (left column) and associated absolute error images (right column)

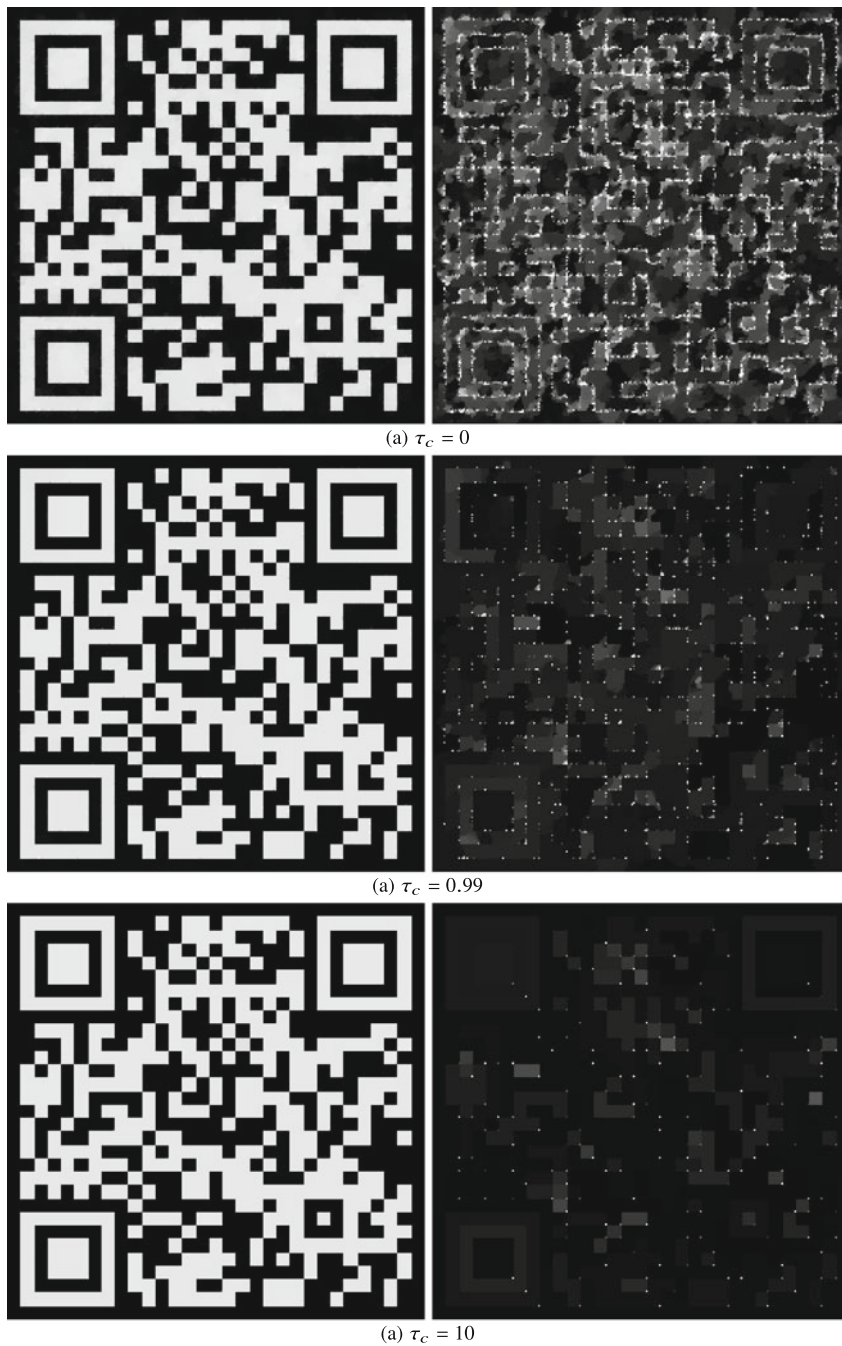


Fig. 14 Separable CNC models. Best denoising results obtained by CNC-S-TV-L₂ on image SPD1 for different τ_c values (left column) and associated absolute error images (right column)

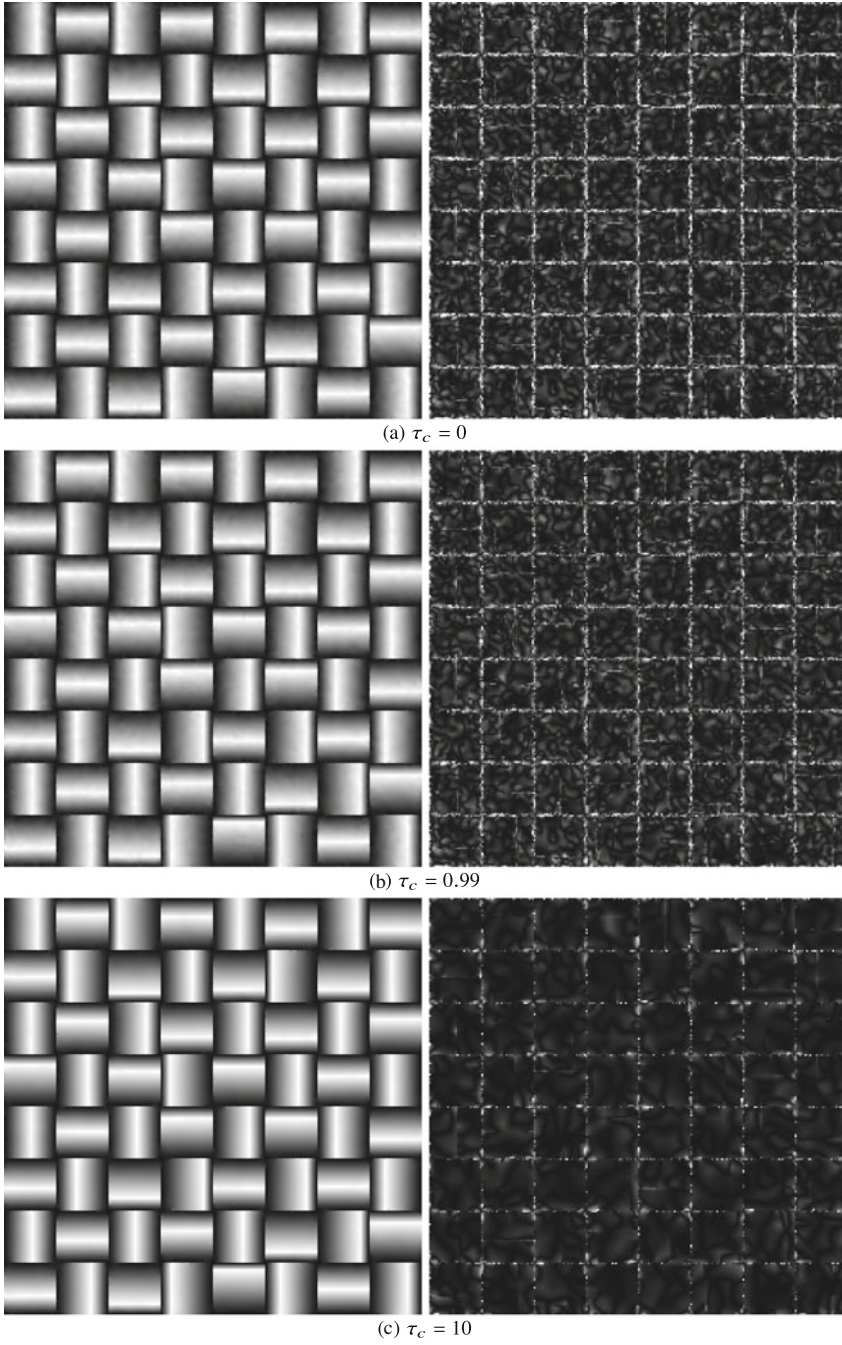


Fig. 15 *Separable CNC models.* Best denoising results obtained by CNC-S- S_2 H-L₂ on image SPD2 for different τ_c values (left column) and associated absolute error images (right column)

usefully applied for any acquisition matrix A , also when A is very ill-conditioned or even numerically singular like it is often the case in deblurring problems. More precisely, we consider the restoration of the piecewise constant image SPD1 and the piecewise affine image SPD2 depicted in the first row of Fig. 10 which, we recall, are characterized by sparse first- and second-order derivatives, respectively.

In accordance with the considered degradation model in (4), the two test images SPD1 and SPD2 have been synthetically corrupted by space-invariant Gaussian blur and AWG noise, as described at the beginning of section “Numerical Examples”. In particular, for the denoising experiment, clearly A is the identity operator, and no synthetic blur is applied, whereas for the deblurring experiment, the Gaussian point-spread function is generated with parameters `band = 7`, `sigma = 1.5`. We then add AWG noise corruptions of standard deviations σ yielding $\text{BSNR}(b, \bar{x}) = 15$ for the denoising case and $\text{BSNR}(b, \bar{x}) = 7.6$ for the deblurring case.

For the restoration, i.e., denoising and/or deblurring, of the degraded SPD1 and SPD2 test images, we consider the non-separable CNC versions, referred to as CNC-NS-TV- L_2 and CNC-NS- S_2 H- L_2 , of the two separable CNC models CNC-S-TV- L_2 and CNC-S- S_2 H- L_2 defined in (103) and (104), respectively. We also consider a slightly different but interesting version of the CNC-NS- S_2 H- L_2 model, referred to as CNC-NS- S_1 H- L_2 , where the Shatten 2-norm (Frobenious norm) has been replaced by the Shatten 1-norm (nuclear norm).

The three considered non-separable CNC models thus read

$$x^* = \arg \min_{x \in \mathbb{R}^n} \mathcal{J}_{\text{NS}}^{(j)}(x; B), \quad j = 1, 2, 3, \quad (108)$$

with cost functions defined by

CNC – NS – TV – L_2 :

$$\mathcal{J}_{\text{NS}}^{(1)}(x; B) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \left(\text{TV}(x) - \left(\text{TV} \square \frac{1}{2} \|B \cdot\|_2^2 \right)(x) \right)}_{\text{NS-TV}(x; B)}, \quad (109)$$

CNC – NS – S_2 H – L_2 :

$$\mathcal{J}_{\text{NS}}^{(2)}(x; B) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \left(S_2\text{H}(x) - \left(S_2\text{H} \square \frac{1}{2} \|B \cdot\|_2^2 \right)(x) \right)}_{\text{NS-}S_2\text{H}(x; B)}, \quad (110)$$

CNC – NS – S_1 H – L_2 :

$$\mathcal{J}_{\text{NS}}^{(3)}(x; B) = \frac{1}{2} \|Ax - b\|_2^2 + \underbrace{\mu \left(S_1\text{H}(x) - \left(S_1\text{H} \square \frac{1}{2} \|B \cdot\|_2^2 \right)(x) \right)}_{\text{NS-}S_1\text{H}(x; B)}. \quad (111)$$

Table 3 ISNR values obtained by restoring the test images SPD1 and SPD2 corrupted by zero-mean AWG noise (Denoise) and space-invariant Gaussian blur (Deblur)

Image	Model	Denoise	Deblur	Image	Model	Denoise	Deblur
SPD1	TV-L ₂	8.84	6.64	SPD2	TV-L ₂	7.21	3.00
	S ₁ H-L ₂	5.56	2.00		S ₁ H-L ₂	7.67	2.50
	S ₂ H-L ₂	4.54	1.90		S ₂ H-L ₂	6.65	2.73
	CNC-NS-TV-L ₂	20.35	6.72		CNC-NS-TV-L ₂	4.11	3.20
	CNC-NS-S ₁ H-L ₂	11.34	2.11		CNC-NS-S ₁ H-L ₂	12.33	2.83
	CNC-NS-S ₂ H-L ₂	9.13	2.00		CNC-NS-S ₂ H-L ₂	10.57	2.73

The parameter matrix B has been constructed using dc-notch filters as described at the end of section “[Construction of Matrix \$B\$](#) ”, so that the three total cost functions above are all convex, and hence, the three models are CNC.

Quantitative and qualitative (visual) results have been produced. In Table 3, we report the ISNR values obtained by the three considered non-separable CNC models on the two test images for both the denoising and deblurring experiments. For comparison, we also report the ISNR values achieved by using the associated purely convex baseline models. For each experiment, the best ISNR results within each class of models are marked in boldface. Figures 16 and 17 show the corrupted images (top rows) and the best restored images computed by the two classes of purely convex models (center rows) and non-separable CNC models (bottom rows), in case of denoising and deblurring, respectively, see the associated ISNR values marked in boldface in Table 3.

From the ISNR values in Table 3 and the visual inspection of the restored images in Figs. 16 and 17, the improvement in accuracy provided by the considered non-convex non-separable regularizers versus the corresponding convex separable baseline regularizers is evident, particularly for the denoising case, and in agreement with the sparsity characteristics of the two images. It is worth remarking that such improvement is obtained without renouncing any of the well-known advantages of (strongly) convex optimization, namely, the existence of a unique (global) minimizer and of numerical algorithms with proved convergence toward such minimizer.

Furthermore, for the denoising results we could also extend the comparison to the CNC models with separable regularizers, which were demonstrated in section “[Examples Using CNC Separable Models](#)” to outperform the baseline purely convex models in inducing sparsity of the gradient magnitudes or the Hessian Shatten 2-norms in the denoised images.

To conclude, we notice that for both the separable and non-separable CNC considered models, the regularization parameter μ has been set manually so as to achieve the best accuracy results in terms of ISNR. In practical applications, clearly this procedure can not be used (the true image \bar{x} is unknown), and also manually tuning μ by visually inspecting the attained results is not practical. Hence, some sort of automatic parameter selection strategy is always highly desirable. Actually,

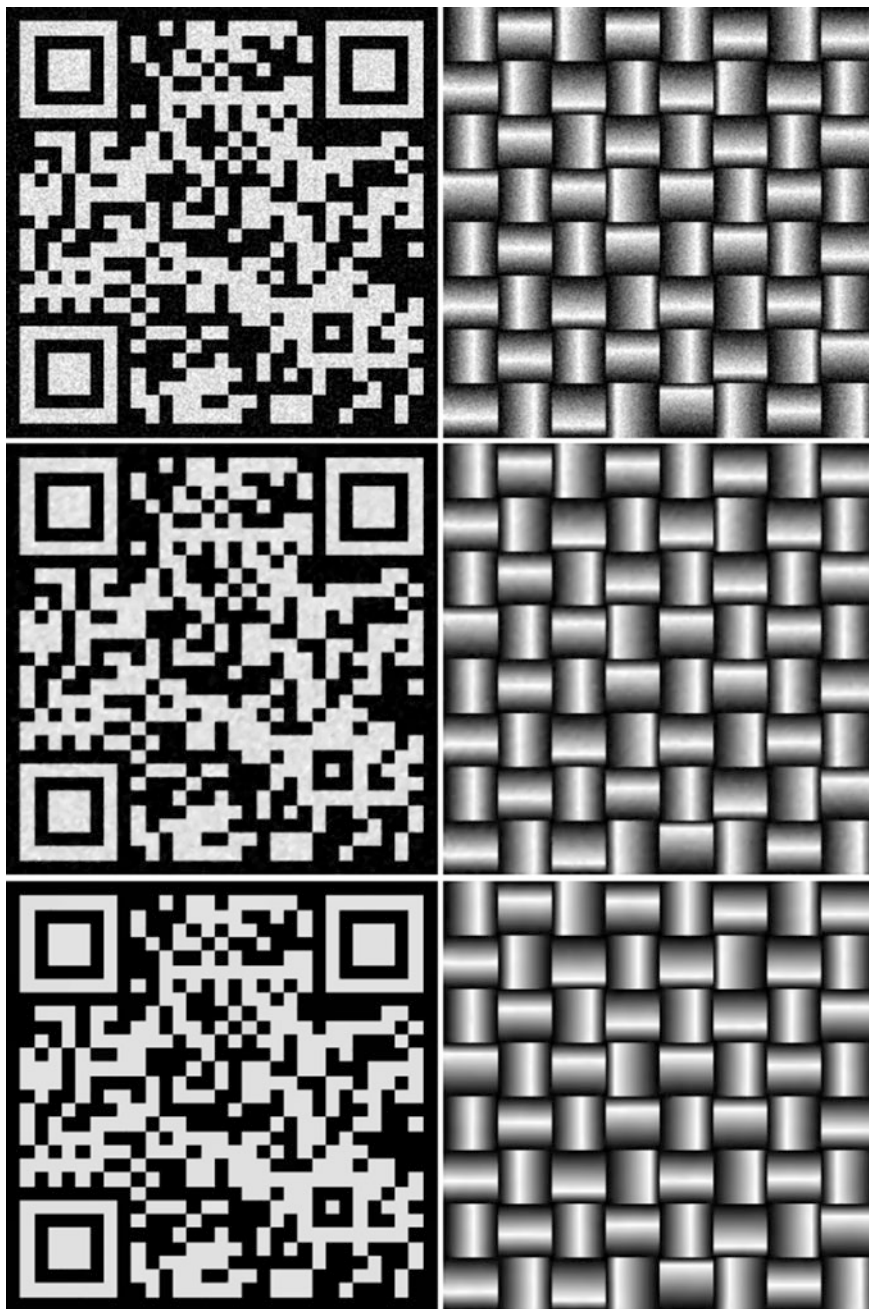


Fig. 16 *Non-separable CNC models.* Denoising results on SPD1 (left column) and SPD2 (right column) corrupted by AWG noise. First row: degraded images (BSNR = 15). Second row: restorations by TV- L_2 (ISNR=8.84), left, and by S_1H-L_2 (ISNR=7.67), right. Third row: restorations by CNC-NS-TV- L_2 (ISNR=20.35), left, and by CNC-NS- S_1H-L_2 (ISNR=12.33), right

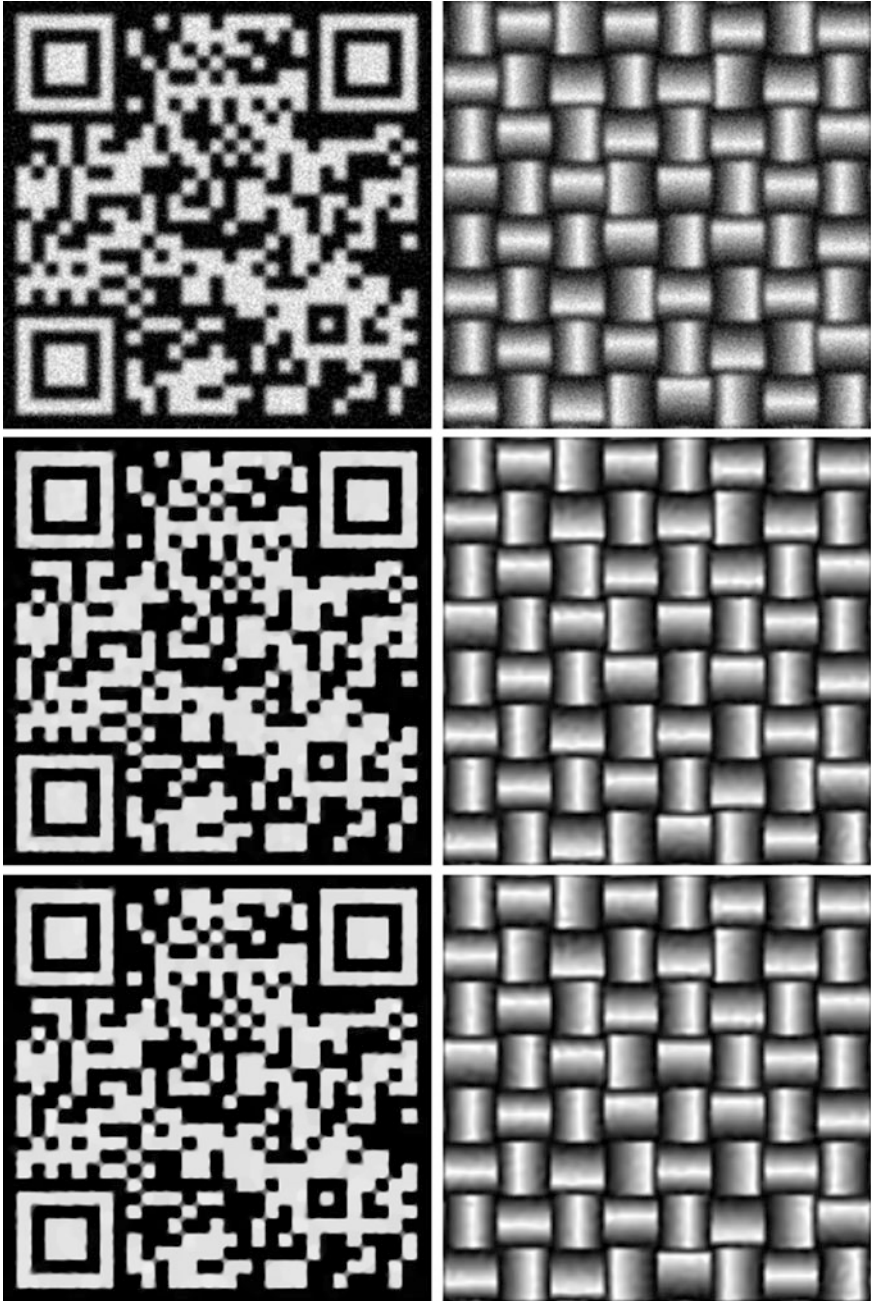


Fig. 17 *Non-separable CNC models.* Deblurring results on SPD1 (left column) and SPD2 (right column) corrupted by blur and AWG noise. First row: degraded images (BSNR = 7.6). Second row: restorations by $TV-L_2$ (ISNR=6.64), left, and by S_1H-L_2 (ISNR=3.00), right. Third row: restorations by $CNC-NS-TV-L_2$ (ISNR=6.72), left, $CNC-NS-S_1H-L_2$ (ISNR=3.20), right

the proposed FB and PDFB numerical solution algorithms can be quite easily equipped with such an automatic strategy. In particular, if one wants to select μ according to the very popular *discrepancy principle* or to the less popular but very effective *residual whiteness principle*, the ADMM approach proposed for solving the backward denoising step can benefit from the adaptive strategies proposed in Lanza et al. Lanza et al. (2016b, 2021, 2020) for the more general class of deblurring problems.

Conclusion

We discussed a CNC strategy for sparsity-inducing regularization of linear least-squares inverse problems. To avoid the intrinsic difficulties related to non-convex optimization, the CNC strategy allows the use of non-convex regularization while maintaining convexity of the total cost function. In this work we analyzed a general class of parameterized non-convex sparsity-promoting separable and non-separable regularizers and their associated CNC variational models. We derived convexity conditions for the total cost functions and we discussed related theoretical properties. A general forward-backward splitting strategy has been presented and applied for the numerical solution of the CNC models considered and a theoretical proof of convergence has been given. A series of numerical experiments related to image denoising and deblurring have been carried out, and the reported results strongly indicate that the considered non-convex regularizers hold the potential for achieving high quality results while remaining in a convex, safe regime.

Acknowledgments This research was supported in part by the National Group for Scientific Computation (GNCS-INDAM), Research Projects 2019/2020.

References

- Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York (2011)
- Bayram, I.: On the convergence of the iterative shrinkage/thresholding algorithm with a weakly convex penalty. IEEE Trans. Signal Process. **64**(6), 1597–1608 (2016)
- Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
- Becker, S., Combettes, P.L.: An algorithm for splitting parallel sums of linearly composed monotone operators, with applications to signal recovery. J. Nonlinear Convex Anal. **15**(1), 137–159 (2014)
- Bello Cruz, J.Y.: On proximal subgradient splitting method for minimizing the sum of two nonsmooth convex functions. Set-Valued Var. Anal **25**, 245–263 (2017)
- Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge, MA (1987)
- Bruckstein, A., Donoho, D., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**(1), 34–81 (2009)
- Burger, M., Papafitsoros, K., Papoutsellis, E., Schönlieb, C.B.: Infimal convolution regularisation functionals of BV and L_p spaces. J. Math. Imaging Vis. **55**(3), 343–369 (2016)

- Cai, G., Selesnick, I.W., Wang, S., Dai, W., Zhu, Z.: Sparsity enhanced signal decomposition via generalized minimax-concave penalty for gearbox fault diagnosis. *J. Sound Vib.* **432**, 213–234 (2018)
- Candés, E.J., Wakin, M.B., Boyd, S.: Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.* **14**(5), 877–905 (2008)
- Carlsson, M.: On convexification/optimization of functionals including an ℓ_2 -misfit term. arXiv preprint arXiv:1609.09378 (2016)
- Castella, M., Pesquet, J.C.: Optimization of a Geman-McClure like criterion for sparse signal deconvolution. In: *IEEE International Workshop on Computational Advances Multi-sensor Adaptive Processing*, pp. 309–312 (2015)
- Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. *Numerische Mathematik* **76**, 167–188 (1997)
- Chan, R., Lanza, A., Morigi, S., Sgallari, F.: Convex non-convex image segmentation. *Numerische Mathematik* **138**(3), 635–680 (2017)
- Chartrand, R.: Shrinkage mappings and their induced penalty functions. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1026–1029 (2014)
- Chen, P.Y., Selesnick, I.W.: Group-sparse signal denoising: non-convex regularization, convex optimization. *IEEE Trans. Signal Proc.* **62**, 3464–3478 (2014)
- Chouzenoux, E., Jeziarska, A., Pesquet, J., Talbot, H.: A majorize-minimize subspace approach for ℓ_2 - ℓ_0 image regularization. *SIAM J. Imag. Sci.* **6**(1), 563–591 (2013)
- Ding, Y., Selesnick, I.W.: Artifact-free wavelet denoising: nonconvex sparse regularization, convex optimization. *IEEE Signal Process. Lett.* **22**(9), 1364–1368 (2015)
- Du, H., Liu, Y.: Minmax-concave total variation denoising. *Signal Image Video Process.* **12**(6), 1027–1034 (2018)
- Geiger, D., Giasi, F.: Parallel and deterministic algorithms from MRF's: surface reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(5), 410–412 (1991)
- Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE PAMI* **6**(6), 721–741 (1984)
- Hansen, P.C.: *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia (1997)
- Hartman, P.: On functions representable as a difference of convex functions. *Pac. J. Math.* **9**(3), 707–713 (1959)
- Hastie, T., Tibshirani, R., Wainwright, M.: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton (2015)
- Huska, M., Lanza, A., Morigi, S., Sgallari, F.: Convex non-convex segmentation of scalar fields over arbitrary triangulated surfaces. *J. Comput. Appl. Math.* **349**, 438–451 (2019a)
- Huska, M., Lanza, A., Morigi, S., Selesnick, I.W.: A convex-nonconvex variational method for the additive decomposition of functions on surfaces. *Inverse Problems* **35**, 124008–124041 (2019b)
- Jensen, J.B., Nielsen, M.: A simple genetic algorithm applied to discontinuous regularization. In: *Proceedings IEEE workshop on NNSP, Copenhagen* (1992)
- Lanza, A., Morigi, S., Sgallari, F.: Convex image denoising via non-convex regularization. *Scale Space Variat. Methods Comput. Vis.* **9087**, 666–677 (2015)
- Lanza, A., Morigi, S., Sgallari, F.: Convex image denoising via nonconvex regularization with parameter selection. *J. Math. Imaging Vis.* **56**(2), 195–220 (2016a)
- Lanza, A., Morigi, S., Sgallari, F.: Constrained TV_p - ℓ_2 model for image restoration. *J. Sci. Comput.* **68**, 64–91 (2016b)
- Lanza, A., Morigi, S., Selesnick, I.W., Sgallari, F.: Nonconvex nonsmooth optimization via convex-nonconvex majorization minimization. *Numerische Mathematik* **136**(2), 343–381 (2017)
- Lanza, A., Morigi, S., Sgallari, F.: Automatic parameter selection based on residual whiteness for convex non-convex variational restoration. In: *Mathematical Methods in Image Processing and Inverse Problems* (eds) Tai XC, Wei S, Liu H. Springer, Singapore, **360**, (2021). <https://doi.org/10.1007/978-981-16-2701-9>
- Lanza, A., Morigi, S., Selesnick, I.W., Sgallari, F.: Sparsity-inducing nonconvex nonseparable regularization for convex image processing. *SIAM J. Imag. Sci.* **12**(2), 1099–1134 (2019)
- Lanza, A., Pragliola, M., Sgallari, F.: Residual whiteness principle for parameter-free image restoration. *Electron. Trans. Numer. Anal.* **53**, 329–351 (2020)

- Lefkimiatis, S., Ward, J., Unser, M.: Hessian Schatten-Norm regularization for linear inverse problems. *IEEE Trans. Image Process.* **22**, 1873–1888 (2013)
- Malek-Mohammadi, M., Rojas, C.R., Wahlberg, B.: A class of nonconvex penalties preserving overall convexity in optimization based mean filtering. *IEEE Trans. Signal Process.* **64**(24), 6650–6664 (2016)
- Nikolova, M.: Estimation of binary images by minimizing convex criteria. *Proc. IEEE Int. Conf. Image Process.* **2**, 108–112 (1998)
- Nikolova, M.: Energy minimization methods. In: Scherzer, O. (ed.) *Handbook of Mathematical Methods in Imaging*, Chapter 5, pp. 138–186. Springer, Berlin (2011)
- Nikolova, M., Ng, M.K., Tam, C.P.: Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. *IEEE Trans. Image Process.* **19**(12), 3073–3088 (2010)
- Parekh, A., Selesnick, I.W.: Convex denoising using non-convex tight frame regularization. *IEEE Signal Process. Lett.* **22**(10), 1786–1790 (2015)
- Parekh, A., Selesnick, I.W.: Enhanced low-rank matrix approximation. *IEEE Signal Process. Lett.* **23**(4), 493–497 (2016)
- Park, T.W., Burrus, C.S.: *Digital Filter Design*. Wiley, New York (1987)
- Portilla, J., Mancera, L.: L0-based sparse approximation: two alternative methods and some applications. In: *Proceedings of SPIE, San Diego*, vol. 6701 (Wavelets XII) (2007)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physics D* **60**(1–4), 259–268 (1992)
- Selesnick, I.W.: Sparse regularization via convex analysis. *IEEE Trans. Signal Process.* **65**(17), 4481–4494 (2017a)
- Selesnick, I.W.: Total variation denoising via the Moreau envelope. *IEEE Signal Process. Lett.* **24**(2), 216–220 (2017b)
- Selesnick, I.W., Bayram, I.: Sparse signal estimation by maximally sparse convex optimization. *IEEE Trans. Signal Proc.* **62**(5), 1078–1092 (2014)
- Selesnick, I.W., Parekh, A., Bayram, I.: Convex 1-D total variation denoising with non-convex regularization. *IEEE Signal Process. Lett.* **22**, 141–144 (2015)
- Selesnick, I.W., Lanza, A., Morigi, S., Sgallari, F.: Non-convex total variation regularization for convex denoising of signals. *J. Math. Imag. Vis.* **62**, 825–841 (2020)
- Setzer, S., Steidl, G., Teuber, T.: Infimal convolution regularizations with discrete l1-type functionals. *Commun. Math. Sci.* **9**(3), 797–827 (2011)
- Shen, L., Xu, Y., Zeng, X.: Wavelet inpainting with the l0 sparse regularization. *J. Appl. Comp. Harm. Anal.* **41**(1), 26–53 (2016)
- Sidky, E.Y., Chartrand, R., Boone, J.M., Pan, X.: Constrained TpV-minimization for enhanced exploitation of gradient sparsity: application to CT image reconstruction. *IEEE J. Trans. Eng. Health Med.* **2**, 1–18 (2014)
- Soubies, E., Blanc-Féraud, L., Aubert, G.: A continuous exact L0 penalty (CEL0) for least squares regularized problem. *SIAM J. Imag. Sci.* **8**(3), 1607–1639 (2015)
- Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **1**, 211–244 (2001)
- Tuy, H.: DC optimization: theory, methods and algorithms. In: *Handbook of Global Optimization*, pp. 149–216. Springer, Boston, (1995)
- Wang, S., Selesnick, I.W., Cai, G., Ding, B., Chen, X.: Synthesis versus analysis priors via generalized minimax-concave penalty for sparsity-assisted machinery fault diagnosis. *Mech. Syst. Signal Process.* **127**, 202–233 (2019)
- Wipf, D.P., Rao, B.D., Nagarajan, S.: “Latent variable Bayesian models for promoting sparsity. In: *IEEE Trans. Inf. Theory* **57**(9), 6236–6255 (2011)
- Yuille, A.L., Rangarajan, A.: The concave-convex procedure. *Neural Comput.* **15**(4), 915–936 (2003)
- Zhang, C.-H.: Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* **38**(2), 894–942 (2010)
- Zou, J., Shen, M., Zhang, Y., Li, H., Liu, G., Ding, S.: Total variation denoising with non-convex regularizers. *IEEE Access* **7**, 4422–4431 (2019)



Subsampled First-Order Optimization Methods with Applications in Imaging

2

Stefania Bellavia, Tommaso Bianconcini, Nataša Krejić,
and Benedetta Morini

Contents

Introduction	62
Convolutional Neural Networks	64
Convolutional Layer	67
Max Pooling Layer	68
Stochastic Gradient and Variance Reduction Methods	69
Gradient Methods with Adaptive Steplength Selection Based on Globalization Strategies	77
Accuracy Requirements	81
Stochastic Line Search	82
Adaptive Regularization and Trust-Region	84
Numerical Experiments	85
The Neural Network in Action	86
Training the Neural Network	88
Implementation Details	90
Results	90
Conclusion	91
References	92

S. Bellavia · B. Morini (✉)
Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze (INdAM-GNCS
members), Firenze, Italia
e-mail: stefania.bellavia@unifi.it; benedetta.morini@unifi.it

T. Bianconcini
Verizon Connect, Firenze, Italia
e-mail: tommaso.bianconcini@verizonconnect.com

N. Krejić
Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi
Sad, Serbia
e-mail: natasak@uns.ac.rs

Abstract

This work presents and discusses optimization methods for solving finite-sum minimization problems which are pervasive in applications, including image processing. The procedures analyzed employ first-order models for the objective function and stochastic gradient approximations based on subsampling. Among the variety of methods in the literature, the focus is on selected algorithms which can be cast into two groups: algorithms using gradient estimates evaluated on samples of very small size and algorithms relying on gradient estimates and machinery from standard globally convergent optimization procedures. Neural networks and convolutional neural networks widely used for image processing tasks are considered, and a classification problem of images is solved with some of the methods presented.

Keywords

Finite-sum minimization · First-order methods · Stochastic gradient · Neural networks · Convolutional neural networks · Image classification

Introduction

The focus of this paper is on finite-sum minimization

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lipschitz smooth function of the form

$$f(x) = \frac{1}{N} \sum_{i=1}^N f_i(x), \quad (2)$$

and each f_i is such that $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We assume that f is bounded from below in \mathbb{R}^n .

The case of interest here is when problem dimension n and N are large numbers. Such finite-sum minimization comprises a variety of applications including problems from machine learning Bottou et al. (2018) and plays an important role in image processing, e.g., in tasks such as image classification, object detection, and image segmentation (Aggarwal 2018; Chollet 2017; Forsyth et al. 2002; Goodfellow et al. 2016; Patterson et al. 2017; Shanmugamani 2018).

In a large-scale regime, working with the objective function f and its gradient in first-order methods, or even Hessian in the second-order methods, may be prohibitively expensive. In order to reduce the computational cost, typically f and its derivatives are approximated using a subset of the summation terms. In particular, such approximation is carried out by *subsampling*, i.e., considering summation terms corresponding to a random sample of indices $\mathcal{S} \subseteq \{1, \dots, N\}$. The random sample set \mathcal{S} is also called mini-batch if it is a small subset of $\{1, \dots, N\}$.

Considering first-order methods, let k be the iteration index and f_k^0 and g_k be subsampled approximation of $f(x_k)$ and $\nabla f(x_k)$, respectively, i.e.,

$$f_k^0 = \frac{1}{|\mathcal{S}_{k,f}|} \sum_{i \in \mathcal{S}_{k,f}} f_i(x_k), \quad (3)$$

$$g_k = \frac{1}{|\mathcal{S}_{k,g}|} \sum_{i \in \mathcal{S}_{k,g}} \nabla f_i(x_k), \quad (4)$$

where $\mathcal{S}_{k,f}$ and $\mathcal{S}_{k,g}$ are random subsets of $\{1, \dots, N\}$ and $|\mathcal{S}_{k,f}|$, $|\mathcal{S}_{k,g}|$ denote their cardinality. Then, the k th iteration of the stochastic gradient procedures we are dealing with has the form

$$x_{k+1} = x_k - \alpha_k g_k, \quad (5)$$

where α_k is a positive steplength. By construction, $\{x_k\}$ is a stochastic process whose behavior depends on the randomly selected samples.

Choosing the size of the sample set and the steplength along the iterations clearly represents the main issue in the realization of subsampled first-order methods and characterizes the procedures. Since there is a large variety of approaches, classifying the large number of methods in the literature on the basis of their features is not a trivial task. In this work, we cast renowned stochastic first-order procedures into two groups along the following arguments. Methods in the first group employ subsampled gradient estimates on very small batch sizes (in some approaches full gradient evaluations are occasionally performed) and do not perform checks for acceptance of the new iterate x_{k+1} , i.e., the computed step is accepted in every iteration. Consequently, the computational cost per iteration is low, and their implementation is simple. The original idea can be traced back to Robbins and Monro (Robbins et al. 1951), who proposed the famous Stochastic Approximation method. With careful and problem-dependent choices of the steplength sequence $\{\alpha_k\}$, theoretical results establish the behavior of the expected function values and gradient norm values. Methods (Andradottir 1996; Delyon and Juditsky 1993; Kesten 1958; Kiefer 1952; Krejić et al. 2013, 2015; Nemirovski et al. 2009; Robbins et al. 1951; Spall 2003; Tan et al. 2016; Yousefian et al. 2012; Xu et al. 2012) belong to such class. The performance of these methods is sensitive to the steplength selection and to stochastic variance reduction techniques (Defazio et al. 2014; Johnson et al. 2013; Kingma and Ba 2015; Nguyen et al. 2017; Schmidt et al. 2017).

Methods in the second class rely on machinery from standard globally convergent optimization procedures such as line search, trust-region, or adaptive overestimation strategies (Bellavia et al. 2019, 2020c; Birgin et al. 2018; Blanchet et al. 2019; Cartis et al. 2018; Chen et al. 2018; Curtis et al. 2019; Krejić et al. 2016; Krejić N et al. 2013; Krejić et al. 2015; Paquette et al. 2020; Tripuraneni et al. 2018) and have been proposed with the aim of overcoming the need of problem-dependent steplengths. In fact, by using subsampled function and gradient estimates, steplength selection is adaptive and made on the basis of some globalization strategy and knowable

quantities. The choice of the sample size can vary from simple heuristics to sophisticated schemes that take into account the progress made by the optimization process itself. A further relevant distinction from the methods in the first group is that, except for Curtis et al. (2019), the accuracy of the function and gradient estimates is controlled adaptively along the iterations and plays a central role in the convergence analysis. Assuming that the variance of random functions and gradients is bounded, specific accuracy requirements can be fulfilled by means of a sufficiently large sample size estimated using probabilistic arguments (Bellavia et al. 2019; Tripuraneni et al. 2018; Tropp 2015). Some approaches Bellavia et al. (2020c), Birgin et al. (2018), Krejić N et al. (2013); Krejić et al. (2015); Krejić et al. (2016) reach eventually full precision functions and gradients, and thus the convergence results are deterministic; in the remaining methods, convergence is stated in terms of probability statements, either in mean square or almost sure.

The work is organized as follows. In section “Convolutional Neural Networks”, we briefly introduce neural networks and convolutional neural networks which are widely used for image processing tasks. In section “Stochastic Gradient and Variance Reduction Methods”, we describe subsampled first-order methods in the first group, while in section “Gradient Methods with Adaptive Steplength Selection Based on Globalization Strategies” we present methods belonging to the second group. Finally, in section “Numerical Experiments”, we solve a classification problem of images, discussing the neural network used, implementation issues, and results obtained with some of the methods presented. All norms in the paper are Euclidean $\|\cdot\| \stackrel{\text{def}}{=} \|\cdot\|_2$ and given a random variable A ; the symbols $Pr(A)$ and $E[A]$ denote the probability and expected value of A , respectively.

Convolutional Neural Networks

Neural networks (NNs) have become a state-of-the-art methodology for classification and regression tasks in artificial intelligence field (Bishop 2006; Hastie et al. 2001). NNs are used to approximate functions $\phi : \mathbb{R}^s \rightarrow \mathbb{R}^t$ whose value is known only at a given set of points $\mathbf{d}_i \in \mathbb{R}^s$, $i = 1, \dots, N$. Letting $\hat{\mathbf{y}}_i = \phi(\mathbf{d}_i)$ for $i = 1, \dots, N$, the pairs $\{(\mathbf{d}_i, \hat{\mathbf{y}}_i)\}_{i=1, \dots, N} \in \mathbb{R}^s \times \mathbb{R}^t$, are available and can be used to train the neural network that is supposed to approximate values of $\phi(\mathbf{d})$ for $\mathbf{d} \neq \mathbf{d}_i$, $i = 1, \dots, N$.

A neural network is a model which is typically represented by a network diagram as the one in Fig. 1. It consists of layers L_1, \dots, L_m , $m \geq 2$; L_1 is called *input layer*, L_m is the *output layer*, and, when $m > 2$, L_2, \dots, L_{m-1} are called *hidden layers*. Each layer L_i contains a finite number n_i of neurons, subject to the constraints $n_1 = s$, $n_m = t$. Given an input data $\mathbf{d} \in \mathbb{R}^s$, the neural network returns an output vector in \mathbb{R}^t .

Given an input data $\mathbf{d} \in \mathbb{R}^s$, a neuron of a NN is modeled as shown in Fig. 2. Let $\mathbf{v}_i = (v_{i,1}, \dots, v_{i,n_i})^T \in \mathbb{R}^{n_i}$ be the output of layer L_i and $\boldsymbol{\sigma}_i = (\sigma_{i,1}, \dots, \sigma_{i,n_i})^T \in \mathbb{R}^{n_i}$ contain the *activation functions* $\sigma_{i,j} : \mathbb{R} \rightarrow \mathbb{R}$. Thus, the output of the j th neuron of the layer L_i , for $i = 2, \dots, m$ is the scalar

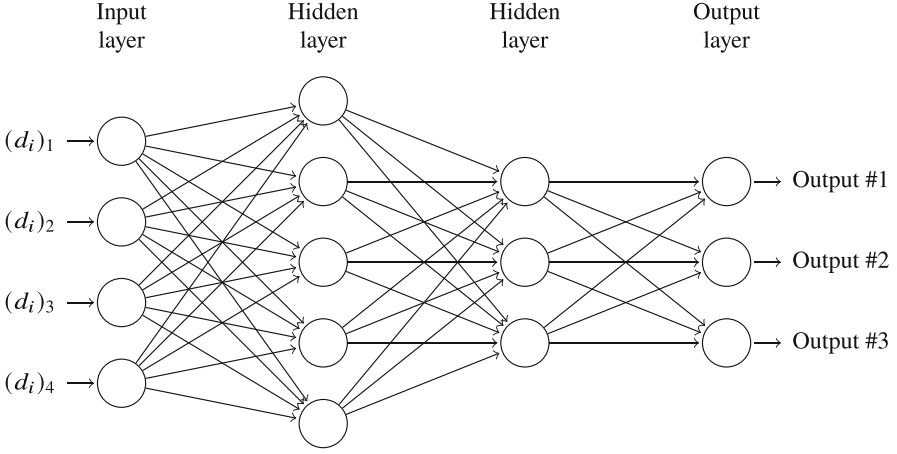


Fig. 1 An example of neural network with two hidden layers, $s=4$, $t=3$

$$v_{i,j} = \sigma_{i,j} \left(\sum_{k=1}^{n_{i-1}} x_{i,j,k} v_{i-1,k} + b_{i,j} \right), \quad (6)$$

where $b_{i,j} \in \mathbb{R}$ is called *bias* and the parameters $x_{i,j,k}$ are called *weights*. Vector \mathbf{v}_1 coincides with the input data \mathbf{d} . Letting $\mathbf{X}_i \in \mathbb{R}^{n_i} \times \mathbb{R}^{n_{i-1}}$ be the matrix with (j,k) -entry given by $x_{i,j,k}$, for $1 \leq j \leq n_i$, $1 \leq k \leq n_{i-1}$ and $\mathbf{b}_i = (b_{i,1}, \dots, b_{i,n_i})^T \in \mathbb{R}^{n_i}$, the output of the whole layer L_i is

$$\mathbf{v}_i = \sigma_i (\mathbf{X}_i \mathbf{v}_{i-1} + \mathbf{b}_i). \quad (7)$$

In fact, the output of each layer is defined recursively by (7) and depends on the output of the previous layer.

Common examples of activation functions are (Bishop 2006; Goodfellow et al. 2016):

- *Linear*: $\sigma(z) = z$
- *Sigmoid or logistic*: $\sigma(z) = 1/(1 + e^{-z})$
- *Tanh*: $\sigma(z) = \tanh(z)$
- *Relu*: $\sigma(z) = \max(0, z)$
- *Elu*: $\sigma(z) = z \cdot \mathbb{X}_{[x \geq 0]} + (e^z - 1) \cdot \mathbb{X}_{[x < 0]}$

where $\mathbb{X}_I : \mathbb{R} \rightarrow \mathbb{R}$ is the indicator function, defined by

$$\mathbb{X}_I(x) = \begin{cases} 1 & x \in I \subseteq \mathbb{R} \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

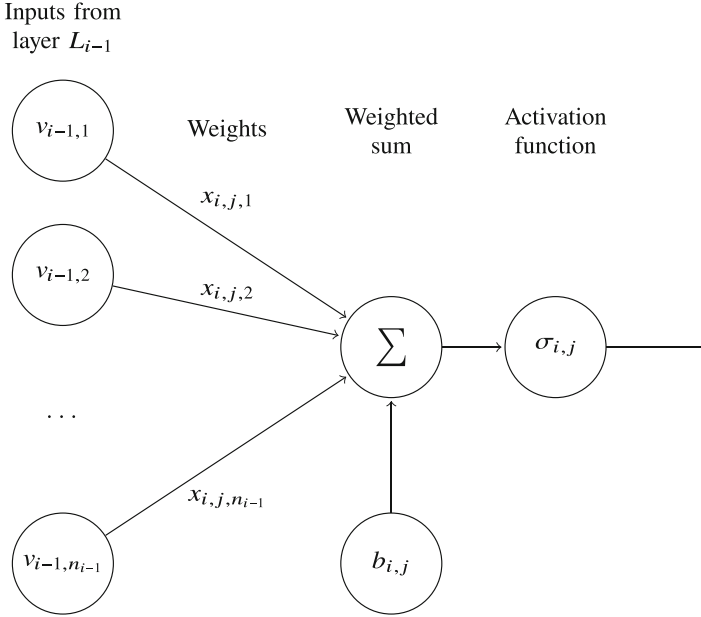


Fig. 2 Mathematical model of j th neuron of layer L_i

The procedure for choosing the parameters $\{(\mathbf{X}_i, \mathbf{b}_i)\}_{i=1,\dots,m}$ is referred to as *training phase*. Let x be the vectorization of $\{(\mathbf{X}_i, \mathbf{b}_i)\}_{i=1,\dots,m}$. Given the set of known data $\{(\mathbf{d}_i, \hat{\mathbf{y}}_i)\}_{i=1,\dots,N}$ (*training set*), the aim is to choose the parameters so that the output $\mathbf{v}_m(x; \mathbf{d}_i)$ of the neural network corresponding to the input \mathbf{d}_i is as close as possible to the value $\hat{\mathbf{y}}_i$ for every $i = 1, \dots, N$.

In order to do that, it is necessary to select a function $\mathbb{E} : \mathbb{R}^t \times \mathbb{R}^t \rightarrow \mathbb{R}$ for measuring the error made by the network on the prediction of each given data and minimize the so-called *loss function*:

$$\frac{1}{N} \sum_{i=1}^N \mathbb{E}(\mathbf{v}_m(x; \mathbf{d}_i), \hat{\mathbf{y}}_i). \quad (9)$$

Since \mathbf{d}_i and $\hat{\mathbf{y}}_i$ are known, the loss function is a special case of (2) where

$$f_i(x) = \mathbb{E}(\mathbf{v}_m(x; \mathbf{d}_i), \hat{\mathbf{y}}_i), \quad \text{for } i = 1, \dots, N.$$

We underline that the minimization of suitable loss functions gives rise to prediction functions that *generalize* information from the available data and avoid *overfitting* of the training set (Bottou et al. 2018, §2).

Convolutional neural networks (CNNs) are a specialized kind of neural network for processing data with a grid-like topology, such as images represented as a two- or three-dimensional grid of pixels. CNNs extract features from the input image which are in some way representative of local neighboring portions of the image. This choice is motivated by the fact that important connections in an image are local (Strang 2019) and that reducing the dimension of weight matrices speeds up the process. This task is achieved exploiting filters commonly used in the computer vision context, such as convolution filter, which are able to extract low level features such as edges, color, and gradient orientation (Forsyth et al. 2002, Chap. 4). These filters are combined with standard neural network layers, so that all the low-level features are combined together. In the following, we give an overview on the main layers used in CNNs and refer the interested reader to Goodfellow et al. (2016, Chap. 9) and (Chollet 2017; Patterson et al. 2017) for additional details.

Convolutional Layer

We consider an image I as a three-dimensional $w \times h \times c$ array, where w is the image width, h is the image height, and c is the number of channels.

Discrete convolution aims to reduce the noise of a signal by applying a weighted average of each entry of the signal and its neighbors. Given an image I sized $w \times h \times c$, an integer $k \geq 1$, a three-dimensional $(2k + 1) \times (2k + 1) \times c$ array W called *kernel*, and a scalar b called *bias*, the discrete convolution between I and W , denoted by $I * W$, is the two-dimensional array defined by

$$(I * W)(i, j) = \sum_s \sum_t \sum_{u=1}^c I(s, t, u) \cdot W(s - i + k + 1, t - j + k + 1, u) + b, \quad (10)$$

for $i = 1, \dots, w - 2k$ and $j = 1, \dots, h - 2k$, where s and t range over all allowed subscripts for I and W , namely, $s = \max\{1, i - k\}, \dots, \min\{i + k, w\}$, $t = \max\{1, j - k\}, \dots, \min\{j + k, h\}$.

The application of a filter to the input yields a two-dimensional array instead of a three-dimensional; see index u in (10). Typically, convolutional layers apply m different filters of the same dimension to the input. Consider m kernels $\{W_\ell\}_{\ell=1, \dots, m}$, each one sized $(2k + 1) \times (2k + 1) \times c$. The output of the convolutional layer is the 3D array defined by

$$(I ** W)(i, j, \ell) = (I * W_\ell)(i, j),$$

where $i = 1, \dots, w - 2k$, $j = 1, \dots, h - 2k$ and $\ell = 1, \dots, m$; thus, every filter adds a channel to the output array. Hence, the output of convolutional layers with m kernels is given by an array of width and length $w - 2k$ and $h - 2k$, respectively, while the new number of channels is equal to the number of filters which have been applied.

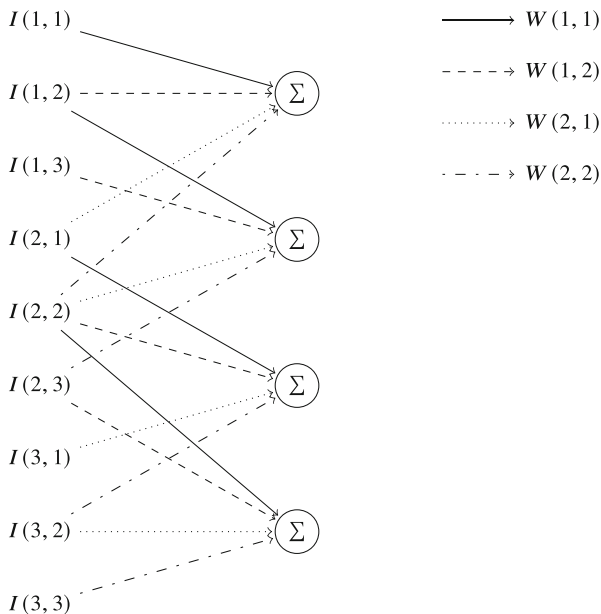


Fig. 3 An example of convolutional layer acting on a $3 \times 3 \times 1$ array I . The kernel W dimension is $2 \times 2 \times 1$. Weights W are shared among different neurons. In this example, the output of the convolutional layer consists of four neurons. Biases and activation functions have been omitted

CNNs are networks composed by at least one convolutional layer and standard layers. In convolutional layers, the entries of the filters are the parameters which are updated during the training. Hence, a convolutional layer consists of $m \cdot ((2k + 1) \cdot (2k + 1) \cdot c + 1)$ trainable parameters, $(2k + 1) \cdot (2k + 1) \cdot c + 1$ for each filter, bias term included. Each element of the array resulting from a convolution can be viewed as a neuron of the type shown in Fig. 2, where some of the connections, corresponding to the indices falling outside the ranges defined in (10), have been dropped (i.e., the corresponding weights are set to 0). In contrast with standard NN layers, convolutional layers share weights among different neurons. The kernel weights are in fact the same in each output neuron, as shown in Fig. 3.

Max Pooling Layer

In order to speed up the training phase by reducing the dimension of the object involved, the max pooling strategy is commonly used in CNN architectures for imaging (Strang 2019). It consists in replacing, for every channel, a square neighborhood with its maximum. More formally, given an image I , max pooling process MP acts as follows:

$$MP(I)(i, j, k) = \max_{(s,t) \in S(i,j)} I(s, t, k), \quad (11)$$

where $S(i, j)$ is a neighborhood of (i, j) .

The square neighborhood is defined by mean of two hyperparameters $\chi_{sl} \geq \chi_{st}$, which are the *spatial extent*, the length of the square edge, and the *stride*, the step which is used to move the square around the image, respectively. When $\chi_{sl} > \chi_{st}$, we talk about *overlapping* pooling. By construction, the max pooling layer does not call for parameters to be trained, and the dimension of the output of MP is smaller than that of the input and given by

$$((w - \chi_{sl})/\chi_{st} + 1) \times ((h - \chi_{sl})/\chi_{st} + 1) \times c,$$

where $w \times h \times c$ is the input dimension. This strategy can be viewed also as a *downsampling* in order to mitigate overfitting during the training.

Stochastic Gradient and Variance Reduction Methods

In this section, we present the widely used stochastic gradient descent (SGD) method (Robbins et al. 1951) and incremental gradient algorithms based on variance reduction such as stochastic variance reduction gradient (SVRG) method (Johnson et al. 2013), SVRG method with Barzilai-Borwein steplengths (SVRG - BB) (Tan et al. 2016), StochAstic Recursive grAdient algoritHm (SARAH) method (Nguyen et al. 2017), stochastic average gradient (SAG) method (Schmidt et al. 2017), and SAGA (Defazio et al. 2014). In the presentation of the convergence properties of these methods, we will make use of the specific form (2) of the problem and of following assumptions.

Assumption 1. Each function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ has Lipschitz continuous gradient, i.e., there exists a constant $L \geq 0$ such that

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\| \quad x, y \in \mathbb{R}^n.$$

This assumption clearly implies that the gradient of objective function is also L -Lipschitz continuous:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad x, y \in \mathbb{R}^n.$$

Assumption 2. The function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is μ strongly convex, i.e., there exists a constant $\mu > 0$ such that

$$f(x) \geq f(y) + (\nabla f(y))^T(x - y) + \frac{\mu}{2}\|x - y\|^2 \quad \text{for all } (x, y) \in \mathbb{R}^n \times \mathbb{R}^n. \quad (12)$$

In case of convex (strongly convex) problems, we denote x_* an (the unique) optimal solution.

The standard gradient descent GD method employing the full (true) gradient (FG) is defined by the following iterative formula:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

The steplength α_k can be fixed in a number of ways, for example, one can apply a line search procedure based on specific requirements on f or take a constant value, $\alpha_k = \alpha, \forall k \geq 0$. If f is convex and Assumption 1 holds, method FG with fixed steplength α converges sublinearly and satisfies the following error bound:

$$f(x_k) - f(x_*) = \mathcal{O}(1/k),$$

provided that $0 < \alpha < 2/L$ (Nesterov 1998, Th. 2.1.13). If additionally f is strongly convex and $0 < \alpha < 2/(\mu + L)$, then FG achieves linear convergence:

$$f(x_k) - f(x_*) = \mathcal{O}(\rho^k),$$

with ρ depending on the condition number L/μ (Nesterov 1998, Th. 2.1.14).

In the case where the number of component functions f_i is large, such as in machine learning applications, the computation of the full gradient is very expensive, and SGD (stochastic gradient descent) appears as an appealing alternative. The method was first proposed in the seminal paper of Robbins and Monro as SA (stochastic approximation) method (Robbins et al. 1951). The main idea of SGD is to replace the expensive gradient $\nabla f(x_k)$ with a significantly cheaper stochastic vector g_k . Here we focus on the case where g_k is an unbiased approximation to $\nabla f(x_k)$, i.e., $E[g_k] = \nabla f(x_k)$, built via (4) with $\mathcal{S}_{k,g}$ chosen uniformly at random from $\{1, \dots, N\}$.

Intuition for using subsampled functions evaluated on random small size sample sets comes from the fact that the training set is often highly redundant, see, e.g., (Bottou et al. 2018). Sample sets $\mathcal{S}_{k,g}$ with small cardinality $|\mathcal{S}_{k,g}|$, in the limit equal to one, are generally used. Whenever $|\mathcal{S}_{k,g}| > 1$, the stochastic approximation of the full gradient is denoted as *mini-batch*; on the other hand, if the sample set reduces to a single element, the stochastic approximation is called *simple* or *basic*. In the following algorithm, without loss of generality, we present SGD referring to the latter case.

ALGORITHM SGD

Step 0: Initialization. Choose an initial point x_0 and a sequence of strictly positive steplengths $\{\alpha_k\}$. Set $k = 0$.

Step 1. Stochastic gradient computation. Choose randomly and uniformly $i_k \in \{1, \dots, N\}$. Set $g_k = \nabla f_{i_k}(x_k)$.

(continued)

Step 2. Iterate computation. Set $x_{k+1} = x_k - \alpha_k g_k$. Increment k by one and go to Step 1.

Since $\{x_k\}$ is a stochastic process whose behavior depends on the random variables $\{i_k\}$, convergence analysis has to be carried out in expectation. Given that one iteration of SGD requires a single gradient $\nabla f_{i_k}(x_k)$, each iteration of the SGD method is significantly cheaper than FG method. Due to the variance introduced by the approximations g_k , in case of fixed steplength, it is not possible to prove convergence of the method to the solution even in the strongly convex case. On the other hand, it can be proved that if there exist positive scalars M_1 and M_2 such that at each iteration of SGD

$$E[\|g_k\|^2] \leq M_1 + M_2 \|\nabla f(x_k)\|^2, \quad (13)$$

and if $\alpha \leq \mu/(LM_2)$, then the expected optimality gap $f(x_k) - f(x_*)$ falls below a problem-dependent value (Bottou et al. 2018, Th. 4.6).

Convergence in expectation can be proved assuming to employ diminishing steplengths, i.e., the sequence $\{\alpha_k\}$ satisfies $\sum_{k=1}^{\infty} \alpha_k = \infty$, $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$. It can be shown (see Nemirovski et al. (2009, p. 1578)) that for strongly convex functions, properly chosen steplengths such as $\alpha_k = \theta/k$ with $\theta > 1/(2\mu)$, and random gradient approximations having bounded variance, one can get

$$E[\|x_k - x_*\|] = \mathcal{O}(1/\sqrt{k}).$$

A further result on expected optimality gap for strongly convex functions is given below.

Theorem 1 (Bottou et al. 2018, Th. 4.7). *Suppose that Assumptions 1 and 2 hold and let x_* be the minimizer of f . Assume that (13) holds at each iteration. Then, if SGD is run with $\alpha_k = \frac{\beta}{\gamma+k}$, $\beta > \frac{1}{\mu}$ and $\gamma > 0$ such that $\alpha_1 \leq \frac{1}{LM_2}$, there exists a scalar $\nu > 0$ such that*

$$E[f(x_k)] - f(x_*) \leq \frac{\nu}{\gamma + k}. \quad (14)$$

The theorem above shows that, in the case of strongly convex problems, SGD converges slower (sublinearly) than FG method and this depends on the variance of the random sampling. Note that the larger M_2 is, the smaller the steplength is, and this implies slow convergence.

Theoretical results for SGD applied to nonconvex optimization problems are available (Bottou et al. 2018, §4.3). In particular, if f is bounded, in expectation $-g_k$ is a direction of sufficient descent for f at x_k and SGD is applied with diminishing steplengths $\{\alpha_k\}$ satisfying $\sum_{k=1}^{\infty} \alpha_k = \infty$, $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, then it can be shown

that the expected gradient norms cannot stay bounded away from zero (Bottou et al. 2018, Th. 4.9).

If the approximate gradient g_k has a large variance, SGD may show slow convergence and bad performance. Taking a larger sample size for $\mathcal{S}_{k,g}$ could help to reduce gradient variance, but large sample may deteriorate the overall computational efficiency of stochastic gradient optimization. In order to improve convergence with respect to SGD, stochastic variance reduction methods have been proposed, see, e.g., Defazio et al. (2014), Johnson et al. (2013), Nguyen et al. (2017), Tan et al. (2016), Schmidt et al. (2017), Wang et al. (2013). In particular, in Wang et al. (2013), a variance reduction technique is proposed by making use of control variates (Ross 2006) to augment the gradient approximation and consequently reduce its variance.

Variance reduction is the core of SVRG (stochastic variance reduction gradient) method presented in Johnson et al. (2013); the algorithm is given below.

ALGORITHM SVRG

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, an inner loop size $m > 0$, a steplength $\alpha > 0$, and the option for the iterate update. Set $k = 1$.

Step 1: Outer iteration, full gradient evaluation.

Set $\tilde{x}_0 = x_{k-1}$. Compute $\nabla f(\tilde{x}_0)$.

Step 2: Inner iterations

For $t = 0, \dots, m - 1$

Uniformly and randomly choose $i_t \in \{1, \dots, N\}$.

Set $\tilde{x}_{t+1} = \tilde{x}_t - \alpha(\nabla f_{i_t}(\tilde{x}_t) - \nabla f_{i_t}(\tilde{x}_0) + \nabla f(\tilde{x}_0))$.

Step 3: Outer iteration, iterate update.

Set $x_k = \tilde{x}_m$ (Option I). Increment k by one and go to Step 1.

Set $x_k = \tilde{x}_t$ for randomly chosen $t \in \{0, \dots, m - 1\}$ (Option II). Increment k by one and go to Step 1.

SVRG consists of outer and inner iterations. At each outer iteration k , the full gradient at x_k is computed. Then a prefixed number m of inner iterations is performed using stochastic gradients and fixed steplength α ; the internal iterates are $\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_m$. At the t th inner iteration, the stochastic gradient used has the form

$$\nabla f_{i_t}(\tilde{x}_t) - \nabla f_{i_t}(\tilde{x}_0) + \nabla f(\tilde{x}_0),$$

with i_t chosen uniformly and randomly in $\{1, \dots, N\}$. This quantity is an unbiased estimation of the gradient. Finally, the new iterate is either the last computed iterate \tilde{x}_m (Option I) or one of the vectors $\tilde{x}_0, \dots, \tilde{x}_{m-1}$ (Option II). Although Option I, taking the new iterate as the last outcome of inner loop, is intuitively more appealing, the convergence results from Johnson et al. (2013) are valid for Option II only. The

results presented in Johnson et al. (2013) cover both the convex and nonconvex cases. For the sake of simplicity, here, we consider the strongly convex case.

Theorem 2 (Johnson et al. 2013, Th 1). *Suppose that Assumptions 1 and 2 hold and that all f_i are convex, and let x_* be the minimizer of f . If m and α satisfy*

$$\theta = \frac{1}{\mu\alpha(1-2L\alpha)m} + \frac{2L\alpha}{1-2L\alpha} < 1, \quad (15)$$

then Algorithm SVRG with Option II generates a sequence such that

$$E[f(x_k)] - f(x_*) \leq \theta^k (f(x_0) - f(x_*)).$$

The above statement clearly demonstrates that convergence in expectation depends on m and α and it is guaranteed taking both a sufficiently large loop size m and a sufficiently small steplength α . Note that θ in (15) depends on the scalars L and μ and condition (15) imposes the following restrictions to α and m : $\alpha < 1/(4L)$ and $m > 2/(\mu\alpha)$.

The linear convergence in expectation of the sequence of the iterates generated by the same algorithm with Option I has been proved later in Tan et al. (2016), and it is given below.

Theorem 3 (Tan et al. 2016, Corollary 1). *Suppose that Assumptions 1 and 2 hold and let x_* be the minimizer of f . If m and α satisfy*

$$\theta = (1 - 2\alpha\mu(1 - \alpha L)^m) + \frac{4\alpha L^2}{\mu(1 - \alpha L)} < 1,$$

then Algorithm SVRG with Option I generates a sequence which converges linearly in expectation

$$E[\|x_k - x_*\|^2] \leq \theta^k \|x_0 - x_*\|^2.$$

The value of m is most often of order $\mathcal{O}(n)$; in Johnson et al. (2013), it is suggested to take $m = 2n$ for convex problems and $m = 5n$ for nonconvex problems. Numerical studies that concentrate on the influence of m and α are available in Tan et al. (2016) as well as the comparison with the method of SVRG type employing adaptive steplengths. Further, in practical applications, it can be convenient to replace the full gradient at outer iterations with a mini-batch stochastic gradient. Application of SVRG to nonconvex problems is briefly discussed in Johnson et al. (2013, §3). Notice that SVRG requires the full gradient which is stored in memory during the whole inner loop execution. Instead of storing all gradients $\nabla f_i(\tilde{x}_0)$ separately, at each inner iteration $\nabla f_i(\tilde{x}_0)$ is evaluated along with $\nabla f_i(\tilde{x}_t)$; this increases the computational cost but reduces the memory requirement drastically. In applications where gradient evaluation is very expensive, the full

gradient is typically replaced with a mini-batch stochastic gradient (Lei et al. 2017). Further, we mention a limited memory approach which gives rise to k -SVRG (Raj et al. 2018).

A variant of SVRG borrows ideas from the spectral gradient method (Barzilai et al. 1988; Raydan et al. 1997) which is very popular modification of the classical FG. The spectral gradient method is based on the idea of approximating the Hessian matrix in each iteration with a multiple of the identity matrix which minimizes the discrepancy from the secant equation and yields an adaptive steplength in each iteration of the gradient method. This steplength is known as Barzilai-Borwein steplength or the spectral coefficient. The adaptive steplengths overcome hand-tuning and do not need to be small, i.e., of order $1/L$ when the Lipschitz constant is large. Therefore, it is reasonable to expect that some advantages of similar type might be expected in the framework of SGD and SVRG methods. The following algorithm is developed in Tan et al. (2016), introducing the Barzilai-Borwein steplengths in the SVRG framework.

ALGORITHM SVRG - BB

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, an inner loop size $m > 0$, an initial steplength $\alpha_0 > 0$. Set $k = 1$.

Step 1: Outer iteration, full gradient evaluation.

Set $\tilde{x}_0 = x_{k-1}$. Compute $\nabla f(\tilde{x}_0)$.

If $k > 0$, then set $\alpha_k = \frac{1}{m} \frac{\|x_k - x_{k-1}\|^2}{(x_k - x_{k-1})^T (\nabla f(x_k) - \nabla f(x_{k-1}))}$

Step 2: Inner iterations

For $t = 0, \dots, m - 1$

Uniformly and randomly choose $i_t \in \{1, \dots, N\}$.

Set $\tilde{x}_{t+1} = \tilde{x}_t - \alpha_k (\nabla f_{i_t}(\tilde{x}_t) - \nabla f_{i_t}(\tilde{x}_0) + \nabla f(\tilde{x}_0))$

Step 3: Outer iteration, iterate update. Set $x_k = \tilde{x}_m$. Increment k by one and go to Step 1.

Note that at the first outer iteration, the steplength is the input data α_0 , while at the successive outer iterations, the steplengths α_k are adaptively chosen and used within inner iterations. The following results are established for strongly convex functions.

Theorem 4 (Tan et al. 2016, Th. 3.8). *Suppose that Assumptions 1 and 2 hold and let x_* be the minimizer of f . Define $\theta = (1 - e^{-2\mu/L})/2$. If m is chosen such that*

$$m > \max \left\{ \frac{2}{\log(1 - 2\theta) + 2\mu/L}, \frac{4L^2}{\theta\mu^2} + \frac{L}{\mu} \right\},$$

then SVRG-BB converges linearly in expectation

$$E[\|x_k - x_*\|^2] < (1 - \theta)^k \|\tilde{x}_0 - x_*\|^2.$$

A number of practical issues regarding the application of variance reduction gradient methods is considered in the literature. All of these methods compute the full gradient at each outer iteration, and this represents the main cost of these algorithms. Results presented in Babanezhad et al. (2015) show that it is possible to perform the outer iterations with increasing batch size for the gradient approximation without compromising the linear convergence rate. Mini-batch methods in inner loop iterations are also considered in Babanezhad et al. (2015).

SAG (Schmidt et al. 2017) method is based on average gradient approximation, which represent an alternative to the gradient estimators previously described. The main idea is to accumulate previously computed stochastic gradient values. The basic version of SAG method Schmidt et al. (2017) is presented in the algorithm below.

ALGORITHM SAG

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, positive steplengths $\{\alpha_k\}$, $y_i = 0$, for $i = 1, \dots, N$. Set $k = 0$.

Step 1: Stochastic gradient update. Uniformly and randomly choose $i_k \in \{1, \dots, N\}$. Set $y_{i_k} = \nabla f_{i_k}(x_k)$.

Step 2: Iterate update. Set $x_{k+1} = x_k - \frac{\alpha_k}{N} \sum_{i=1}^N y_i$. Increment k by one and go to Step 1.

SAG method uses a gradient estimation for $\nabla f(x_k)$ composed of the sum along all terms in the gradient, in the spirit of FG, but the cost of each iteration is the same as SDG. Remarkably, at the price of keeping track of a $N \times n$ matrix containing the gradient values computed through the iterations, SAG achieves almost the same convergence rate than FG. In fact, unlike SDG, convergence of SAG can be achieved taking constant steplength $\alpha_k = 1/(16L)$, $\forall k \geq 0$ and the optimality gap on average iterates achieve the same error bound $\mathcal{O}(1/k)$ as FG for convex function and linear convergence for strongly convex functions (Schmidt et al. 2017, Th. 1). If the Lipschitz constant is not available, a strategy for its estimation is given in Schmidt et al. (2017, §4.6). The following result concerns strongly convex problems.

Theorem 5 (Schmidt et al. 2017, Th. 1). *Suppose that Assumptions 1 and 2 hold. Let x_* be the minimizer of f . If $\alpha = 1/(16L)$, then*

$$E[f(x_k)] - f(x_*) \leq \left(1 - \min\left\{\frac{\mu}{16L}, \frac{1}{8N}\right\}\right)^k C_0,$$

where $C_0 > 0$ depends on x_* , x_0 , f , L , N .

Note that for ill-conditioned problems where $N < (2L)/\mu$, N does not play a role in the convergence rate, and the SAG algorithm has nearly the same convergence rate as the FG method with a step size of $1/(16L)$, even though it uses iterations which are N times cheaper. This indicates that in case of ill-conditioned problems, the convergence rate is not affected by the use of out-of-date gradient values. A SAG extension, called SAGA, has been also proposed in Defazio et al. (2014). SAGA exploits SVRG-like unbiased approximations of the gradient and combines ideas of SAG and SVRG algorithms; a fixed steplength is employed. The interested reader can find additional details about SAGA in Defazio et al. (2014).

SARAH method Nguyen et al. (2017) is a further variant of SGD based on accumulated stochastic information. Unlike SAGA, SARAH is based on the idea of variance reduction and biased estimations of the gradient; the algorithm is sketched below.

ALGORITHM SARAH

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, an inner loop size $m > 0$, a steplength $\alpha > 0$. Set $k = 1$.

Step 1: Outer iteration, full gradient evaluation.

Set $\tilde{x}_0 = x_{k-1}$. Compute $y_0 = \nabla f(\tilde{x}_0)$. Set $\tilde{x}_1 = \tilde{x}_0 - \alpha y_0$.

Step 2: Inner iterations.

For $t = 1, \dots, m - 1$

Uniformly and randomly choose $i_t \in \{1, \dots, N\}$.

Compute $y_t = \nabla f_{i_t}(\tilde{x}_t) - \nabla f_{i_t}(\tilde{x}_{t-1}) + y_{t-1}$.

Set $\tilde{x}_{t+1} = \tilde{x}_t - \alpha y_t$.

Step 3: Outer iteration, iterate update. Set $x_k = \tilde{x}_t$ for randomly chosen $t \in \{0, \dots, m\}$. Increment k by one and go to Step 1.

As already mentioned, y_t is a biased estimator of the gradient as

$$E[y_t] = \nabla f(\tilde{x}_t) - \nabla f(\tilde{x}_{t-1}) + y_{t-1} \neq \nabla f(\tilde{x}_t).$$

The convergence results presented in Nguyen et al. (2017) cover both the convex and strongly convex cases, as well as address complexity analysis; the result for the strongly convex case is given below.

Theorem 6 (Nguyen et al. 2017, Th. 4). *Suppose that Assumptions 1 and 2 hold and that each function f_i , $1 \leq i \leq N$ is convex. If α and m are such that*

$$\sigma = \frac{1}{\mu\alpha(m+1)} + \frac{\alpha L}{2 - \alpha L} < 1, \quad (16)$$

then the sequence $\{\|\nabla f(x_k)\|\}$ generated by Algorithm SARAH satisfies

$$E[\|\nabla f(x_k)\|^2] \leq \sigma^k \|\nabla f(x_0)\|^2.$$

We observe that condition (16) imposes the upper bound $1/L$ on the steplength α , while the analogous condition (15) governing the convergence of SVRG imposes the tighter bound $\alpha < 1/(4L)$; further, for any α and m , it holds $\sigma < \theta$. An additional advantage of SARAH is that if α is small enough, then the stochastic steps computed converge linearly in the inner loop in expectation.

Theorem 7 (Nguyen et al. 2017, Th. 1b). *Suppose that Assumption 1 holds and each function f_i , $1 \leq i \leq N$ is μ -strongly convex with $\mu > 0$. If $\alpha \leq 2/(\mu + L)$, then for any $t \geq 1$*

$$E[\|y_t\|^2] \leq \left(1 - \frac{2\mu L\alpha}{\mu + L}\right) E[\|y_{t-1}\|^2] \leq \left(1 - \frac{2\mu L\alpha}{\mu + L}\right)^t E[\|\nabla f(\tilde{x}_0)\|^2].$$

Gradient Methods with Adaptive Steplength Selection Based on Globalization Strategies

Gradient methods discussed in the previous section employ stochastic (possibly and occasionally full) gradient estimates and do not rely on any machinery from standard globally convergent optimization procedures such as line search, trust-region, or adaptive overestimation strategies. On the other hand, a few and recent papers (Bellavia et al. 2019, 2020c; Blanchet et al. 2019; Curtis et al. 2018; Chen et al. 2018; Curtis et al. 2019; Paquette et al. 2020) rely on such strategies for selecting the steplength and part of them mimic traditional step acceptance rules using stochastic estimates of functions and gradients. The purpose of these methods is to partially overcome the dependence of the steplengths from the Lipschitz constant of the gradient, i.e., lack of natural scaling, which appears in the convergence results of SGD and its variants given in section “[Stochastic Gradient and Variance Reduction Methods](#)”; see Curtis et al. (2019, §1).

One relevant proposal in the field of stochastic trust-region methods is TRiSh (Trust-Region-*ish*) algorithm (Curtis et al. 2019). TRiSh uses a stochastic gradient estimate g_k of $\nabla f(x_k)$ and a careful steplength selection which, to a certain extent, mimics a trust-region strategy. TRiSh algorithm is sketched below.

ALGORITHM TRISH

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, positive steplengths $\{\alpha_k\}$, positive $\{\gamma_{1,k}\}$ and $\{\gamma_{2,k}\}$ such that $\gamma_{1,k} > \gamma_{2,k}$, $\forall k \geq 0$. Set $k = 0$.

Step 1: Step computation. Compute a gradient estimate $g_k \in \mathbb{R}^n$.

(continued)

Step 2: Steplength selection. Set

$$s_k = \begin{cases} -\gamma_{1,k}\alpha_k g_k & \text{if } \|g_k\| \in \left[0, \frac{1}{\gamma_{1,k}}\right) \\ -\alpha_k \frac{g_k}{\|g_k\|} & \text{if } \|g_k\| \in \left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right] \\ -\gamma_{2,k}\alpha_k g_k & \text{if } \|g_k\| \in \left(\frac{1}{\gamma_{2,k}}, \infty\right). \end{cases} \quad (17)$$

Set $x_{k+1} = x_k + s_k$, increment k by one, and go to Step 1.

The relationship between the norms of $s_k = x_{k+1} - x_k$ and g_k is shown in Fig. 4. The norm of the step, as function of the norm of the stochastic gradient, is continuous. When $\|g_k\| \in \left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right]$, the step s_k can be viewed as a trust-region problem since it solves the trust-region problem:

$$\min_{\|s\| \leq \alpha_k} f(x_k) + g_k^T s. \quad (18)$$

If the norm of the stochastic gradient is below $1/\gamma_{1,k}$, then the steplength is $\gamma_{1,k}\alpha_k$, while if the norm is larger than $1/\gamma_{2,k}$, then the steplength is $\gamma_{2,k}\alpha_k$ with $\gamma_{2,k} < \gamma_{1,k}$. Note that the trust-region machinery is used for building the step, but unlike standard trust-region strategies, it does not employ step acceptance conditions and therefore it does not affect the choice of the steplengths $\{\alpha_k\}$. Examples in Curtis et al. (2019, §2) show that a pure trust-region algorithm, taking steps from (18) independently of the norm of the stochastic gradient, is not guaranteed to converge; this would be the case if $\gamma_{1,k} \gg 0$ and $\gamma_{2,k} \approx 0$. Hence, the convergence theory of TRish is based on an appropriate upper bound for $\gamma_{1,k}/\gamma_{2,k}$. The theoretical results for TRish are similar to those of SGD since both methods take steps along the stochastic gradient; on the other hand, SGD possesses no natural scaling, while

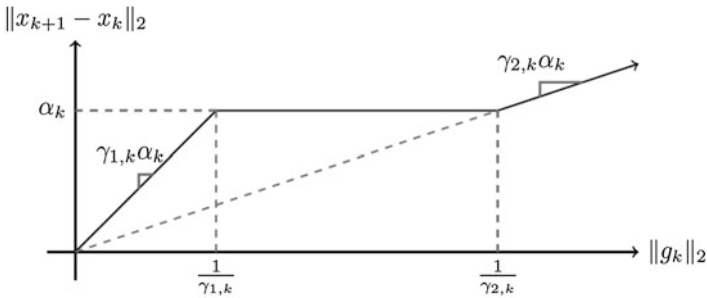


Fig. 4 Relationship between $\|x_{k+1} - x_k\|$ and $\|g_k\|$

TRiSh exploits normalized steps whenever $\|g_k\| \in \left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right]$. This issue can be interpreted as an adaptive choice of the steplength which is $\alpha_k/\|g_k\|$ instead of α_k itself; it is expected to improve numerical performance upon traditional SGD, and this is confirmed by the numerical results provided in Curtis et al. (2019, §2) and in the subsequent section “Numerical Experiments”.

We summarize some results from the convergence analysis presented in Curtis et al. (2019). Let us assume that Assumption 1 holds, g_k is an unbiased estimator of $\nabla f(x_k)$ satisfying inequality (13) for any $k \geq 0$, f is bounded below by $f_* = \inf_{x \in \mathbb{R}^n} f(x) \in \mathbb{R}$, and the Polyak-Lojasiewicz condition holds at any $x \in \mathbb{R}^n$ with $\mu > 0$, i.e.,

$$2\mu(f(x) - f_*) \leq \|\nabla f(x)\|^2, \quad \forall x \in \mathbb{R}^n. \quad (19)$$

Note that (19) holds if f is μ -strongly convex.

The first convergence result of TRiSh deals with constant choices for the parameters $\gamma_{1,k} = \gamma_1$, $\gamma_{2,k} = \gamma_2$, and $\alpha_k = \alpha$ for all $k \geq 0$ (Curtis et al. 2019, Theorem 1). Provided that γ_1/γ_2 and α are bounded from above by quantities involving μ , L , and M_1, M_2 in (13), then TRiSh has expected optimality gap:

$$E[f(x_k)] - f_* \leq c_1 + c_2^{k-2}(f(x_0) - f_* - c_1),$$

where $c_1 > 0$ and $c_2 \in (0, 1)$ are scalars depending on α , γ_1 , γ_2 . In fact, using a constant steplength depending on the Lipschitz constant L , the expected optimality gap is guaranteed to be reduced below a given threshold as in SGD. A comparison of the steplength bound in TRiSh with that in the classical SGD method can be found in Curtis et al. (2019, p.207).

Convergence can be proved to be linear if the variance of the stochastic gradient decreases linearly (Curtis et al. 2019, Theorem 4). Specifically, if additionally the stochastic gradient satisfies

$$E\left[\|g_k\|^2\right] \leq c\zeta^{k-1} + \|\nabla f(x_k)\|^2, \quad (20)$$

for all $k \geq 0$ and some $c > 0$, $\zeta \in (0, 1)$, then

$$E[f(x_k)] - f_* \leq \omega\rho^{k-1},$$

where $\omega > 0$ and $\rho \in (0, 1)$. Assumption (20) on gradients can be satisfied if g_k is computed by subsampling with increasing sample size.

A further convergence result covers the cases of sublinearly diminishing steplengths Curtis et al. (2019, Theorem 2) and resembles the corresponding result for SGD method. If the steplengths α_k are sublinearly diminishing, i.e., $\alpha_k = \beta/(\nu + k)$ for some positive β and ν properly chosen, $\gamma_{1,k} = \gamma_1 > 0$, $\gamma_1 - \gamma_{2,k} = \eta\alpha_k$, $\forall k$ and some $\eta \in (0, 1)$, then

$$E[f(x_k)] - f_* \leq \frac{\phi}{\nu + k},$$

for all k , with ϕ positive. We refer to Curtis et al. (2019) for more convergence results, including the case where the Polyak-Lojasiewicz condition is not satisfied.

Other approaches exploit globalization procedures more closely than TRiSh, with the aim of computing the steplength adaptively and testing, at each iteration, some verifiable criterion on progress toward optimality. To establish such control, they need stochastic estimates of functions, in addition to gradient estimates required in all the approaches described so far, and impose dynamic accuracy in stochastic function and gradient approximations. The general scheme for such procedures is given below. We will say that iteration k is successful whenever the acceptance criterion tested in Step 2 is fulfilled, unsuccessful otherwise. Acceptance criteria employed in literature will be presented in the sections “[Stochastic Line Search](#)” and “[Adaptive Regularization and Trust-Region](#)”.

ALGORITHM LSANDTR

Step 0: Initialization. Choose an initial point $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, parameters governing the steplength selection, and the accuracy requirement in gradient and function. Set $k = 0$.

Step 1: Step computation. Compute a gradient estimate $g_k \in \mathbb{R}^n$ and form a step $s_k = -\alpha_k g_k$.

Step 2: Step acceptance. Compute estimates f_k^0 and f_k^s of $f(x_k)$ and $f(x_k + s_k)$ and test for acceptance of $x_k + s_k$. If the iteration is successful, set $x_{k+1} = x_k + s_k$; otherwise, set $x_{k+1} = x_k$.

Step 3: Parameters' update. Compute α_{k+1} and update parameters governing the accuracy requirements in the computation of functions and gradients. Increment k by one and go to Step 1.

The above scheme includes the stochastic line search method proposed in Paquette et al. (2020), the stochastic trust-region method proposed in Blanchet et al. (2019) and Chen et al. (2018), and the adaptive overestimation method proposed in Bellavia et al. (2019). Accuracy in function and gradient approximations is controlled acknowledging that f has a central role since it is the quantity we ultimately wish to decrease. Specifically, it is assumed that f_k^0 , f_k^s , and g_k are sufficiently accurate in probability, conditioned on the past, and an adaptive absolute accuracy for the objective function and an adaptive relative accuracy for the gradient are imposed. These requirements are supposed to be satisfied probabilistically. The method given in Curtis et al. (2018) belongs to the previous framework but uses the exact function in Step 2. Thus, it only imposes adaptive relative accuracy on the gradient.

Accuracy Requirements

As a general setting, let g_k be an estimate of $\nabla f(x_k)$, $\epsilon_g > 0$ be the accuracy requirement, and I_k be the event defined as

$$I_k = \{\|g_k - \nabla f(x_k)\| \leq \epsilon_g\}, \quad \epsilon_g > 0. \quad (21)$$

A gradient estimate g_k is said to be p_g -probabilistically sufficiently accurate whenever

$$\Pr(\mathbb{1}_{I_k} = 1) \geq p_g \quad \text{with } p_g \in (0, 1), \quad (22)$$

with $\mathbb{1}_{I_k} = 1$ if g_k is such that the event I_k holds, $\mathbb{1}_{I_k} = 0$ otherwise.

In a similar way, let f_k^0 and f_k^s be estimates of $f(x_k)$ and $f(x_k + s_k)$, $\epsilon_f > 0$ be the accuracy requirement, and J_k be the event defined as

$$J_k = \{|f_k^0 - f(x_k)| \leq \epsilon_f \quad \text{and} \quad |f_k^s - f(x_k + s_k)| \leq \epsilon_f\}, \quad \epsilon_f > 0. \quad (23)$$

Estimates f_k^0 and f_k^s are said to be p_f -probabilistically sufficiently accurate whenever the event J_k in (23) satisfies the condition

$$\Pr(\mathbb{1}_{J_k} = 1) \geq p_f, \quad \text{with } p_f \in (0, 1). \quad (24)$$

As for problem (2), the computation of f_k^0 , f_k^s and g_k can be performed by averaging functions f_i and gradients ∇f_i in uniformly and randomly selected subsamples of the set $\{1, \dots, N\}$. In order to satisfy (22) and (24) probabilistically, the size of uniform sampling $|\mathcal{S}_{k,f}|$ and $|\mathcal{S}_{k,g}|$ can be bounded below via the Bernstein inequality (Tropp 2015). In particular, in Bellavia et al. (2019, Theorem 6.2) it is shown that given $\epsilon_g > 0$, g_k is p_g -probabilistically sufficiently accurate if the cardinality $|\mathcal{S}_{k,g}|$ of the set $\mathcal{S}_{k,g}$ in (4) satisfies

$$|\mathcal{S}_{k,g}| \geq \min \left\{ N, \left\lceil \frac{2}{\epsilon_g} \left(\frac{V_g}{\epsilon_g} + \frac{2\omega_g(x_k)}{3} \right) \log \left(\frac{n+1}{1-p_g} \right) \right\rceil \right\}, \quad (25)$$

where $E(\|\nabla f_i(x) - \nabla f(x)\|^2) \leq V_g$ and $\max_{i \in \{1, \dots, N\}} \|\nabla f_i(x)\| \leq \omega_g(x)$, or

$$|\mathcal{S}_{k,g}| \geq \min \left\{ N, \left\lceil \frac{4\omega_g(x_k)}{\epsilon_g} \left(\frac{2\omega_g(x_k)}{\epsilon_g} + \frac{1}{3} \right) \log \left(\frac{n+1}{1-p_g} \right) \right\rceil \right\}. \quad (26)$$

Similarly, given $\epsilon_f > 0$, f_k^0 is p_f -probabilistically sufficiently accurate if the cardinality $|\mathcal{S}_{k,f}|$ of the set $\mathcal{S}_{k,f}$ in (3) satisfies

$$|\mathcal{S}_{k,f}| \geq \min \left\{ N, \left\lceil \frac{2}{\epsilon_f} \left(\frac{V_f}{\epsilon_f} + \frac{2\omega_f(x_k)}{3} \right) \log \left(\frac{2}{1-p_f} \right) \right\rceil \right\}, \quad (27)$$

where $E(|f_i(x) - f(x)|^2) \leq V_f$ and $\max_{i \in \{1, \dots, N\}} |f_i(x)| \leq \omega_f(x)$, or

$$|\mathcal{S}_{k,f}| \geq \min \left\{ N, \left\lceil \frac{4\omega_f(x_k)}{\epsilon_f} \left(\frac{2\omega_f(x_k)}{\epsilon_f} + \frac{1}{3} \right) \log \left(\frac{n}{1-p_f} \right) \right\rceil \right\}. \quad (28)$$

It is worth noting that in (25)–(28) failure probabilities $1 - p_f$, $1 - p_g$ appear in the logarithmic terms and therefore their contribution is damped even if they are very small. Specific accuracy requirements made will be specialized in the following subsections.

Stochastic Line Search

A stochastic line search method, which falls into the general scheme LSandTR, is given in Paquette et al. (2020). At iteration k , the computation of the step s_k and the stochastic line search are performed using a constant $\theta \in (0, 1)$ and a positive parameter δ_k . Given α_k , a probability $p_g \in (0, 1)$, a constant $\kappa > 0$, and letting $\epsilon_g = \kappa \alpha_k \|g_k\|$, the gradient estimate g_k formed in Step 1 is supposed to be p_g -probabilistically sufficiently accurate, i.e., to satisfy (22) with $\epsilon_g = \kappa \alpha_k \|g_k\|$.

With g_k at hand, the step s_k in Step 1 takes the form $s_k = -\alpha_k g_k$, and in Step 2 the Armijo condition

$$f_k^s \leq f_k^0 - \theta \alpha_k \|g_k\|^2, \quad (29)$$

is tested for acceptance. This condition is a stochastic variant of Armijo condition (Armijo et al. 1966) as f_k^0 and f_k^s are stochastic estimates of $f(x_k)$ and $f(x_k + s_k)$. Values f_k^0 and f_k^s are supposed to meet two requirements. First, given a probability $p_f \in (0, 1)$ and letting $\epsilon_f = \kappa \alpha_k^2 \|g_k\|^2$, f_k^0 and f_k^s are required to satisfy (24), namely, to be p_f -probabilistically sufficiently accurate with $\epsilon_f = \kappa \alpha_k^2 \|g_k\|^2$. Second, given a constant $\kappa_f > 0$, the sequence of estimates $\{f_k^0, f_k^s\}$ is supposed to satisfy the following variance conditions for all $k \geq 0$:

$$\begin{aligned} E[|f_k^0 - f(x_k)|^2] &\leq \max\{\kappa_f \alpha_k^2 \|\nabla f(x_k)\|^4, \theta^2 \delta_k^4\}, \\ E[|f_k^s - f(x_k + s_k)|^2] &\leq \max\{\kappa_f \alpha_k^2 \|\nabla f(x_k)\|^4, \theta^2 \delta_k^4\}. \end{aligned}$$

Note that both accuracy requirements on functions and gradients are adaptive and the function has to be approximated with higher accuracy than the gradient.

Moreover, observe that the variance condition depends on the parameter δ_k , the steplength α_k , and the norm of the true gradient.

The k th iteration is successful if (29) is met, unsuccessful otherwise. Whenever the iteration is successful, parameters are updated in Step 3 as follows:

$$\alpha_{k+1} = \max\{\gamma\alpha_k, \alpha_{\max}\}$$

$$\delta_{k+1}^2 = \begin{cases} \gamma\delta_k^2 & \text{if } \alpha_k \|g_k\|^2 \geq \delta_k^2 \\ \gamma^{-1}\delta_k^2 & \text{otherwise} \end{cases}$$

for some fixed $\gamma > 1$ and $\alpha_{\max} > 0$. On the other hand, when the iteration is unsuccessful, Step 3 consists in updating

$$\alpha_{k+1} = \gamma^{-1}\alpha_k, \quad \delta_{k+1}^2 = \gamma^{-1}\delta_k^2.$$

The rules for choosing α_k and δ_k either enlarge or reduce accuracy in stochastic estimates based on fulfillment of the decrease condition (29) and the magnitude of the expected improvement of f_k^s over f_k^0 . In fact, the parameter α_k affects the accuracy of gradient and function estimates and is enlarged when the iteration is successful, diminished otherwise. On the other hand, the parameter δ_k affects the variance of function estimates and is intended to guess how much the true function decreases. In fact, the decrease obtained in (29) does not guarantee a similar reduction in the true function as well. Hence, δ_k^2 is enlarged only in the case where the iteration is successful, and $\alpha_k \|g_k\|^2$ is not smaller than δ_k^2 , that is, when the variance of function values is not larger than the square of the decrease in the approximate function. Interestingly, $\alpha_k \|g_k\|$ may not diminish as $\|g_k\|$ decreases and consequently accuracy requirements do not necessarily become more stringent along iterations.

In Paquette et al. (2020) stochastic complexity results have been established for convex, strongly convex, and general nonconvex, smooth problems; they imply convergence results. In case of μ -strongly convex problems, under suitable assumptions on the stochastic process, Paquette et al. (2020, Th. 4.18) shows that there exist probabilities p_g, p_f sufficiently close to one and satisfying $p_g p_f > \frac{1}{2}$ and a constant $\nu \in (0, 1)$ such that the expected number T_ϵ of iterations needed to satisfy

$$f(x_k) - f(x_*) \leq \epsilon$$

is such that

$$E[T_\epsilon] \leq \mathcal{O}(1) \frac{p_g p_f}{2p_g p_f - 1} \frac{(L\kappa\alpha_{\max})^3}{\mu} (\log(\Phi_0) + \log(\epsilon^{-1}))$$

where x_* is the minimizer of f and Φ_0 is a problem-dependent positive scalar. We refer to Paquette et al. (2020) for the complete set of results. As a final comment, the

implementation of the above stochastic line search method encounters the problem that $\epsilon_g = \kappa \alpha_k \|g_k\|$ depends on the norm of the vector g_k that has to be computed. Following Cartis et al. (2018), the computation of the approximated gradient g_k by subsampling can be performed via an inner iterative process. The approximated gradient g_k is computed via (25) or (26) using a predicted sample size. Then, if the predicted accuracy is larger than the required accuracy, the sample size is progressively increased until the accuracy requirement is satisfied.

Adaptive Regularization and Trust-Region

Trust-region and adaptive regularization methods are classes of optimization methods based on a nonlinear steplength control and can be cast into a unifying framework as shown in Toint (2013). Variants of these methods based on estimates for functions and derivatives are proposed in Bellavia et al. (2019), Blanchet et al. (2019), Chen et al. (2018), Wang and Yuan (2019). Here we focus on the case where first-order models are used at each iterations and discuss the adaptive regularization method named AR1DA (Adaptive Regularization with Dynamic Accuracy and first-order model) developed in Bellavia et al. (2019). It shares similarities with STORM, and we refer to Blanchet et al. (2019, §3) for details on this latter algorithm and its stochastic properties. The AR1DA method employs first-order random models with adaptive regularization of order two. The regularization parameter $\sigma_k > 0$ controls the steplength, and a parameter $\omega_k \in (0, 1)$ controls the level of accuracy required in the estimate f_k^0 , f_k^s , and g_k . In fact, the gradient estimate g_k formed in Step 1 is supposed to be p_g -probabilistically sufficiently accurate, with $\epsilon_g = \omega_k \|g_k\|$. Once g_k has been computed, the step s_k in Step 1 is found by minimizing a regularized first-order random model $m_k(s)$ for $f(x_k + s)$ around x_k :

$$\min_{s \in \mathbb{R}^n} m_k(s) = f_k^0 + g_k^T s + \frac{1}{2} \|s\|^2,$$

with f_k^0 being an approximation to $f(x_k)$. Trivially the step takes the form $s_k = -\frac{1}{\sigma_k} g_k$, i.e., $\alpha_k = \frac{1}{\sigma_k}$ in Step 1 of the general scheme LSandTR.

Acceptance of the step is tested using the rules employed in trust-region and regularization methods, but different from the standard approaches, here the function values and the gradient involved are approximated. Using function estimates f_k^0 and f_k^s for $f(x_k)$ and $f(x_k + s_k)$, the test for acceptance is

$$\rho_k = \frac{f_k^0 - f_k^s}{f_k^0 - (f_k^0 + g_k^T s_k)} \geq \eta_1, \quad \eta_1 \in (0, 1). \quad (30)$$

Values f_k^0 and f_k^s are supposed to be p_f -probabilistically sufficiently accurate with $\epsilon_f = \omega_k (g_k^T s_k)$.

Summarizing, the iteration is successful, i.e., the trial point $x_k + s_k$ is accepted as the new iterate, if $\rho_k \geq \eta_1$, unsuccessful otherwise. The updating rule for σ_k and ω_k is

$$\sigma_{k+1} = \begin{cases} \max\{\gamma^{-1}\sigma_k, \sigma_{\min}\} & \text{if } \rho_k \geq \eta_1 \\ \gamma\sigma_k & \text{otherwise} \end{cases},$$

and

$$\omega_{k+1} = \min\left(\kappa_\omega, \frac{1}{\sigma_{k+1}}\right),$$

for some fixed $\gamma > 1$, $\sigma_{\min} > 0$, and $\kappa_\omega \in (0, 1/(2\eta_1))$. Specifically, in case of successful iterations, the regularization parameter is decreased, and the parameter that rules the accuracy requirements is increased. On the other hand, in case of unsuccessful iterations, σ_k is increased and tighter accuracy requirements are imposed on function and gradient approximations.

In Bellavia et al. (2019), complexity analysis in high probability for AR1DA is carried out. Assume for sake of simplicity $p_g = p_f$ and let $\bar{p} \in (0, 1)$ be a prescribed probability for meeting the approximate first-order optimality condition:

$$\|\nabla f(x_k)\| \leq \epsilon, \quad (31)$$

with $\epsilon > 0$. In Bellavia et al. (2019, Th. 7.1), it is shown that if $1 - p_g = \mathcal{O}\left((1 - \bar{p})\epsilon^2/3\right)$, then AR1DA needs at most $\mathcal{O}(\epsilon^{-2})$ iterations and approximate evaluations of the objective function to satisfy (31) with probability at least \bar{p} .

From a practical point of view, the approximated gradient g_k is computed via (25) or (26) using a predicted accuracy requirement, say, ϵ_p . Then, with g_k at hand, if $\epsilon_p > \omega_k \|g_k\|$, then ϵ_p is progressively decreased and g_k recomputed until $\epsilon_p \leq \omega_k \|g_k\|$ or $\epsilon_p < \epsilon$. We finally mention that the algorithm is stopped whenever the condition

$$\|g_k\| \leq \frac{\epsilon}{1 + \omega_k}$$

holds. Remarkably, the accuracy requirement $\epsilon_g = \omega_k \|g_k\|$ guarantees that (31) holds with probability at least p_g .

Numerical Experiments

In this section, we show the performance of three methods previously discussed: SG, SVRG, and TRish applied in the training phase of a CNN. We train a neural network on cifar-10 (Krizhevsky 2009), a classical image recognition dataset. This dataset contains 60000 colored images with a resolution of 32×32 pixels divided

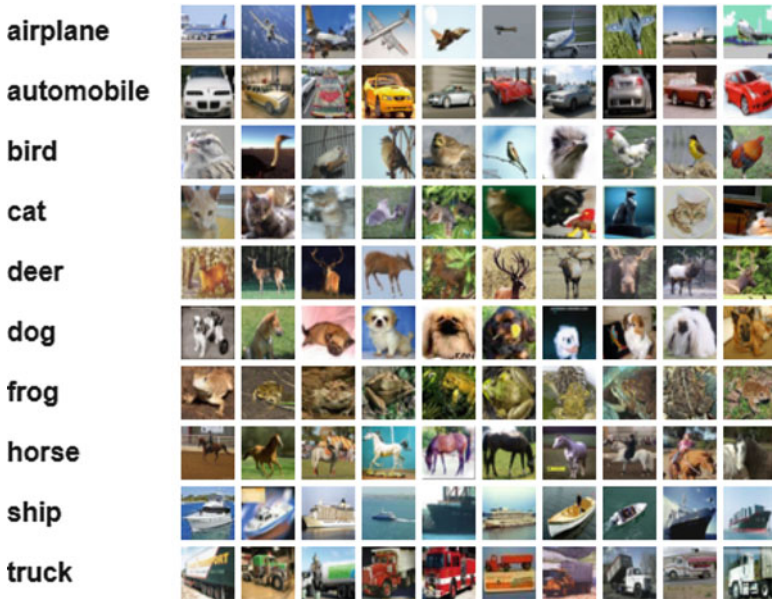


Fig. 5 Some random images from each class of cifar-10 dataset (Image taken from <https://www.cs.toronto.edu/~kriz/cifar.html>)

into a training set (5/6 of the images) and a testing set (1/6 of the images). The images are classified into ten homogeneously distributed classes: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. In Fig. 5, we show some images from the dataset. The color model of cifar-10 images is RGB, i.e., each pixel of an image is represented by three numbers (typically integers) which vary between 0 and 255 and represent the intensity of each channel; hence, the image can be viewed as a $32 \times 32 \times 3$ matrix. It is common to normalize the intensity of each channel between 0 and 1.

The training set is constituted by $N = 50000$ data $\{(\mathbf{d}_i, \hat{\mathbf{y}}_i)\}_{i=1, \dots, N}$, where $\mathbf{d}_i \in \mathbb{R}^{3072}$ is the vector containing the i th image stacked by columns and $\hat{\mathbf{y}}_i \in \mathbb{R}^{10}$ contains value 1 for the actual category of the i th image and 0 for any other category.

The Neural Network in Action

We describe the NN used in our experiments which consists of 14 layers and is displayed in Fig. 6.

The first layer of our network is convolutional (see section “[Convolutional Layer](#)”) with 32 filters and a 3×3 kernel; the activation function is *elu*. The number of filters reshapes the tensor so that the number of channels becomes equal to the number of filters in the convolutional layer. The width and the height of the image

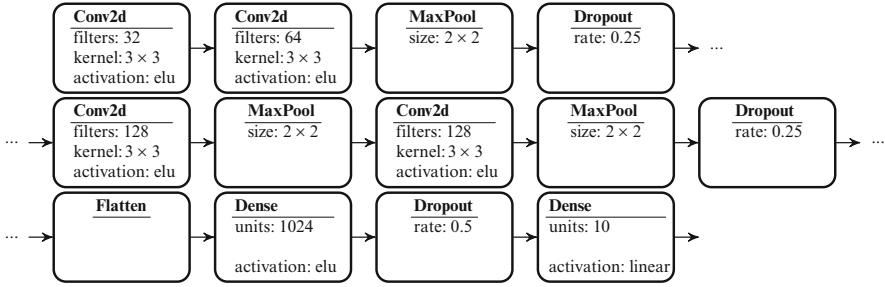


Fig. 6 Architecture of the neural network used for cifar-10. Four convolutional layers mixed with max pooling layers are followed by two dense layers

are changed too, accordingly to section “[Convolutional Layer](#)”, and become both equal to 30. Summarizing, the output of the first layer has size $30 \times 30 \times 32$ and is received by the second layer which is again a convolutional layer with 64 filters, a 3×3 kernel, and *elu* as the activation function. After the second layer, the tensor shape becomes $28 \times 28 \times 64$. The third layer is a max pooling layer (see section “[Max Pooling Layer](#)”), which applies a 2×2 max filter on every channel; this halves the dimension of every slice of the tensor. The fourth layer is a Dropout layer with rate 0.25 which does not alter the shape of the tensor but randomly selects 25% of the values of the tensor and sets them to 0; this phase is commonly performed to avoid overfitting. Next, we apply two times a convolutional layer with 128 filters and a 3×3 kernel followed by a max pooling. After such four layers, a further Dropout layer with rate 0.25 is used; the resulting tensor shape is $2 \times 2 \times 128$. At this stage, the process for transforming the tensor into an array of probabilities is started. First, a Flatten layer vectorizes the $2 \times 2 \times 128$ tensor and returns a one-dimensional array with 512 values. Second, a Dense layer with 1024 neurons is used; the input array with 512 entries is transformed using the *elu* activation function. Third, a Dropout layer with rate 0.5 is used, and, finally, a Dense layer with ten neurons returns an array with 10 entries. Since the network output is expected to be a vector $\mathbf{v}_m = (v_{m,1}, \dots, v_{m,10})^T$ such that $v_{m,j}$ represents the probability of an input image of being part of the j th category for $j = 1, \dots, 10$, in the last layer, we use the *softmax* function defined as

$$SM(\mathbf{z}) = \frac{e^z}{\sum_{j=1}^t e^{z_j}}, \quad (32)$$

where $\mathbf{z} \in \mathbb{R}^t$. This function resembles all the outputs of the neurons within the very last layer and produces positive estimates that sum up to 1.

Every layer of the network, except the last, can be viewed as a step forward in generating information to be used for classification. The vector of dimension 1024 built at the penultimate layer is essentially a set of features which have been extracted from the original image. More insight into the outputs of intermediate

Fig. 7 An image of a frog from cifar-10 dataset



layers, after training out network, we fed it with the image of the frog in Fig. 7 and analyzed the output of the four convolutional layers. These outputs are displayed in Fig. 8; the channels are plotted side by side for a total of 16 channels per row. In the first plot, we display the 32 channels of the tensor built at the first convolutional layer; the shape of the frog is pretty recognizable in all channels. After the second and the third layer, the image of the frog is no longer recognizable. Even if, after the fourth convolutional layer, the 4×4 pixels of each channel have not apparent connection with the original image, they still contain enough information. The dimension of the input has been reduced, and the condensed information contained in the array is used to generate the 1024 entries which provide the features needed for the final classification. As we will see in the numerical results subsection, the information spread by the network allows, after network training, to correctly classify new entries with satisfactory accuracy.

Training the Neural Network

In the training phase, in order to measure the error made by the network on the prediction of each data, we used the loss function (9) where \mathbb{E} is *categorical cross-entropy function* defined as

$$\mathbb{E}(\mathbf{v}_m(x; \mathbf{d}_i), \hat{\mathbf{y}}_i) = - \sum_{j=1}^{10} \hat{y}_{ij} \log(v_{m,j}(x; \mathbf{d}_i)).$$

In the training phase, the weights of each layer of the network are updated via the minimization of the loss function; any of the methods previously described can be applied.

The training procedure consists in shuffling the training dataset and splitting it into mini-batches. The neural network is fed with each of such mini-batches in order to compute the approximated value of the gradient and to update the network

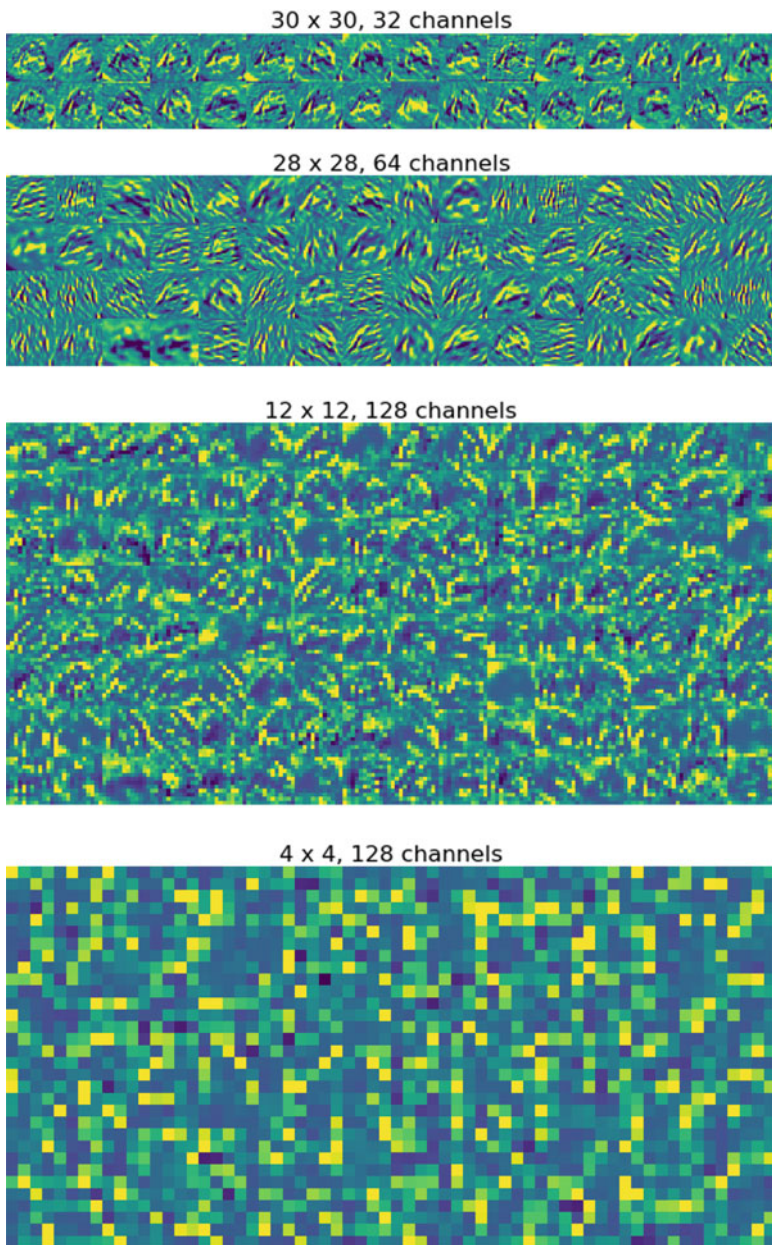


Fig. 8 *Intermediate activation*: output of intermediate convolutional layers. The network is fed with the image of a frog in Fig. 7. The color gradient we used for the intensity spans from yellow (lowest intensity) to blue (highest)

weights using any of the methods described in previous sections. Once the whole dataset has been used, the procedure is repeated. In machine learning terminology, the number of iterations needed to the neural network to handle each entry of the dataset is called an *epoch* of the training.

Implementation Details

We implemented the neural network and the training routine using the Python library Keras (<https://keras.io/>) and Tensorflow (<https://www.tensorflow.org/>) for handling the backend on the GPU, a NVIDIA Quadro M1000M. Keras comes with an utility to get the cifar-10 dataset split in training and test. We adapted one of the examples contained into Keras library (<https://www.tensorflow.org/tutorials/images/cnn>) to develop the network architecture previously described.

The SGD optimizer, presented in section “[Stochastic Gradient and Variance Reduction Methods](#)”, is included in Keras. After fine-tuning, we ran it using steplength $\alpha_k = 10^{-2}$, $\forall k \geq 0$. SVRG, presented in section “[Stochastic Gradient and Variance Reduction Methods](#)”, was run using an available implementation (<https://github.com/idiap/importance-sampling>); in such implementation, the SVRG gradient update rules are wrapped around the Keras framework. The full gradient on the outer iteration of SVRG was replaced by a SG computed on a mini-batch of 1000 training samples; the outer iteration was scheduled to be performed 32 times per epoch. The steplength for the inner iteration was set to 10^{-2} . TRish optimizer presented in section “[Gradient Methods with Adaptive Steplength Selection Based on Globalization Strategies](#)” has been implemented from scratch. After fine-tuning, the hyperparameters were set as follows: $\alpha_k = 10^{-1}$, $\forall k \geq 0$, $\gamma_{1,k} = 1$, $\forall k \geq 0$, and $\gamma_{2,k} = 10^{-3}$, $\forall k \geq 0$.

All the three methods have been implemented in a mini-batch manner as described at the end of the previous section. The batch size used for all training runs is 32, i.e., g_k was computed through (4) with $|\mathcal{S}_{k,g}| = 32$. The methods under comparison do not use the objective function at all; then its approximation is not needed.

Results

SGD, SVRG, and TRish were run imposing a number of 25 epochs. At the end of each epoch, the accuracy on both training and testing sets was measured. The accuracy is defined as the percentage of samples for which the classifier assigned the highest probability to the actual class. In Fig. 9, we report the accuracy achieved by each method both on the training and on the testing set during the training. The accuracy is evaluated at the end of each epoch.

TRish method appears to be the most effective in classification. Our experience showed that in the large majority of TRish iterations, the normalized step arising from the minimization of the trust-region subproblem (18) is selected. We recall that the key difference in the gradient methods under investigation is that TRish

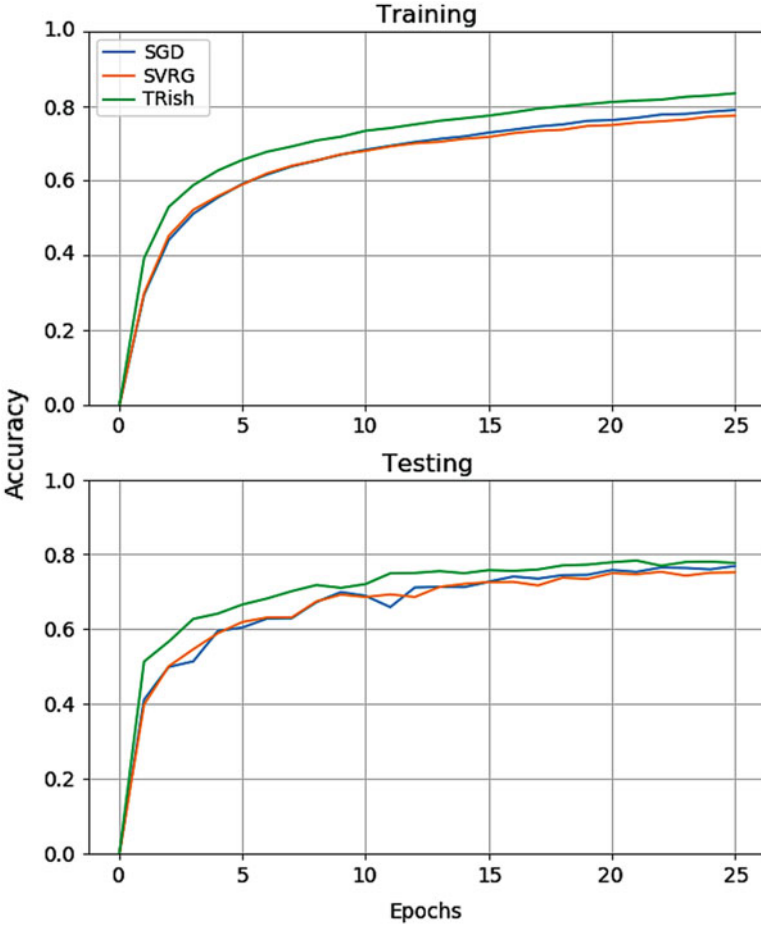


Fig. 9 The trend of training and test accuracy during the epochs

can take normalized steps and this can be viewed as an adaptive steplength selection as the step taken is $s_k = -\frac{\alpha_k}{\|g_k\|} g_k$ instead of $-\alpha_k g_k$. The adaptive approach used in TRish clearly improves classification on the testing set with respect to SGD and SVRG run with prefixed steplength. In fact, after only two epochs, TRish is already more accurate than SGD and SVRG and gives approximately 74% of accuracy on the test set after 12 epoch.

Conclusion

Optimization methods play a key role in machine learning applications. In this work, several subsampled first-order optimization methods suited for machine learning applications have been revised both from a theoretical and algorithmic point of

view. Stochastic procedures for solving convex and nonconvex problems applicable to neural networks and convolutional neural networks have been discussed, and numerical experience on a convolutional neural network designed for classifying images has been presented. Our presentation aims to show how the specific features of the optimization problems arising in the training phase of neural networks give rise to stochastic procedures which can address the numerical solution of convex and nonconvex problems.

The presented procedures are recent and part of the state of the art in optimization for machine learning. The literature on this topic is immense and steadily increasing, and this presentation is not comprehensive of the variety of existing first-order methods. We focused on methods with well-assessed convergence analysis. However we are aware of widely adopted methods which are less theoretically well founded than the procedures presented but are successful in machine learning. At this regard, we would like to mention SGD with momentum (Rumelhart et al. 1986; Loizou 2017) and ADAM (Kingma and Ba 2015; Sashank 2018). Both methods aim to speed the convergence rate of SGD method in the solution of ill-conditioned problems where the surface in a neighborhood of local optima curves more steeply in one direction than in another. In fact, in such cases a common drawback of steepest descent methods is that iterates zigzag toward the solution (Nocedal et al. 1999; Sutton 1986). To avoid that, SGD with momentum makes use of a search direction which is a combination of the current gradient approximation and the step (first-order momentum of the stochastic gradient) used at the previous iteration. ADAM method computes individual adaptive steplengths for updating the iterate component-wise on the basis of the current first- and second-order momentum of the stochastic gradient.

We conclude underling a current growing interest in second-order methods for nonconvex finite-sum optimization problems; see, e.g., Aggarwal (2018), Bellavia et al. (2020, 2021, 2019, 2020a,b), Berahas et al. (2020), Bollapragada et al. (2019), Bottou et al. (2018), Byrd et al. (2016), Byrd et al. (2012), Erdogdu et al. (2015), Liu et al. (2018), Roosta-Khorasani et al. (2019), Strang (2019), Xu et al. (2016, 2019).

Acknowledgments The financial support of INdAM-GNCS Projects 2019 and 2020 is gratefully acknowledged by the first and by the fourth authors. Thanks are due the referee whose comments improved the presentation of this paper.

References

- Aggarwal, C.C.: *Neural Networks and Deep Learning*. Springer (2018)
- Andradottir, S.: A scaled stochastic approximation algorithm. *Manag. Sci.* **42**, 475–498 (1996)
- Armijo, L.: Minimization of functions having lipschitz continuous first partial derivatives. *Pac. J. Math.* **16**, 1–3 (1966)
- Babanezhad, R., Ahmed, M.O., Virani, A., Schmidt, M., Konečný, J., Sallinen S.: Stop wasting my gradients: Practical SVRG. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 2 pp. 2251–2259 (2015)

- Barzilai, J., Borwein, J.: Two-point step size gradient. *IMA J. Numer. Anal.* **8**, 141–148 (1988)
- Bellavia, S., Gurioli, G.: Complexity analysis of a Stochastic cubic regularisation method under inexact gradient evaluations and dynamic Hessian accuracy, (2020). arXiv:2001.10827
- Bellavia, S., Gurioli, G., Morini, B., Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. *IMA J. Numer. Anal.* **41**, 764–799 (2021). <https://doi.org/10.1093/imanum/drz076>
- Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM J. Optimiz.* **29**, 2881–2915 (2019)
- Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: High-order Evaluation Complexity of a Stochastic Adaptive Regularization Algorithm for Nonconvex Optimization Using Inexact Function Evaluations and Randomly Perturbed Derivatives (2020a) arXiv:2005.04639
- Bellavia, S., Krejić, N., Krklec Jerinkić, N.: Subsampled Inexact Newton methods for minimizing large sums of convex function. *IMA J. Numer. Anal.* **40**, 2309–2341 (2020b)
- Bellavia, S., Krejić, N., Morini, B.: Inexact restoration with subsampled trust-region methods for finite-sum minimization. *Comput. Optim. Appl.* **76**, 701–736 (2020c)
- Berahas, A.S., Bollapragada, R., Nocedal, J.: An investigation of Newton-sketch and subsampled Newton methods. *Optim. Method Softw.* **35**, 661–680 (2020)
- Bertsekas, D.P., Tsitsiklis, J.N.: Gradient convergence in gradient methods with errors. *SIAM J. Optimiz.* **10**, 627–642 (2000)
- Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer (2006)
- Birgin, G.E., Krejić, N., Martínez, J.M.: On the employment of Inexact Restoration for the minimization of functions whose evaluation is subject to programming errors. *Math. Comput.* **87**, 1307–1326 (2018)
- Blanchet, J., Cartis, C., Menickelly, M., Scheinberg, K.: Convergence rate analysis of a stochastic trust region method via submartingales. *INFORMS J. Optim.* **1**, 92–119 (2019)
- Bollapragada, R., Byrd, R., Nocedal, J.: Exact and inexact subsampled Newton methods for optimization. *IMA J. Numer. Anal.* **39**, 545–578 (2019)
- Bottou, L., Curtis, F.C., Nocedal, J.: Optimization methods for large-scale machine learning. *SIAM Rev.* **60**, 223–311 (2018)
- Byrd, R.H., Hansen, S.L., Nocedal, J., Singer Y.: A stochastic quasi-Newton method for large-scale optimization. *SIAM J. Optimiz.* **26**, 1008–1021 (2016)
- Byrd, R.H., Chin, G.M., Nocedal, J., Wu, Y.: Sample size selection in optimization methods for machine learning. *Math. Program.* **134**, 127–155 (2012)
- Cartis, C., Scheinberg, K.: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Program.* **169**, 337–375 (2018)
- Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. *Math. Program.* **169**, 447–487 (2018)
- Chollet, F.: *Deep Learning with Python*. Manning Publications Co. (2017)
- Curtis, F.E., Scheinberg, K., Shi, R.: A stochastic trust region algorithm based on careful step normalization. *INFORMS J. Optimiz.* **1**, 200–220 (2019)
- Defazio, A., Bach, F., Lacoste-Julien, S.: SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems 27 (NIPS 2014)*
- Delyon, B., Juditsky, A.: Accelerated stochastic approximation. *SIAM J. Optimiz.* **3**, 868–881 (1993)
- Erdogdu, M.A., Montanari, A.: Convergence rates of sub-sampled Newton methods, *Advances in Neural Information Processing Systems 28 (NIPS 2015)*
- Forsyth, D.A., Ponce, J.: *Computer Vision: A Modern Approach*. Pearson (2002)
- Friedlander, M.P., Schmidt, M.: Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Sci. Comput.* **34**, 1380–1405 (2012)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. The MIT Press (2016)
- Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer New York Inc. (2001)

- Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Proceedings of the 26th International Conference on Neural Information Processing Systems 26 (NIPS 2013)
- Lei, L., Jordan, M.I.: Less than a single pass: Stochastically controlled stochastic gradient method. In: Proceedings of the Twentieth Conference on Artificial Intelligence and Statistics (AISTATS) (2017)
- Liu, L., Liu, X., Hsieh, C.-J., Tao, D.: Stochastic second-order methods for non-convex optimization with inexact Hessian and gradient (2018) arXiv:1809.09853
- Loizou, N., Richtárik, P.: Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods (2017) arXiv:1712.09677
- Kesten, H.: Accelerated stochastic approximation. *Ann. Math. Statist.* **29**, 41–59 (1958)
- Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* **23**, 462–466 (1952)
- Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization, 3rd International Conference on Learning Representations, ICLR 2015 (2015) arXiv: 1412.6980
- Krejić, N., Lužanin, Z., Stojkowska, I.: A gradient method for unconstrained optimization in noisy environment. *App. Numer. Math.* **70**, 1–21 (2013)
- Krejić, N., Lužanin, Z., Ovcin, Z., Stojkowska, I.: Descent direction method with line search for unconstrained optimization in noisy environment. *Optim. Method. Soft.* **30**, 1164–1184 (2015)
- Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research. Springer (1999)
- Krejić, N., Martínez, J.M.: Inexact restoration approach for minimization with inexact evaluation of the objective function. *Math. Comput.* **85**, 1775–1791 (2016)
- Krejić, N., Krklec, N.: Line search methods with variable sample size for unconstrained optimization. *J. Comput. Appl. Math.* **245**, 213–231 (2013)
- Krejić, N., Krklec Jerinkić N.: Nonmonotone line search methods with variable sample size. *Numer. Algorithms* **68**, 711–739 (2015)
- Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report, University of Toronto (2009)
- Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. *SIAM J. Optimiz.* **19**, 1574–1609 (2009)
- Nesterov, Y.: *Introductory lectures on convex programming, Volume I: Basic course*. Lecture Notes (1998)
- Nocedal, J., Sartenaer, A., Zhu, C.: On the behavior of the gradient norm in the steepest descent method. *Comput. Optim. Appl.* **22**, 5–35 (2002)
- Nguyen, L.M., Liu, J., Scheinberg, K., Takač, M., SARAH: A novel method for machine learning problems using stochastic recursive gradient. In: Proceedings of the 34th International Conference on Machine Learning (2017) pp. 2613–2621
- Paquette, C., Scheinberg, K.: A stochastic line search method with expected complexity analysis. *SIAM J. Optim.* **30**, 349–376 (2020)
- Patterson, J., Gibson, A.: *Deep Learning: A Practitioner’s Approach*. O’Reilly Media, Inc (2017)
- Pilanci, M., Wainwright, M.J.: Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM J. Optimiz.* **27**, 205–245 (2017)
- Polak, E., Royset, J.O.: Efficient sample sizes in stochastic nonlinear programming. *J. Comput. Appl. Math.* **217**, 301–310 (2008)
- Raj, A., Stich, S.U.: *k-SVRG: Variance Reduction for Large Scale Optimization* (2018) arXiv:1805.00982
- Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **7**, 26–33 (1997)
- Ross, S.: *Simulation*. Elsevier, 4th edn. (2006)
- Roosta-Khorasani, F., Mahoney, M.W.: Sub-sampled Newton methods. *Math. Progr.* **174**, 293–326 (2019)
- Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**, 400–407 (1951)

- Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986)
- Sashank, J.R., Satyen, K.A., Sanjiv, K.U.: On the convergence of Adam and beyond. In: 6th International Conference on Learning Representations (ICLR 2018)
- Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. *Math. Program.* **162**, 83–112 (2017)
- Shanmugamani, R.: *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*. Packt Publishing (2018)
- Spall J.C.: *Introduction to Stochastic Search and Optimization*. Wiley-Interscience series in discrete mathematics (2003)
- Spall J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerosp. Electron. Syst.* **34**, 817–823 (1998)
- Shanmugamani, R.: *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*. Packt Publishing (2018)
- Strang, G.: *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press (2019)
- Sutton, R.: Two problems with back propagation and other steepest descent learning procedures for networks. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pp. 823–832 (1986)
- Tan, C., Ma, S., Dai, Y., Qian, Y.: Barzilai-Borwein step size for stochastic gradient descent. *Advances in Neural Information Processing Systems* **29**, 685–693 (2016)
- Toint, P.L.: Nonlinear stepsize control, trust regions and regularizations for unconstrained optimization. *Optimiz. Meth. Softw.* **28**, 82–95 (2013)
- Tripuraneni, N., Stern, M., Regier, M., Jordan, M.I.: Stochastic cubic regularization for fast nonconvex optimization. *Adv. Neural Inf. Proces. Syst.* **31**, 2899–2908 (2018)
- Tropp, J.: An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.* **8**(1–2), 1–230 (2015)
- Yousefian, F., Nedic, A., Shanbhag, U.V.: On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica* **48**, 56–67 (2012)
- Wang, C., Chen, X., Smola, A., Xing, E.: Variance reduction for stochastic gradient optimization. *Adv. Neural Inf. Proces. Syst.* **26**, 181–189 (2013)
- Xu, Z., Dai, Y.H.: New stochastic approximation algorithms with adaptive step sizes. *Optim. Lett.* **6**, 1831–1846 (2012)
- Xu, P., Yang, J., Roosta-Khorasani, F., Ré, C., Mahoney, M.W.: Sub-sampled Newton methods with non-uniform sampling. *Adv. Neural Inf. Proces. Syst.* **29**, 3008–3016 (2016)
- Xu, P., Roosta-Khorasani, F., Mahoney, M.W.: Newton-type methods for non-convex optimization under inexact Hessian information. *Math. Program.* (2019). <https://doi.org/10.1007/s10107-019-01405-z>
- Wang, X., Yuan, Y.X.: Stochastic Trust Region Methods with Trust Region Radius Depending on Probabilistic Models (2019). arXiv:1904.03342



Bregman Methods for Large-Scale Optimization with Applications in Imaging

3

Martin Benning and Erlend Skaldehaug Riis

Contents

Introduction	98
Bregman Proximal Methods	99
A Unified Framework for Implicit and Explicit Gradient Methods	101
Bregman Proximal Gradient Method	102
Bregman Iteration	104
Linearized Bregman Iteration as Gradient Descent	104
Bregman Iterations as Iterative Regularization Methods	106
Inverse Scale Space Flows	107
Accelerated Bregman Methods	108
Incremental and Stochastic Bregman Proximal Methods	110
Stochastic Mirror Descent	111
The Sparse Kaczmarz Method	111
Deep Neural Networks	113
Bregman Incremental Aggregated Gradient	114
Bregman Coordinate Descent Methods	116
The Bregman Itoh–Abe Method	117
Equivalencies of Certain Bregman Coordinate Descent Methods	119
Saddle-Point Methods	120
Alternating Direction Method of Multipliers	121
Primal-Dual Hybrid Gradient Method	122
Applications	124
Robust Principal Component Analysis	125
Deep Learning	127
Student-t Regularized Image Denoising	129
Conclusions and Outlook	131
References	132

M. Benning (✉)

The School of Mathematical Sciences, Queen Mary University of London, London, UK
e-mail: m.benning@qmul.ac.uk

E. S. Riis

The Department of Applied Mathematics and Theoretical Physics, Cambridge, UK

© Springer Nature Switzerland AG 2023

K. Chen et al. (eds.), *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, https://doi.org/10.1007/978-3-030-98661-2_62

97

Abstract

In this chapter we review recent developments in the research of Bregman methods, with particular focus on their potential use for large-scale applications. We give an overview on several families of Bregman algorithms and discuss modifications such as accelerated Bregman methods, incremental and stochastic variants, and coordinate descent-type methods. We conclude this chapter with numerical examples in image and video decomposition, image denoising, and dimensionality reduction with auto-encoders.

Keywords

Optimization · Bregman proximal methods · Bregman iterations · Inverse problems · Nesterov acceleration · Mirror descent · Kaczmarz method · Coordinate descent · Itoh-Abe method · Alternating direction method of multipliers · Primal-dual hybrid gradient · Robust principal components analysis · Deep learning · Image denoising

Introduction

Bregman methods have a long history in mathematical research areas such as optimization, inverse and ill-posed problems, statistical learning theory, and machine learning. In this review, we mainly focus on the areas of optimization and inverse and ill-posed problems and the application of popular Bregman methods to potentially large-scale problems. Following Lev Bregman's seminal work in 1967 (Bregman 1967), it was not before the work of Censor and Lent (1981) in 1981 that the use of Bregman methods has slowly but steadily been popularized in the area of mathematical optimization, shortly followed by the advent of the mirror descent algorithm (Nemirovsky and Yudin 1983). Bregman proximal methods, which we discuss in greater detail in the following section, were first introduced by Censor and Zenios in their seminal work in 1992 (Censor and Zenios 1992), shortly followed by Teboulle (1992), Teboulle and Chen (1993), and Eckstein (1993). Bregman methods have been extensively studied since, see, for example, Bauschke et al. (2003) and references therein, and many notable extensions were developed, with one of the most popular ones in the context of inverse and ill-posed problems being the so-called Bregman iteration (Osher et al. 2005), which is based on a generalized Bregman distance notion (Kiwiel 1997b). Bregman iterations have been shown to possess favorable regularization properties over traditional linear iterative regularization methods, especially in the context of imaging and image processing applications, and therefore gained a lot of attention in those research fields. We refer to Osher et al. (2005), Burger (2016), and Benning and Burger (2018) for an overview on Bregman iterations.

The goal of this chapter is to provide a non-exhaustive overview over some recent developments in the adaptation of Bregman methods to handle potentially

large-scale problems. These extensions range from simple linearizations to accelerated versions of Bregman methods, incremental and stochastic adaptations, and coordinate descent variants to Bregman extensions of popular primal-dual frameworks. The chapter is therefore structured as follows. In section “[Bregman Proximal Methods](#)” we give an overview over Bregman proximal methods and some notable extensions. In section “[Accelerated Bregman Methods](#)” we discuss accelerations of the linearized Bregman iteration, before we focus on incremental and stochastic variants in section “[Incremental and Stochastic Bregman Proximal Methods](#).” Subsequently, we discuss coordinate descent-type Bregman methods in section “[Bregman Coordinate Descent Methods](#)” and saddle-point formulations of Bregman algorithms in section “[Saddle-Point Methods](#).” We present several application examples in section “[Applications](#)” before concluding this chapter with section “[Conclusions and Outlook](#).”

Bregman Proximal Methods

The Bregman proximal method or Bregman proximal algorithm is defined as the following iterative procedure. Starting with an initial value $x^0 \in \mathbb{R}^n$, we compute

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) + D_R(x, x^k) \right\}, \quad (1)$$

for $k \in \mathbb{N}$. Here $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function that we wish to minimize via (1). We assume that F is bounded from below and that both F and R satisfy conditions that guarantee existence and uniqueness of the solution of (1), without discussing them in greater detail. The term $D_R(x, y)$ denotes the Bregman distance w.r.t. a convex and continuously differentiable function $R : \mathbb{R}^n \rightarrow \mathbb{R}$, which is defined as

$$D_R(x, y) = R(x) - R(y) - \langle \nabla R(y), x - y \rangle, \quad (2)$$

for all $x, y \in \mathbb{R}^n$, see Bregman (1967) and Censor and Lent (1981). In the following example, we recall a few relevant examples of Bregman distances.

Example 1 (Bregman distances). For a symmetric, positive semi-definite matrix $Q \in \mathbb{R}^{n \times n}$ and the function $R(x) := \frac{1}{2} \langle Qx, x \rangle$, we observe

$$D_R(x, y) = \frac{1}{2} \langle Q(x - y), x - y \rangle.$$

Special cases include the squared Euclidean distance if Q is the identity matrix and the squared Mahalanobis distance (cf. Mahalanobis 1936) if Q is a covariance matrix.

The generalized Kullback-Leibler divergence, i.e.,

$$D_R(x, y) = \sum_{j=1}^n \left[x_j \log \left(\frac{x_j}{y_j} \right) + y_j - x_j \right],$$

can be obtained by choosing R as the (shifted, negative) Boltzmann-Shannon entropy, i.e., $R(x) := \sum_{j=1}^n [x_j \log(x_j) - x_j]$. Other notable examples include the Itakura–Saito distance (cf. Itakura 1968) and the Hellinger distance (cf. Hellinger 1909).

Note that $D_R(x, y) \geq 0$ is guaranteed for all $x, y \in \mathbb{R}^n$ due to the convexity of R . Before we are briefly going to discuss how this Bregman framework unifies implicit and explicit gradient methods in the following section, we want to recall some basic and well-known properties of (1).

Corollary 1. *Let $F : \mathbb{R}^n \rightarrow \mathbb{R}$ and $R : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable functions, where R is also convex, and suppose for some $\bar{x} \in \mathbb{R}^n$ that x^* is defined as*

$$x^* := \arg \min_{x \in \mathbb{R}^n} \{F(x) + D_R(x, \bar{x})\}. \quad (3)$$

Then, the following identity holds:

$$F(x^*) + D_F(x, x^*) + D_R(x, x^*) + D_R(x^*, \bar{x}) = F(x) + D_R(x, \bar{x}). \quad (4)$$

Corollary 1 can easily be verified by computing the optimality condition of (3), subsequent computation of the inner product of the optimality condition with $x^* - x$, and the use of the three-point identity for Bregman distances, first proven in Chen and Teboulle (1993, Lemma 3.1). Corollary 1 allows us to verify the following convergence result of the Bregman method with convergence rate $1/k$ for convex functions F .

Theorem 1. *Let $F : \mathbb{R}^n \rightarrow \mathbb{R}$ and $R : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable and convex functions. Suppose \hat{x} is a global minimizer of F that exists. Then, for any x^0 , the iterates (1) satisfy*

$$F(x^k) - F(\hat{x}) \leq \frac{D_R(\hat{x}, x^0) - D_R(\hat{x}, x^k)}{k},$$

for $k \in \mathbb{N}$.

Proof. Applying Corollary 1 for $x^* = x^{k+1}$, $\bar{x} = x^k$, and $x = \hat{x}$ yields

$$F(x^{k+1}) + D_F(\hat{x}, x^{k+1}) + D_R(\hat{x}, x^{k+1}) + D_R(x^{k+1}, x^k) = F(\hat{x}) + D_R(\hat{x}, x^k),$$

which implies

$$F(x^{k+1}) - F(\hat{x}) \leq D_R(\hat{x}, x^k) - D_R(\hat{x}, x^{k+1}),$$

due to the convexity of F and R . Summing up this inequality from $k = 0, \dots, K-1$ leads to

$$\sum_{k=0}^{K-1} F(x^{k+1}) - K F(\hat{x}) \leq D_R(\hat{x}, x^0) - D_R(\hat{x}, x^K).$$

Applying Corollary 1 again – but this time for $x^* = x^{k+1}$, $\bar{x} = x^k$ and $x = x^k$ – leaves us with

$$F(x^{k+1}) + D_F(x^k, x^{k+1}) + D_R(x^k, x^{k+1}) + D_R(x^{k+1}, x^k) = F(x^k) + \underbrace{D_R(x^k, x^k)}_{=0},$$

which in return implies $F(x^{k+1}) \leq F(x^k)$ due to the convexity of F and R (which is also an immediate consequence of the variational formulation of the Bregman method). Hence, we observe $K F(x^K) \leq \sum_{k=0}^{K-1} F(x^{k+1})$, which concludes the proof.

Remark 1. Note that the conditions on F and R in Theorem 1 alone do not necessarily guarantee uniqueness or even existence of x^{k+1} in (1). However, if the solution exists and is unique and computable, then Theorem 1 applies.

Let us now turn our attention to implicit and explicit gradient methods and how they can both be formulated as special cases of (1).

A Unified Framework for Implicit and Explicit Gradient Methods

While it is common in numerical analysis to distinguish between implicit and explicit methods, a feature of the Bregman framework is that it covers both types of methods. This can be seen by considering (1), i.e.,

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) + D_J(x, x^k) \right\}, \quad (5)$$

for the special choice of $J : \mathbb{R}^n \rightarrow \mathbb{R}$ with

$$J(x) := \begin{cases} R(x) & \text{implicit} \\ \frac{1}{\tau} R(x) - F(x) & \text{explicit} \end{cases}. \quad (6)$$

Evaluating the Bregman distance w.r.t. J turns (5) into

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \begin{array}{ll} F(x) + D_R(x, x^k) & \text{implicit} \\ F(x^k) + \langle \nabla F(x^k), x - x^k \rangle + \frac{1}{\tau} D_R(x, x^k) & \text{explicit} \end{array} \right\};$$

Hence, we can construct Bregman methods that are either implicit or explicit w.r.t. ∇F . Whenever we use J as the notation of our function throughout this manuscript, we implicitly refer to J as defined in (6). Whenever we use R , we refer to a function R that is not of the form $\frac{1}{\tau}R - F$. Note that we rediscover the traditional gradient descent algorithm for the choice $R(x) = \frac{1}{2}\|x\|^2$ as a special case of the explicit formulation. Furthermore, note that the explicit formulation

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x^k) + \langle \nabla F(x^k), x - x^k \rangle + \frac{1}{\tau} D_R(x, x^k) \right\} \quad (7)$$

is also known as mirror descent (Ben-Tal et al. 2001; Beck and Teboulle 2003; Juditsky et al. 2011), Bregman gradient method (Teboulle 2018), or recently also as NoLips (Bauschke et al. 2017). In order to guarantee convergence of (5), one usually has to guarantee convexity of J . In the explicit setting, this implies that τ and R have to be chosen to ensure convexity of $\frac{1}{\tau}R - F$ or equivalently that F is $1/\tau$ -smooth if R is also a quadratic function. The latter condition has basically been proposed in Bauschke et al. (2017) and further discussed in Benning et al. (2017a,b) and Bolte et al. (2018). It has also been shown that if the step size τ is chosen such that $cR - F$ is convex, for a some constant $c > 0$ and a function F , the estimate $0 < \tau \leq \left((1 + \gamma(R)) - \delta \right) / c$ is sufficient to guarantee convergence under mild assumptions that are outlined in detail in Bauschke et al. (2017). Here $\gamma(R)$ denotes the symmetry coefficient defined as

$$\gamma(R) := \inf \left\{ D_R(x, y) / D_R(y, x) \mid (x, y) \in (\text{int dom } R)^2 \setminus \{x, y \mid x=y\} \right\} \in [0, 1],$$

and δ is a constant that satisfies $\delta \in (0, 1 + \gamma(R))$. In the following section, we want to review the special case of Bregman gradient methods where F is the sum of two functions.

Bregman Proximal Gradient Method

An interesting, special case frequently considered in the literature is the case where F is a sum of two functions L and S , i.e., the Bregman method reads

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ L(x) + S(x) + D_J(x, x^k) \right\}, \quad (8)$$

where we assume that $L : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function. The function $S : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ on the other hand is proper, lower semi-continuous (l.s.c.) and convex, for $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$. If we choose $J(x) := \frac{1}{2\tau} \|x\|^2 - L(x)$ in the spirit of (6), then (8) reads

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \left\| x - \left(x^k - \tau \nabla L(x^k) \right) \right\|^2 + \tau S(x) \right\}, \\ &=: (I + \tau S)^{-1} \left(x^k - \tau \nabla L(x^k) \right), \end{aligned}$$

where $(I + \tau S)^{-1} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is known as the proximal map or resolvent, see, for instance, (Parikh et al. 2014). This is the classical proximal gradient method, also known as forward backward splitting (Lions and Mercier 1979). More general proximal gradient methods can be derived for different choices of J and S , for example, the entropic mirror descent algorithm (Nemirovsky and Yudin 1983; Beck and Teboulle 2003; Beck 2017; Doan et al. 2018), i.e.,

$$x_j^{k+1} = \frac{x_j^k \exp \left(-\tau (\nabla L(x^k))_j \right)}{\sum_{j=1}^n x_j^k \exp \left(-\tau (\nabla L(x^k))_j \right)},$$

for $j \in \{1, \dots, n\}$, the difference of the negative Boltzmann Shannon entropy as defined in Example 1 and the function L , i.e., $J(x) := \frac{1}{\tau} \sum_{j=1}^n [x_j \log(x_j) - x_j] - L(x)$ with the convention $0 \log(0) \equiv 0$, and the characteristic function

$$S(x) := \begin{cases} 0 & x \in \Sigma \\ +\infty & x \notin \Sigma \end{cases},$$

over the simplex constraint

$$\Sigma := \left\{ x \in \mathbb{R}^n \mid x_j \geq 0, \forall j \in \{1, \dots, n\}, \sum_{j=1}^n x_j = 1 \right\}.$$

We also mention *variable metric proximal gradient methods*, an important class of algorithms which may be viewed as an instance of Bregman proximal gradient methods where the Bregman function J_k is iteration-dependent. Denoting by $(A_k)_{k \in \mathbb{N}}$ a sequence of symmetric positive definite matrices, which act as preconditioners, we define $J_k(x) := \frac{1}{2\tau_k} \langle x, A_k x \rangle - L(x)$. Note that if $S \equiv 0$, $A_k = \nabla^2 L(x^k)$, and $\tau_k = 1$, then one recovers the Newton method for L

$$x^{k+1} = x^k - (\nabla^2 L(x^k))^{-1} \nabla L(x^k).$$

More generally when $S \neq 0$, one may choose A_k to be an approximation to the Hessian of L at x^k , so as to incorporate elements of quasi-Newton methods to the proximal gradient scheme. These schemes were studied by Bonnans et al. (1995) and later studied for non-convex objective functions (Chouzenoux et al. 2014; Frankel et al. 2015), Hilbert spaces (Combettes and Vũ 2014), and extensions to inertial methods (Bonettini et al. 2018), to mention a few examples.

In the next section, we focus on extensions of the Bregman proximal methods to convex but nonsmooth functions.

Bregman Iteration

A very important generalization of (1), first proposed in Osher et al. (2005), allows us to also use convex but nonsmooth functions J as defined in (6) instead of convex and continuously differentiable functions J . Suppose we are given a proper, l.s.c. and convex function $J : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$. Then its subdifferential, defined as

$$\partial J(y) := \left\{ p \in \mathbb{R}^n \mid J(x) - J(y) \geq \langle p, x - y \rangle, \forall x \in \mathbb{R}^n \right\},$$

is non-empty. It therefore makes sense to extend the definition (2) to a generalized Bregman distance (Kiwiel 1997a) for subdifferentiable functions, i.e.,

$$D_J^p(x, y) = J(x) - J(y) - \langle p, x - y \rangle,$$

for $p \in \partial J(y)$. A generalization of (1), commonly known as Bregman iteration, can then be defined as

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) + D_J^{p^k}(x, x^k) \right\}, \quad (9a)$$

$$p^{k+1} = p^k - \nabla F(x^{k+1}), \quad (9b)$$

for initial values $x^0 \in \mathbb{R}^n$ and $p^0 \in \partial J(x^0)$. Note that Corollary 1 and Theorem 1 also apply to Bregman iterations (cf. Benning and Burger 2018, Corollary 6.5), as those statements did not utilize any potential differentiability of J . Furthermore, note that the explicit variant of the Bregman iteration is known as the linearized Bregman iteration and has extensively been studied in Yin et al. (2008), Cai et al. (2009a,b,c), and Yin (2010).

Linearized Bregman Iteration as Gradient Descent

With the particular choice $J(x) = \frac{1}{2\tau} \|x\|^2 + \frac{1}{\tau} R(x) - F(x)$, the Bregman iteration (9) turns into the linearized Bregman iteration, which reads

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \left\{ F(x^k) + \langle \nabla F(x^k), x - x^k \rangle + \frac{1}{2\tau} \|x - x^k\|^2 + \frac{1}{\tau} D_R^{q^k}(x, x^k) \right\}, \\ &= (I + \partial R)^{-1} \left(x^k + q^k - \tau \nabla F(x^k) \right), \end{aligned} \quad (10a)$$

$$q^{k+1} = q^k - \left(x^{k+1} - x^k + \tau \nabla F(x^k) \right), \quad (10b)$$

where $(I + \partial R)^{-1}$ denotes the proximal mapping w.r.t. the function R and $q^k \in \partial R(x^k)$ the subgradient of R at x^k that is iteratively defined via (10b) and some initial value $q^0 \in \partial R(x^0)$. Suppose we assume that $(x^k + q^k)/\tau - \nabla F(x^k)$ is in the range of some matrix $A \in \mathbb{R}^{m \times n}$ and that we therefore can substitute $\tau A^\top b^k := x^k + q^k - \tau \nabla F(x^k)$. Then (10) can be written as

$$x^{k+1} = (I + \partial R)^{-1}(\tau A^\top b^k), \quad (11a)$$

$$A^\top b^{k+1} = A^\top b^k - \nabla F(x^{k+1}). \quad (11b)$$

In the following, we want to focus on the special case $F(x) = \frac{1}{2} \|Ax - b^\delta\|^2$ with $\nabla F(x) = A^\top (Ax - b^\delta)$ for a matrix $A \in \mathbb{R}^{m \times n}$, for which (11) simplifies to

$$x^{k+1} = (I + \partial R)^{-1}(\tau A^\top b^k), \quad (12a)$$

$$b^{k+1} = b^k - \left(Ax^{k+1} - b^\delta \right), \quad (12b)$$

with initial value $b^0 = b^\delta$, given the assumption that the initial values of the original formulation were $x^0 = 0$ and $p^0 = 0$. Note that we can also write (12) as

$$b^{k+1} = b^k - \left(A(I + \partial R)^{-1} \left(\tau A^\top b^k \right) - b^\delta \right). \quad (13)$$

Hence, if we can identify an energy G_τ for which we can associate its gradient ∇G_τ with $A(I + \partial R)^{-1} \left(\tau A^\top \cdot \right) - b^\delta$, we can consider the linearized Bregman iteration a gradient descent method applied to this specific energy. In Yin (2010) and Huang et al. (2013), this energy has been identified as

$$G_\tau(b) := \frac{\tau}{2} \|A^\top b\|^2 - \langle b, b^\delta \rangle - \frac{1}{\tau} \tilde{R}(\tau A^\top b),$$

where \tilde{R} denotes the Moreau-Yosida regularization of R (cf. Moreau 1965; Yosida 1964), i.e.,

$$\tilde{R}(z) := \inf_{x \in \mathbb{R}^n} \left\{ R(x) + \frac{1}{2} \|x - z\|^2 \right\}.$$

Since the gradient of the Moreau-Yosida regularization of R reads $\nabla \tilde{R}(z) = z - (I + \partial R)^{-1}(z)$ (see, for instance, Attouch et al. 2014, Proposition 17.2.1), we easily verify

$$\nabla G_\tau(b) = A(I + \partial R)^{-1}(\tau A^\top b) - b^\delta.$$

As a consequence, (13) is equivalent to

$$b^{k+1} = b^k - \nabla G_\tau(b^k),$$

and the linearized Bregman iteration for $F(x) = \frac{1}{2}\|Ax - b^\delta\|^2$ reduces to a gradient descent method. This equivalence will be useful when studying acceleration methods.

Bregman Iterations as Iterative Regularization Methods

Bregman iterations are not only useful for solving optimization problems but are also extremely important in the context of solving inverse and ill-posed problems. The reason for this is that Bregman iterations can be used as iterative regularization methods. If we consider the deterministic linear inverse problem

$$Ax^\dagger = b^\dagger, \tag{14}$$

for a given matrix $A \in \mathbb{R}^{m \times n}$, the aim of solving this inverse problem is to approximate x^\dagger in (14), for given A and data b^δ with $\|b^\dagger - b^\delta\| \leq \delta$. Here, δ is a known, positive bound on the error of the measured data b^δ and the data b^\dagger that satisfies (14).

Suppose we consider a convex function F that depends on A and b^δ , which we will denote as F_{b^δ} . It then can easily be shown that the iterates of (9) satisfy

$$D_J^{p^{k+1}}(x^\dagger, x^{k+1}) < D_J^{p^k}(x^\dagger, x^k),$$

for all indices $k \leq k^*(\delta)$ that satisfy Morozov's discrepancy principle (Morozov 1966), i.e.,

$$F_{b^\delta}(x^{k^*(\delta)}) \leq \eta\delta < F_{b^\delta}(x^k),$$

for a parameter $\eta \geq 1$, see Osher et al. (2005) and Burger et al. (2007). Note that for $\eta > 1$ it can be guaranteed that $k^*(\delta)$ is finite. With the additional regularity assumption that x^\dagger satisfies the so-called range condition (Benning and Burger 2018, Definition 5.8), i.e.,

$$x^\dagger \in \arg \min_{x \in \mathbb{R}^n} \{F_g(x) + R(x)\},$$

for some data $g \in \mathbb{R}^m$, one can prove the error estimate

$$D_J^p(x^\dagger, x^k) \leq \frac{\|w\|^2}{2k} + \delta\|w\| + \delta^2 k,$$

for the special case $F_{b^\delta}(x) := \frac{1}{2}\|Ax - b^\delta\|^2$, see Burger et al. (2007, Theorem 4.3). Here, w is defined as $w := g - Ax^\dagger \in \mathbb{R}^m$, which satisfies the source condition $A^*w \in \partial J(x^\dagger)$, cf. (Chavent and Kunisch 1997; Burger and Osher 2004). If $k^*(\delta)$ is of order $1/\delta$, we therefore observe

$$D_J^{p^{k^*(\delta)}}(x^\dagger, x^{k^*(\delta)}) = \mathcal{O}(\delta);$$

Hence, $x^{k^*(\delta)}$ converges to x^\dagger in terms of the Bregman distances if δ converges to zero.

For more details on how to use Bregman iterations in the context of (linear) inverse problems, we refer the reader to Osher et al. (2005), Resmerita and Scherzer (2006), Schuster et al. (2012), Burger (2016), and Benning and Burger (2018). For the remainder of this paper, we want to discuss modifications of Bregman iterations and Bregman proximal methods that are suitable to large-scale optimization and inverse problems.

Inverse Scale Space Flows

In what follows, we describe the *inverse scale space* (ISS) flow, a system of differential equations which can be derived as the continuous time limit of the Bregman iterations. For a Bregman function $J : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ and objective function $F : \mathbb{R}^n \rightarrow \mathbb{R}$, this flow is given by

$$\dot{p}(t) = -\nabla F(x(t)), \quad p(t) \in \partial J(x(t)). \quad (15)$$

It is straightforward to verify that Bregman iterations (9b) and linearized Bregman iterations (10) can be derived, respectively, as the forward and backward Euler discretization of (15).

The term *inverse scale space flow* was coined by Scherzer and Groetsch (2001) in 2001. In addition to its connection to Bregman schemes, the ISS flow itself is an active topic of research. Initially studied by Burger et al. (2006, 2007, 2013), and Burger (2016), it has found applications in nonlinear spectral analysis by Burger et al. (2016), Gilboa et al. (2016), and Schmidt et al. (2018).

The ISS flow itself has largely been studied in the context of scale space methods and data filtering, where the objective functions generally take the more specific forms $\|x - b^\dagger\|^2/2$ or $\|Ax - b^\dagger\|^2/2$. We mention some papers that address questions regarding the existence and uniqueness results for solutions to (15). Burger et al. (2007) proved existence, uniqueness, and certain regularity properties of the solution

to the flow when J is the total variation seminorm. These results were extended by Frick and Scherzer (2007) to all convex, proper, lower semicontinuous functions J , while in Burger et al. (2013), Burger et al. characterize the solution to the flow explicitly for the case $J = \|\cdot\|_1$. We note that while these studies do not assume strict convexity of J , strong convexity is ensured for F by the $\|\cdot\|^2$ term in F (restricted to the range of the linear operator A), so that the iterations (and flow) are still well-defined.

By supposing that J were twice continuously differentiable and μ -convex for some $\mu > 0$ (i.e., strongly convex with parameter μ , see Hiriart-Urruty and Lemaréchal 1993), we can provide an additional interpretation of the ISS flow, rewriting (15) as

$$\dot{x}(t) = -(\nabla^2 J(x(t)))^{-1} \nabla F(x(t)). \quad (16)$$

With this formulation, one can interpret the Hessian of $J(x(t))$ as a preconditioner for the flow. Furthermore, by using the chain rule, we derive an energy dissipation law for the system

$$\frac{d}{dt} F(x(t)) = \langle \dot{x}(t), \nabla F(x(t)) \rangle = -\langle \dot{x}(t), \nabla^2 J(x(t)) \dot{x}(t) \rangle \leq -\mu \|\dot{x}(t)\|^2,$$

where the final inequality follows from μ -convexity of J . Furthermore, observe that if $J = F$, (16) reduces to a continuous-time variant of Newton's method. One may tie this back to the variable metric proximal gradient methods, which were designed to incorporate quasi-Newton preconditioning to proximal gradient methods.

In section “The Bregman Itoh–Abe Method,” we describe the Bregman Itoh–Abe (BIA) method (Benning et al. 2020), an iterative system derived by applying structure-preserving methods from numerical integration to the flow. Thus the ISS flow provides an alternative way to consider variational formulations for formulating Bregman schemes.

Accelerated Bregman Methods

Not only when dealing with large-scale problems, reducing the number of iterations is an important goal to achieve when designing an algorithm. In Theorem 1 we have seen that the Bregman proximal method (1) has a convergence rate of order $1/k$. In the wake of Nesterov (1983), many acceleration strategies have been developed for first-order optimization methods that aim at minimizing convex functions. As we focus on Bregman methods, we want to highlight the following adaptation of Nesterov (1983), first developed in Huang et al. (2013) for quadratic functions F . There, the authors consider the linearized Bregman iteration, i.e., (9) for the choice $J(x) = \frac{1}{2\tau} \|x\|^2 + \frac{1}{\tau} R(x) - F(x)$, as shown in (10). We have seen that (10) can be formulated as the gradient descent (13) for the special case $F(x) = \frac{1}{2} \|Ax - b^\delta\|^2$. The authors in Huang et al. (2013) have applied the idea of Nesterov acceleration to formulation (13), which reads

$$b^{k+1} = (1 + \beta_k)b^k - \beta_k b^{k-1} - \nabla G_\tau((1 + \beta_k)b^k - \beta_k b^{k-1}), \quad (17)$$

where $\{\beta_k\}_{k \in \mathbb{N}}$ is a sequence of positive scalars. Applying τA^\top to both sides of the equation and substituting $\tau A^\top b^k = x^k + q^k - \tau A^\top (Ax^k + b^\delta)$ then yields the equivalent formulation

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) + (1 + \beta_k) D_J^{p^k}(x, x^k) - \beta_k D_J^{p^{k-1}}(x, x^{k-1}) \right\}, \quad (18a)$$

$$p^{k+1} = (1 + \beta_k)p^k - \beta_k p^{k-1} - \nabla F(x^{k+1}), \quad (18b)$$

for $J(x) = \frac{1}{2\tau} \|x\|^2 + \frac{1}{\tau} R(x) - F(x)$, $F(x) = \frac{1}{2} \|Ax - b^\delta\|^2$, $p^k = \frac{1}{\tau}(x^k + q^k) - \nabla F(x^k) \in \partial J(x^k)$, and $q^k \in \partial R(x^k)$ for all $k \in \mathbb{N}$.

Remark 2. We want to emphasize that the equivalence between (17) and (18) does not hold for arbitrary functions F as we have exploited the linearity of ∇F by making use of $\nabla F((1 + \beta_k)x^k - \beta_k x^{k-1}) = (1 + \beta_k)\nabla F(x^k) - \beta_k \nabla F(x^{k-1})$.

Note that (17) can also be written in less compact form as

$$x^{k+1} = (I + \partial R)^{-1}(z^k), \quad (19a)$$

$$y^{k+1} = z^k - \tau \nabla F(x^{k+1}), \quad (19b)$$

$$z^{k+1} = (1 + \beta_{k+1})y^{k+1} - \beta_{k+1}y^k, \quad (19c)$$

if we substitute $y^k = \tau A^\top b^k$. Following the same approach as in Chambolle and Dossal (2015), (19) can also be written as

$$x^{k+1} = (I + \partial R)^{-1}(z^k), \quad (20a)$$

$$y^{k+1} = z^k - \tau \nabla F(x^{k+1}), \quad (20b)$$

$$z^{k+1} = \left(1 - \frac{1}{t_{k+1}}\right)y^{k+1} + \frac{1}{t_{k+1}}u^{k+1}, \quad (20c)$$

$$u^{k+1} = y^k + t_{k+1}(y^{k+1} - y^k). \quad (20d)$$

for $\beta_k := (t_k - 1)/t_{k+1}$ and a sequence $\{t_k\}_{k \in \mathbb{N}}$ of positive parameters.

An open problem which has attracted interest in recent years concerns whether accelerated versions of Bregman (proximal) gradient methods with generic, strongly convex Bregman distances are possible (Teboulle 2018). In a recent work by Dragomir et al. (2019), this question is partly answered in the negative, concluding that for Bregman distances, based on smooth functions R or functions R that satisfy that $\frac{1}{\tau}R - F$ is convex, the $\mathcal{O}(1/k)$ convergence rate is optimal for first-order

methods that use previous gradient and Bregman proximal evaluations. However, for more restrictive function classes, faster convergence rates can be achieved, as has been shown in Hanzely et al. (2018) and Gutman and Peña (2018).

Acceleration strategies such as Nesterov acceleration have also been analyzed in the context of iterative regularization strategies (e.g., (9) combined with early stopping as described in section “Bregman Iterations as Iterative Regularization Methods”), see, for instance, Matet et al. (2017), Neubauer (2017), Garrigos et al. (2018), and Calatroni et al. (2019).

Incremental and Stochastic Bregman Proximal Methods

Many large-scale problems, in particular in machine learning, involve the minimization of functions of the form

$$F(x) := \frac{1}{m} \sum_{i=1}^m f_i(x). \quad (21)$$

In other words, the objective function is a sum of m individual functions. If m happens to be extremely large, computing the gradient of F can be computationally extremely expensive, rendering the application of traditional methods such as (1) or (18) computationally infeasible. Feasible alternatives are methods that make use of gradients that are only based on a subset $B \subset \{1, \dots, m\}$ of all indices. Such methods include incremental gradient methods (Bertsekas et al. 2011a) and stochastic gradient methods (Robbins and Monro 1951). If we assume that F in (21) is of the form

$$F(x) = L(x) + S(x) = \frac{1}{m} \sum_{i=1}^m \ell_i(x) + \frac{1}{m} \sum_{i=1}^m s_i(x), \quad (22)$$

an incremental version of the Bregman proximal gradient as in (8) can be formulated as

$$x^k = \arg \min_{x \in \mathbb{R}^n} \left\{ \ell_{i(k)}(x) + s_{i(k)}(x) + D_{J_k}(x, x^{k-1}) \right\}. \quad (23)$$

Here $i : \mathbb{N} \rightarrow \{1, \dots, m\}$ denotes the index function $i(x) := x \bmod m$, although other cycle orderings are certainly possible as well. A special case of (23) is the classical incremental proximal gradient method (Bertsekas et al. 2011b)

$$x^k = (I + \tau_k \partial s_{i(k)})^{-1} \left(x^{k-1} - \tau_k \nabla \ell_{i(k)}(x^{k-1}) \right)$$

for the choice of $J_k(x) = \frac{1}{2\tau_k} \|x\|^2 - \ell_{i(k)}(x)$. If we further pick $s_i \equiv 0$ for all i , we obtain the classical incremental gradient descent (Widrow and Hoff 1960; Bertsekas et al. 2011a), i.e.,

$$\begin{aligned}
x^k &= x^{k-1} - \tau_k \nabla \ell_{i(k)}(x^{k-1}), \\
&= x^{k-1} - \tau_k \nabla f_{i(k)}(x^{k-1}),
\end{aligned}
\tag{24}$$

as a special case.

In the following sections, we discuss extensions of stochastic gradient descent (SGD) and Kaczmarz methods in the Bregman framework, before highlighting the connection between single cycles of incremental Bregman proximal methods and deep neural network architectures.

Stochastic Mirror Descent

Stochastic gradient descent generalizes naturally to the Bregman proximal setting with the *stochastic mirror descent* (SMD) method (recall that mirror descent is equivalent to the Bregman gradient or linearized Bregman iteration). SMD is one of the most popular families of methods for stochastic optimization, and the method is defined as Nemirovski et al. (2009)

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \{\tau_k \langle \nabla f_{i(k)}(x^k), x \rangle + D_J^{p^k}(x, x^k)\}. \tag{25}$$

As in the setting of incremental descent methods, $i(k) \in \{1, \dots, n\}$ represents a sequence of indices, which in the setting of SMD are typically randomized.

SMD was originally introduced by Nemirovsky and Yudin (1983), while subsequent, significant contributions include Nemirovski et al. (2009), Nesterov (2009), and Xiao (2010). The framework and its convergence analysis were further extended by Duchi et al. (2012) to cases where the samples from the distribution are not assumed to be independent.

Similar to SGD, the SMD algorithms are suitable for large-scale optimization and online learning settings, yet furthermore they come with the added benefits of Bregman iterations of exploiting structures in the data. Because of this, SMD is one of the most widely used family of methods for large-scale stochastic optimization (Azizan and Hassibi 2018; Zhou et al. 2017).

In the aforementioned works on SMD, the Bregman function J is assumed to be differentiable. In contrast, the use of nonsmooth Bregman functions, e.g., that invoke the ℓ^1 -norm, is significant in the context of Bregman iterations and sparse signal processing. In the following section, we cover a Bregman method for sparse reconstruction of linear systems which can be seen as an instance of SMD, using the nonsmooth Bregman function $J(x) = \|x\|^2/2 + \lambda \|x\|_1$.

The Sparse Kaczmarz Method

The Kaczmarz method is a scheme for solving quadratic problems of the form $\min_x \langle x, Ax \rangle / 2 - \langle b, x \rangle$. The method was originally introduced by Kaczmarz

(1937) and later by Gordon et al. (1970) under the name *algebraic reconstruction technique*. In this section, we review the extension of *Kaczmarz methods* to *sparse Kaczmarz methods* (Lorenz et al. 2014b) and their block variants. The motivation for sparse Kaczmarz methods is to find sparse solutions to linear problems $Ax = b$ via the problem formulation

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|x\|^2 + \lambda \|x\|_1 : Ax = b \right\}. \quad (26)$$

We first briefly review the original Kaczmarz method. For $x^0 = 0$, time steps $\tau_k > 0$, and a sequence of indices $(i(k))_{k \in \mathbb{N}}$, the (randomized) Kaczmarz method is given by

$$x^{k+1} = x^k - \tau_k (\langle a_{i(k)}, x^k \rangle - b_{i(k)}) a_{i(k)}. \quad (27)$$

Here $a_{i(k)}$ denotes the i^{th} row vector of A . If $i(k)$ comprise a subset of indices, then the block-variant of the Kaczmarz method is given by

$$x^{k+1} = x^k - \tau_k a_{i(k)}^\dagger (a_{i(k)} x^k - b_{i(k)}),$$

where $a_{i(k)}$ denotes the submatrix formed by the row vectors of A indexed by $i(k)$ and $a_{i(k)}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $a_{i(k)}$. The iterates of the randomized Kaczmarz methods converge linearly to a solution of $Ax = b$ (Gower and Richtárik 2015).

Lorenz et al. (2014b) proposed a sparse Kaczmarz method as follows. Given starting points $x^0 = z^0 = 0$, the updates are given by

$$\begin{aligned} z^{k+1} &= z^k - \tau_k (\langle a_{i(k)}, x^k \rangle - b_{i(k)}) a_{i(k)}, \\ x^{k+1} &= S_\lambda(z^{k+1}). \end{aligned} \quad (28)$$

Here S_λ denotes the soft-thresholding operator with threshold λ . The iterates $(x^k)_{k \in \mathbb{N}}$ converge linearly to a solution of (26) (Schöpfer and Lorenz 2019, Theorem 3.2).

A block variant of the sparse Kaczmarz method was proposed in Lorenz et al. (2014b). For blocks of rows of A denoted by sets of indices $i(k)$, it consists of the updates

$$\begin{aligned} z^{k+1} &= z^k - \tau_k a_{i(k)}^\top (a_{i(k)} x^k - b_{i(k)}), \\ x^{k+1} &= S_\lambda(z^{k+1}). \end{aligned} \quad (29)$$

Note that this uses the transpose $a_{i(k)}^\top$, unlike the standard block Kaczmarz method which uses the pseudo-inverse $a_{i(k)}^\dagger$. This too converges to a solution of (26) (Lorenz et al. 2014a, Corollary 2.9).

The sparse (block-)Kaczmarz method (29) has connections to two aforementioned Bregman schemes. First, one may verify that it corresponds to the SMD method (25) for $J(x) = \|x\|^2/2 + \lambda\|x\|_1$ and $F(x) = \sum_{i=1}^n |\langle a_i, x \rangle - b_i|^2$. Second, if one takes the entire matrix A as each block, then one recovers the linearized Bregman method for the same J (Lorenz et al. 2014b).

As with the general SMD method, the sparse Kaczmarz method is particularly suitable in online reconstruction settings, where the rows of the linear system A and/or data entries b are not all available instantly but successively are made available over time. We refer the reader to Lorenz et al. (2014b) for numerical examples which include the application of online compressed sensing.

Deep Neural Networks

We can generalize the incremental Bregman proximal gradient (23) by including an additional, potentially nonlinear projection $H_k : \mathbb{R}^{n_{k-1}} \rightarrow \mathbb{R}^{n_k}$, to obtain

$$x^k = \arg \min_{x \in \mathbb{R}^{n_k}} \left\{ \ell_k(x) + s_k(x) + D_{J_k}(x, H_k(x^{k-1})) \right\}, \quad (30)$$

for a sequence of dimensions $\{n_k\}_{k=1}^l$ with $n_k \in \mathbb{N}$ for all $k = 1, \dots, l$. We are interested in a single cycle of this incremental Bregman proximal method only, which is why we have simplified the indexing notation from $i(k)$ to k throughout this subsection. In the following, we want to demonstrate how certain deep neural network architectures are special cases of (30). This connection was first investigated in the context of variational networks by Kobler et al. (2017), in the context of Bregman methods by Benning and Burger (2018), and in the context of proximal gradient methods by Frerix et al. (2017), Combettes and Pesquet (2018), and Bertocchi et al. (2019). Gradient-based learning with Bregman algorithms has also been studied in the context of image segmentation by Ochs et al. in (2015), and Bregman distances are used to analyze regularization strategies based on neural networks (Li et al. 2020). With the following example, we want to demonstrate how a class of feedforward neural networks coincides with (30).

Example 2 (Feedforward neural network with ReLU activation function). In this example we want to demonstrate how basic feedforward neural networks can be interpreted as variants of Algorithm (30). If we, for instance, choose $\{\ell_k\}_{k=1}^l$ to be of the form

$$\ell_k(x) := \frac{1}{2} \langle (I - M_k)x - 2b_k, x \rangle,$$

for quadratic matrices $\{M_k\}_{k=1}^l$ and vectors $\{b_k\}_{k=1}^l$ with $M_k \in \mathbb{R}^{n_k \times n_k}$ and $b_k \in \mathbb{R}^{n_k}$, which has the gradient

$$\nabla \ell_k(x) = \left(I - \frac{1}{2} (M_k + M_k^\top) \right) x - b_k,$$

and if we choose $\{s_k\}_{k=1}^l$ of the form

$$s_k(x) := \chi_{\geq 0}(x) = \begin{cases} 0 & \forall j : x_j \geq 0 \\ \infty & \exists j : x_j < 0 \end{cases}$$

for all $k \in \{1, \dots, l\}$, then we easily verify that for the choice $J_k(x) = \|x\|^2/2 - \ell_k(x)$ the update

$$x^k = \max \left(0, A_k(x^{k-1}) + b_k \right),$$

with $A_k := \frac{1}{2}(M_k + M_k^\top) \circ H_k$ is the unique solution of (30). Hence, we can consider this l -layer feedforward neural network with rectified linear units (ReLU) as activation functions (Nair and Hinton 2010) as a special case of the modified incremental Bregman gradient method (30) if we further guarantee that x^0 is chosen to be the input of the network.

Many other neural network architectures can be recovered in similar fashion to Example 2, where different activation functions can be recovered as proximal mappings for different choices of functions s_k , such as in Combettes and Pesquet (2018), and Bertocchi et al. (2019). For a recent overview of machine learning algorithms in the context of inverse problems, we refer to Arridge et al. (2019).

Bregman Incremental Aggregated Gradient

Two particularly interesting instances of incremental Bregman proximal methods are the *incremental aggregated gradient* (IAG) method (Blatt et al. 2007) and its stochastic counterpart *stochastic averaged gradient* (SAG) (Schmidt et al. 2017). For the sake of brevity, we focus on the incremental version in this paper. The IAG method reads

$$x^{k+1} = x^k - \frac{\tau_k}{m} g^k, \quad (31a)$$

$$g^{k+1} = g^k - \nabla f_{i(k+1)}(x^{k+1-m}) + \nabla f_{i(k+1)}(x^{k+1}). \quad (31b)$$

Here $\{\tau_k\}_{k \in \mathbb{N}}$ is a sequence of positive scalars and $i : \mathbb{N} \rightarrow \{1, \dots, m\}$ is defined as in section “[A Unified Framework for Implicit and Explicit Gradient Methods.](#)” Please also note that m arbitrary points $x^{1-m}, x^{2-m}, \dots, x^0$ have to be chosen as initialization. It is easy to see and has also been pointed out in Blatt et al. (2007) that (31) can be rewritten as

$$x^{k+1} = x^k - \frac{\tau_k}{m} \sum_{l=0}^{m-1} \nabla f_{i(k-l)}(x^{k-l}), \quad (32)$$

for $k \geq m$. Note that this is equivalent to the following characterization in terms of Bregman distances, in analogy to the explicit gradient descent characterization in section “[A Unified Framework for Implicit and Explicit Gradient Methods](#)”: if we rewrite (21) to $F(x) = \sum_{l=0}^{m-1} f_{i(k-l)}(x)$ for any $k \in \mathbb{N}$ and suppose we consider a Bregman method of the form

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) + \frac{1}{2\tau_k} \|x - x^k\|^2 - \frac{1}{m} \sum_{l=0}^{m-1} D_{f_{i(k-l)}}(x, x^{k-l}) \right\}, \quad (33a)$$

$$\begin{aligned} &= \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{m} \sum_{l=0}^{m-1} \left[f_{i(k-l)}(x^{k-l}) + \langle \nabla f_{i(k-l)}(x^{k-l}), x - x^{k-l} \rangle \right] \right. \\ &\quad \left. + \frac{1}{2\tau_k} \|x - x^k\|^2 \right\}, \end{aligned} \quad (33b)$$

then it becomes evident from computing the optimality condition of (33a) that the update (33b) is equivalent to (32) and hence (31) for $k \geq m$. Note that we can rewrite (33a) to

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ F(x) - \frac{1}{m} \sum_{l=1}^{m-1} D_{f_{i(k-l)}}(x, x^{k-l}) + D_{J_k}(x, x^k) \right\}, \quad (34)$$

for $J_k(x) := \frac{1}{2\tau_k} \|x\|^2 - \frac{1}{m} f_{i(k)}(x)$. The notable difference to the conventional IAG method is that we can replace the Bregman distance $D_{J_k}(x, x^k)$ in (34) with more generic Bregman distances. As in section “[A Unified Framework for Implicit and Explicit Gradient Methods](#),” we can for example choose $J_k(x) = \frac{1}{2\tau_k} \|x\|^2 + \frac{1}{\tau_k} R(x) - \frac{1}{m} f_{i(k)}(x)$ and therefore derive incremental Bregman iterations of the form

$$\begin{aligned} x^{k+1} &= (I + \partial R)^{-1} \left(x^k + q^k - \frac{\tau_k}{m} g^k \right) \\ q^{k+1} &= q^k - \left(x^{k+1} - x^k + \frac{\tau_k}{m} g^k \right), \\ g^{k+1} &= g^k - \nabla f_{i(k+1)}(x^{k+1-m}) + \nabla f_{i(k+1)}(x^{k+1}), \end{aligned}$$

where $q^k \in \partial R(x^k)$ for all k . Hence, substituting $y^k = x^k + q^k - \frac{\tau_k}{m} g^k$ yields the equivalent formulation

$$\begin{aligned}
x^{k+1} &= (I + \partial R)^{-1} \left(y^k \right), \\
g^{k+1} &= g^k - \nabla f_{i(k+1)}(x^{k+1-m}) + \nabla f_{i(k+1)}(x^{k+1}), \\
y^{k+1} &= y^k - \frac{\tau_{k+1}}{m} g^{k+1}.
\end{aligned}$$

If F is of the form (22), where $s_i = s$ for some (convex) function $s : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ for all indices $i \in \{1, \dots, m\}$ and if we choose $J_k(x) = \frac{1}{m\tau_k} R(x) - \frac{1}{m} \ell_{i(k)}(x)$ for continuously differentiable R , we recover the proximal-like incremental aggregated gradient (PLIAG) method, recently proposed in Zhang et al. (2017), which reads

$$\begin{aligned}
x^{k+1} = \arg \min_{x \in \mathbb{R}^n} & \left\{ s(x) + \sum_{l=0}^{m-1} \left[\ell_{i(k-l)}(x^{k-l}) + \langle \nabla \ell_{i(k-l)}(x^{k-l}), x - x^{k-l} \rangle \right] \right. \\
& \left. + \frac{1}{\tau_k} D_R(x, x^k) \right\}.
\end{aligned}$$

Needless to say, many different IAG or SAG methods can be derived for different choices of $\{J_k\}_{k=1}^m$. Choosing J_k such that convergence of the above algorithms is guaranteed is a delicate issue and involves carefully chosen assumptions, cf. Zhang et al. (2017, Section 2.3). Convergence guarantees for J_k as defined above with an arbitrary (proper, convex, and l.s.c.) function R which is an open problem. Having considered incremental variants of Bregman proximal algorithms, we now want to review coordinate descent adaptations of this algorithm in the following section.

Bregman Coordinate Descent Methods

In the previous section, we have reviewed Bregman adaptations of popular algorithms for minimizing objective functions that are sums of individual objective functions that occur in numerous large-scale applications, such as empirical risk minimization in machine learning.

In this section, we want to focus on Bregman adaptations of algorithms that aim to minimize multi-variable functions $F : \mathbb{R}^n \rightarrow \mathbb{R}$ by minimizing the objective with respect to one variable at a time. If we consider (1) for example, a simple coordinate descent adaption is

$$x_i^{k+1} = \arg \min_{x \in \mathbb{R}} \left\{ F(x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}, x, x_{i+1}^k, \dots, x_n^k) + D_{J_i}(x, x_i^k) \right\},$$

See, for example, Hua and Yamashita (2016), Corona et al. (2019a,b), Ahookhosh et al. (2019), Benning et al. (2020), and Gao et al. (2020). In the following, we want to give a brief overview on Bregman coordinate descent-type methods, with particular emphasis on an Itoh-Abe discrete gradient-based method, and also

highlight their connections to traditional coordinate descent algorithms (and their Bregman adaptations) such as successive over-relaxation (SOR).

The Bregman Itoh–Abe Method

The Bregman Itoh–Abe (BIA) method (Benning et al. 2020) is a particular form for coordinate descent, derived by applying the discrete gradient method to the ISS flow (15). Discrete gradients are methods from geometric numerical integration for solving differential equations while preserving geometric structures – for details on geometric numerical integration, see, e.g., Hairer et al. (2006) and McLachlan and Quispel (2001) – and have found several applications to optimization, e.g., Benning et al. (2020), Grimm et al. (2017), Ehrhardt et al. (2018), Riis et al. (2018), and Ringholm et al. (2018) due to their ability to preserve energy dissipation laws.

A discrete gradient is an approximation to a gradient that must satisfy two properties as follows.

Definition (Discrete gradient). Let F be a continuously differentiable function. A *discrete gradient* is a continuous map $\bar{\nabla}F : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that for all $x, y \in \mathbb{R}^n$,

$$\langle \bar{\nabla}F(x, y), y - x \rangle = F(y) - F(x) \quad (\text{Mean value}), \quad (35)$$

$$\lim_{y \rightarrow x} \bar{\nabla}F(x, y) = \nabla F(x) \quad (\text{Consistency}). \quad (36)$$

Given a choice of $\bar{\nabla}F$, starting points $x^0, p^0 \in \partial J(x^0)$, and time steps $(\tau_k)_{k \in \mathbb{N}}$, the Bregman discrete gradient scheme is defined as

$$p^{k+1} = p^k - \tau_k \bar{\nabla}F(x^k, x^{k+1}), \quad p^{k+1} \in \partial J(x^{k+1}). \quad (37)$$

As with the other Bregman schemes, this is a discretization of (15). Furthermore, the following dissipation property is an immediate consequence of the definition of discrete gradients.

Remark 3. When $J(x) = \|x\|^2/2$, then the ISS flow reduces to the Euclidean gradient flow, and we refer to the corresponding BIA method simply as the Itoh–Abe (IA) method.

Proposition. Suppose J is μ -convex and that (x^{k+1}, p^{k+1}) solves the update (35) given (x^k, p^k) and time step $\tau_k > 0$. Then

$$F(x^{k+1}) - F(x^k) = -\frac{1}{\tau_k} D_J^{\text{symm}}(x^k, x^{k+1}) \leq -\frac{\mu}{\tau_k} \|x^k - x^{k+1}\|^2, \quad (38)$$

where $D_J^{\text{symm}}(x, y)$ is the symmetrized Bregman distance defined as

$D_J^{symm}(x, y) := D_J^p(x, y) + D_J^q(y, x) = \langle p - q, y - x \rangle$ for $p \in \partial J(y)$, $q \in \partial J(x)$.

Proof. By (35) and (37) respectively, we have

$$F(x^{k+1}) - F(x^k) = \langle \bar{\nabla} F(x^k, x^{k+1}), x^{k+1} - x^k \rangle = -\frac{1}{\tau_k} \langle p^{k+1} - p^k, x^{k+1} - x^k \rangle.$$

The result then follows from monotonicity of convex functions, see, e.g., Hiriart-Urruty and Lemaréchal (1993, Theorem 6.1.2).

While there are various discrete gradients (see, e.g., McLachlan et al. 1999), the *Itoh–Abe discrete gradient* (Itoh and Abe 1988) (also known as the coordinate increment discrete gradient) is of particular interest in optimization as it is derivative-free and can be implemented for nonsmooth functions. It is defined as

$$\bar{\nabla} F(x, y) = \begin{pmatrix} \frac{F(y_1, x_2, \dots, x_n) - F(x)}{y_1 - x_1} \\ \frac{F(y_1, y_2, x_3, \dots, x_n) - F(y_1, x_2, \dots, x_n)}{y_2 - x_2} \\ \vdots \\ \frac{F(y) - F(y_1, \dots, y_{n-1}, x_n)}{y_n - x_n} \end{pmatrix}, \tag{39}$$

where $0/0$ is interpreted as $\partial_i F(x)$.

The BIA method is derived by plugging in the Itoh–Abe discrete gradient for $\bar{\nabla} F$ in (37). Provided that J is separable in the coordinates, i.e., $J(x) = \sum_{i=1}^n J_i(x_i)$, for $J_i : \mathbb{R} \rightarrow \overline{\mathbb{R}}$, then this method reduces to sequential updates along the coordinates. Specifically, it can be written as

$$p_i^{k+1} = p_i^k - \tau_{k,i} \frac{F(y^{k,i}) - F(y^{k,i-1})}{x_i^{k+1} - x_i^k}, \quad p_i^{k+1} \in \partial J_i(y_i^{k,i}), \tag{40}$$

$$y^{k,i} = [x_1^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k, \dots, x_n^k], \quad i = 1, \dots, n.$$

In addition to having a derivative-free formulation, the BIA method has convergence guarantees for a large group of objective functions. In particular, if the Bregman function J is nonsmooth and strongly convex, and if F is locally Lipschitz continuous with a regularity assumption (see Benning et al. 2020 for details), the BIA scheme converges to a set of *Clarke stationary points* (Benning et al. 2020, Theorem 4.5). Clarke stationarity refers to the optimality criteria $0 \in \partial^C F(x)$, where $\partial^C F(x)$ denotes the *Clarke subdifferential* of F at x (Clarke 1990).

This scheme comes with the cost that the updates (40) are in general implicit. However, for the cases

$$\begin{aligned}
 J(x) &= \frac{1}{2}\|x\|^2, & J(x) &= \frac{1}{2}\|x\|^2 + \lambda\|x\|_1, \\
 F(x) &= \frac{1}{2}\|Ax - b^\delta\|^2, & F(x) &= \frac{1}{2}\|Ax - b^\delta\|^2 + \gamma\|x\|_1,
 \end{aligned}$$

the updates are explicit (Benning et al. 2020).

In section “[Student-t Regularized Image Denoising](#),” we present an example of a nonsmooth, nonconvex image denoising model, previously considered in Benning et al. (2020), for which one can significantly speed up convergence by exploiting sparsity in the residual $x^* - x^\delta$.

Equivalencies of Certain Bregman Coordinate Descent Methods

In what follows, we briefly discuss and draw connections between various approaches to coordinate descent methods using Bregman distances. This builds on the observation by Miyatake et al. (2018) that the Itoh–Abe method applied to quadratic functions $F(x) = \langle x, Ax \rangle / 2 - \langle b, x \rangle$ is equivalent to the Gauss–Seidel and successive-over-relaxation (SOR) methods (Young 1971).

The explicit coordinate descent method (Beck and Tetrushvili 2013; Wright 2015) for minimizing F is given by

$$\begin{aligned}
 y^{k,0} &= x^k \\
 y^{k,i} &= y^{k,i-1} - \bar{\tau}_i [\nabla F(y^{k,i-1})]_i e^i, \\
 x^{k+1} &= y^{k,n},
 \end{aligned} \tag{41}$$

where $\bar{\tau}_i > 0$ is the time step and e^i denotes the i^{th} basis vector. As mentioned in Wright (2015), the SOR method is also equivalent to the coordinate descent method with F as above and the time steps scaled coordinate-wise by $1/A_{i,i}$. Hence, in this setting, the Itoh–Abe discrete gradient method is equivalent not only to SOR methods but to explicit coordinate descent.

Furthermore, these equivalencies extend to discretizations of the inverse scale space flow for certain quadratic objective functions and certain forms of Bregman functions J . Consider a quadratic function $F(x) = \langle x, Ax \rangle / 2 - \langle b, x \rangle$ where A is symmetric and positive definite, and denote by B the diagonal matrix for which $A_{i,i} = B_{i,i}$ for each i . Given a scaling parameter $\omega > 0$ and the Bregman function

$$J(x) = \frac{1}{2\omega} \langle x, Bx \rangle + \lambda\|x\|_1, \tag{42}$$

The Itoh–Abe method yields a sparse SOR scheme as detailed in Benning et al. (2020). We may compare this to a *Bregman linearized coordinate descent* scheme

$$\begin{aligned}
y^{k,0} &= x^k, \quad p^k \in \partial J(x^k), \\
z_i &= \arg \min_y [\nabla F(y^{k,i-1})]_i \cdot y + D_J^{p^k}(y^{k,i-1}, y^{k,i-1} + ye^i), \\
y^{k,i} &= y^{k,i-1} + z_i e^i, \\
x^{k+1} &= y^{k,n},
\end{aligned}$$

where J is given by (42) for some $\omega = \omega_E \in (0, 2)$. One can verify that these schemes are equivalent if one sets $\omega_E = \frac{1}{1/\omega + 1/2}$. We furthermore mention that these equivalencies also hold if we were to consider (implicit) Bregman iterations rather than linearized ones.

Remark 4. It is worth noting at this stage that while the Kaczmarz method (27) is closely related to SOR (Oswald and Zhou 2015), this connection does not carry over to the BIA method versus the sparse Kaczmarz method.

Saddle-Point Methods

Many problems in imaging (Chambolle and Pock 2016a) and machine learning (Goldstein et al. 2015; Adler and Öktem 2018) can be formulated as minimization problems of the form

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} G(x) + F(z) \quad \text{subject to} \quad K(x, z) = c. \quad (43)$$

Here $G : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ and $F : \mathbb{R}^m \rightarrow \overline{\mathbb{R}}$ are proper and lower semi-continuous and usually also convex functions, the operator $K : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^s$ is a bounded, and usually linear operator and $c \in \mathbb{R}^s$ are a vector. A classical linear example for K is

$$K(x, z) = Ax + Bz,$$

where $A \in \mathbb{R}^{s \times n}$ and $B \in \mathbb{R}^{s \times m}$ are matrices (Boyd et al. 2011).

In terms of optimization, the equality constraint can be incorporated with the help of a Lagrange multiplier $y \in \mathbb{R}^s$. We can then re-formulate (43) as finding a saddle point of an augmented Lagrange function, i.e., we solve

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \max_{y \in \mathbb{R}^s} \mathcal{L}_\delta(x, z; y)$$

for the augmented Lagrangian

$$\mathcal{L}_\delta(x, z; y) := G(x) + F(z) + \langle y, K(x, z) - c \rangle + \frac{1}{2\delta} \|K(x, z) - c\|^2, \quad (44)$$

where $\delta > 0$ is a positive scalar. For the special case $K(x, z) = Ax - z$ and $c \equiv 0$, one can replace $F(Ax)$ with its convex conjugate and formulate the alternative saddle-point problem

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} G(x) + \langle Ax, y \rangle - F^*(y), \quad (45)$$

where the convex conjugate or Fenchel conjugate F^* of F is defined as

$$F^*(y) := \sup_{x \in \mathbb{R}^n} \langle x, y \rangle - F(x).$$

We want to emphasize that extensions for nonconvex functions (Li and Pong 2015; Moeller et al. 2015; Möllenhoff et al. 2015) and extensions for nonlinear operators A (Valkonen 2014; Benning et al. 2015; Clason and Valkonen 2017) or nonlinear replacements of the dual product (Clason et al. 2019) exist. In the following, we review Bregman algorithms for the numerical computation of solutions of those saddle-point formulations.

Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM), (Gabay 1983), is a coordinate descent method applied to the augmented Lagrangian functional (44). The augmented Lagrangian is furthermore modified to also include appropriate penalization terms, so that we compute

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \mathcal{L}_\delta(x, z^k; \mu^k) + D_{J_x}(x, x^k), \quad (46a)$$

$$z^{k+1} = \arg \min_{z \in \mathbb{R}^m} \mathcal{L}_\delta(x^{k+1}, z; y^k) + D_{J_z}(z, z^k), \quad (46b)$$

$$y^{k+1} = \arg \max_{y \in \mathbb{R}^m} \mathcal{L}_\delta(x^{k+1}, z^{k+1}; y) - D_{J_y}(y, y^k), \quad (46c)$$

in an alternating fashion. To our knowledge, the first adaptation of ADMM to more general Bregman functions was proposed in Wang and Banerjee (2014). In the setting discussed here, the functions J_x , J_z , and J_y are convex and continuously differentiable functions. In the most basic scenario, we choose $K(x, z) = Ax + Bz$, J_x , and J_y as the zero functions, i.e., $J_x(x) = 0$ and $J_z(z) = 0$ for all $x \in \mathbb{R}^n$ $z \in \mathbb{R}^m$, while J_y is chosen to be a positive multiple of the squared Euclidean norm $J_y(y) := \frac{1}{2\tau} \|y\|^2$. Then (46) reduces to the classical ADMM setting (cf. Boyd et al. 2011)

$$\begin{aligned} x^{k+1} &= \left(A^\top A + \delta \partial G \right)^{-1} \left(A^\top \left(c - (Bz^k + \delta y^k) \right) \right), \\ z^{k+1} &= \left(B^\top B + \delta \partial F \right)^{-1} \left(B^\top \left(c - (Ax^{k+1} + \delta y^k) \right) \right), \\ y^{k+1} &= y^k + \tau \left(Ax^{k+1} + Bz^{k+1} - c \right). \end{aligned}$$

Depending on the choices of J_x , J_z , and J_y , many other useful variants are possible, such as

$$\begin{aligned}
x^{k+1} &= (I + \tau_x \delta \partial G)^{-1} \left(x^k - \tau_x A^\top (Ax^k + Bz^k + \delta y^k - c) \right), \\
z^{k+1} &= (I + \tau_z \delta \partial F)^{-1} \left(z^k - \tau_z B^\top (Ax^{k+1} + Bz^k + \delta y^k - c) \right), \\
y^{k+1} &= y^k + \tau_y (Ax^{k+1} + Bz^{k+1} - c),
\end{aligned}$$

for the choices $J_x(x) = \frac{1}{2\delta\tau_x}\|x\|^2 - \frac{1}{2\delta}\|Ax\|^2$, $J_z(z) = \frac{1}{2\delta\tau_z}\|z\|^2 - \frac{1}{2\delta}\|Bz\|^2$, and $J_y(y) = \frac{1}{2\tau_y}\|y\|^2$, which is fully explicit with respect to the operators A and B . Moreover, J_x is convex for $0 < \tau_x < \|A\|^2$, while J_z is convex for $0 < \tau_z < \|B\|^2$. A unified Bregman framework for primal-dual algorithms is discussed in greater detail in Zhang et al. (2011).

Primal-Dual Hybrid Gradient Method

In this section we focus on the special saddle-point formulation (45). It is straightforward to verify that for convex G and F a saddle point $(\hat{x}, \hat{y})^\top$ is characterized by the optimality system

$$0 \in \partial G(\hat{x}) + A^\top \hat{y}, \quad (47a)$$

$$0 \in \partial F^*(\hat{y}) - A\hat{x}. \quad (47b)$$

It is sensible and has indeed been suggested in Chambolle and Pock (2016b), and Hohage and Homann (2014) to solve this nonlinear inclusion problem with a fixed point algorithm of the form

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G(x^{k+1}) + A^\top y^{k+1} \\ \partial F^*(y^{k+1}) - Ax^{k+1} \end{pmatrix} + \partial J(x^{k+1}, y^{k+1}) - \partial J(x^k, y^k). \quad (48)$$

Here ∂J denotes the subdifferential of some convex function $J : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$. For the choice

$$\begin{aligned}
J(x, y) &:= \frac{1}{2} \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_M^2 && \text{with} && \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_M &:= \sqrt{\left\langle M \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} \right\rangle} \\
&&& && \text{and} & M &:= \begin{pmatrix} \frac{1}{\tau} I & -A^\top \\ -A & \frac{1}{\sigma} I \end{pmatrix},
\end{aligned}$$

and $\tau\sigma\|A\|^2 < 1$, we obtain the conventional primal-dual hybrid gradient (PDHG) method (with relaxation parameter set to one) as proposed and discussed in Zhu and Chan (2008), Pock et al. (2009), Esser et al. (2010), and Chambolle and Pock (2011, 2016a), which reads

$$x^{k+1} = (I + \tau\partial G)^{-1} \left(x^k - \tau A^\top y^k \right), \quad (49a)$$

$$y^{k+1} = (I + \sigma\partial F^*)^{-1} \left(y^k + \sigma A(2x^{k+1} - x^k) \right). \quad (49b)$$

Note that we can reformulate (48) to

$$\begin{aligned} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in & \begin{pmatrix} \partial G(x^{k+1}) - \partial G(\hat{x}) + A^\top(y^{k+1} - \hat{y}) \\ \partial F^*(y^{k+1}) - \partial F^*(\hat{y}) - A(x^{k+1} - \hat{x}) \end{pmatrix} \\ & + \partial J(x^{k+1}, y^{k+1}) - \partial J(\hat{x}, \hat{y}) - \left(\partial J(x^k, y^k) - \partial J(\hat{x}, \hat{y}) \right), \end{aligned} \quad (50)$$

if we add the optimality system (47) to (48), for a saddle point $(\hat{x}, \hat{y})^\top$. Taking a dual product of

$$\begin{pmatrix} \partial G(x^{k+1}) - \partial G(\hat{x}) + A^\top(y^{k+1} - \hat{y}) \\ \partial F^*(y^{k+1}) - \partial F^*(\hat{y}) - A(x^{k+1} - \hat{x}) \end{pmatrix}$$

with $(x^{k+1} - \hat{x}, y^{k+1} - \hat{y})^\top$ therefore yields

$$\begin{aligned} & \left\langle \begin{pmatrix} \partial G(x^{k+1}) - \partial G(\hat{x}) + A^\top(y^{k+1} - \hat{y}) \\ \partial F^*(y^{k+1}) - \partial F^*(\hat{y}) - A(x^{k+1} - \hat{x}) \end{pmatrix}, \begin{pmatrix} x^{k+1} - \hat{x} \\ y^{k+1} - \hat{y} \end{pmatrix} \right\rangle \\ & = D_G^{\text{symm}}(x^{k+1}, \hat{x}) + D_{F^*}^{\text{symm}}(y^{k+1}, \hat{y}) \geq 0. \end{aligned}$$

Here $D_J^{\text{symm}}(x, y)$ denotes the symmetric Bregman distance $D_J^{\text{symm}}(x, y) = D_J^q(x, y) + D_J^p(y, x) = \langle p - q, x - y \rangle$, for subgradients $p \in \partial J(x)$ and $q \in \partial J(y)$, which is also known as Jeffreys–Bregman divergence and closely related to other symmetrizations such as Jensen–Bregman divergences (Nielsen and Boltz 2011) and Burbea Rao distances (Burbea and Rao 1982a,b). As an immediate consequence, we observe

$$\begin{aligned} 0 & \geq \left\langle \partial J(x^{k+1}, y^{k+1}) - \partial J(x^k, y^k), \begin{pmatrix} x^{k+1} - \hat{x} \\ y^{k+1} - \hat{y} \end{pmatrix} \right\rangle \\ & = D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} \right) - D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right) + D_J \left(\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right), \end{aligned}$$

where we have made use of the three-point identity for Bregman distances (Chen and Teboulle 1993). Thus, we can conclude

$$D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} \right) + D_J \left(\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right) \leq D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right)$$

for all iterates. Consequently, the iterates are bounded in the Bregman distance setting with respect to J . Summing up the dual product of (48) with $(x^{k+1} - \hat{x}, y^{k+1} - \hat{y})^\top$ therefore yields

$$\begin{aligned} & \sum_{k=0}^N \left[D_G^{\text{symm}}(x^{k+1}, \hat{x}) + D_{F^*}^{\text{symm}}(y^{k+1}, \hat{y}) \right] + \sum_{k=0}^N D_J \left(\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right) \\ &= \sum_{k=0}^N \left[D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right) - D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} \right) \right] \leq D_J \left(\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}, \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} \right) \\ &< +\infty. \end{aligned}$$

Hence, we can conclude $D_G^{\text{symm}}(x^N, \hat{x}) \rightarrow 0$, $D_{F^*}^{\text{symm}}(y^N, \hat{y}) \rightarrow 0$, and $D_J \left(\begin{pmatrix} x^N \\ y^N \end{pmatrix}, \begin{pmatrix} x^k \\ y^k \end{pmatrix} \right)^\top \rightarrow 0$ for $N \rightarrow \infty$. If G and F^* are at least convex and if J is strongly convex with respect to some norm, one can further guarantee convergence of the corresponding iterates in norm to a saddle-point (x, y) solution of (45) with standard arguments. For more details, analysis, and extensions of PDHG methods, we refer the reader to Chambolle and Pock (2016a).

Applications

In the following we want to show applications for some of the Bregman algorithms discussed in this review chapter. We want to emphasize that none of the applications shown are really large-scale applications. The idea of this section is rather to demonstrate that the algorithms are applicable to a wide range of different problems, offering the potential to enhance actual large-scale problems. We focus on three combinations of applications and algorithms: robust principal component analysis via the accelerated linearized Bregman iteration, deep learning with an incremental proximal Bregman architecture, and image denoising via the Bregman Itoh–Abe method.

Robust Principal Component Analysis

Robust principal component analysis is an extension of principal component analysis first proposed in Candès et al. (2011). The key idea is to decompose a matrix $X \in \mathbb{R}^{m \times n}$ into a low-rank matrix $L \in \mathbb{R}^{m \times n}$ and a sparse matrix $S \in \mathbb{R}^{m \times n}$ by solving the optimization problem

$$\min_{L, S} \alpha_1 \|L\|_* + \alpha_2 \|S\|_1 \quad \text{subject to} \quad X = L + S. \quad (51)$$

Here $\|S\|_1$ is the one norm of the matrix S , i.e., $\|S\|_1 = \sum_{i=1}^m \sum_{j=1}^n |s_{ij}|$, while $\|L\|_*$ denotes the nuclear norm of L , which is the one norm of the singular values of L , i.e., $\|L\|_* = \sum_{j=1}^{\min(n,m)} \sigma_j$, for $L = U \Sigma V^*$ with $\Sigma_{ij} = \begin{cases} \sigma_j & i = j \\ 0 & i \neq j \end{cases}$ and U and V being orthogonal. There are numerous strategies for solving (51) numerically (Bouwman et al. 2018); we focus on using the accelerated linearized Bregman iteration as discussed in section “Accelerated Bregman Methods.” For this we use formulation (12) of the linearized Bregman iteration, respectively (19), in the accelerated case. We choose $A = (I \ I)^\top$, $b^\delta = X$, and $R = \alpha_1 \|\cdot\|_* + \alpha_2 \|\cdot\|_1$ and therefore obtain

$$\begin{aligned} L^{k+1} &= (I + \alpha_1 \partial \|\cdot\|_*)^{-1} (\tau X^k), \\ S^{k+1} &= (I + \alpha_2 \partial \|\cdot\|_1)^{-1} (\tau X^k), \\ X^{k+1} &= X^k - (L^{k+1} + S^{k+1} - X), \end{aligned}$$

in the case of (12), respectively

$$\begin{aligned} L^{k+1} &= (I + \alpha_1 \partial \|\cdot\|_*)^{-1} (\tau Y^k), \\ S^{k+1} &= (I + \alpha_2 \partial \|\cdot\|_1)^{-1} (\tau Y^k), \\ X^{k+1} &= Y^k - (L^{k+1} + S^{k+1} - X), \\ Y^{k+1} &= (1 + \beta_{k+1})X^{k+1} - \beta_{k+1}X^k, \end{aligned}$$

in the case of (17), for $X^0 := X$. We choose the parameters to be $\tau = 1/\|A\|^2 = 1/2$, $\alpha_1 = 10\sqrt{\max(m, n)}$, $\alpha_2 = 10$, and $\beta_k = (k-1)/(k+3)$ for $k \geq 1$. Note that the latter automatically implies $Y^0 = X$. We run the algorithm on two test datasets; inspired by Brunton and Kutz (2019), the first one is the Yale Faces B dataset (Lee et al. 2005), and the second one is a video sequence of a Cornell box with a moving

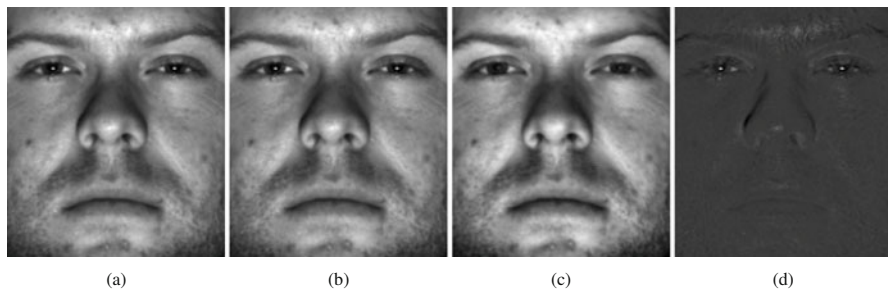


Fig. 1 From left to right: the first image of the Yale B faces database, its approximation which is the sum of a low-rank and a sparse matrix, the low-rank matrix, and the sparse matrix. (a) Original (b) Approximation (c) Low-rank part (d) Sparse part

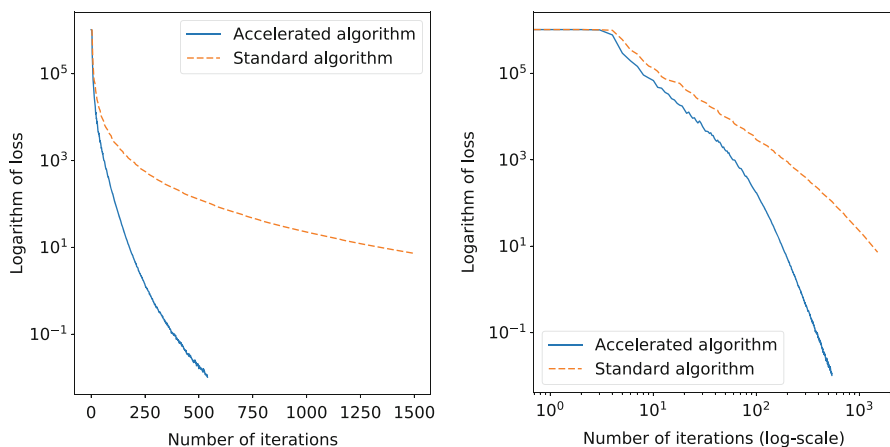


Fig. 2 This is an empirical validation of the different convergence rates of the linearized Bregman iteration and its accelerated counterpart (with regular scaling of the iterations on the left-hand side and a logarithmic scaling on the right-hand side)

shadow, from Benning et al. (2007). Figure 1 shows the first image of the Yale B faces database, its approximation, and its decomposition into a low-rank and a sparse part.

The more important aspect in terms of this review paper is certainly the comparison between the linearized Bregman iteration and its accelerated counterpart. A log-scale plot of the decrease of the loss function $\frac{1}{2} \|L + S - X\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm, over the course of the iterations of the two algorithms is visualized in Fig. 2. The plot is an empirical validation that (18) converges at rate $\mathcal{O}(1/k^2)$ as opposed to the $\mathcal{O}(1/k)$ rate of its non-accelerated counterpart.

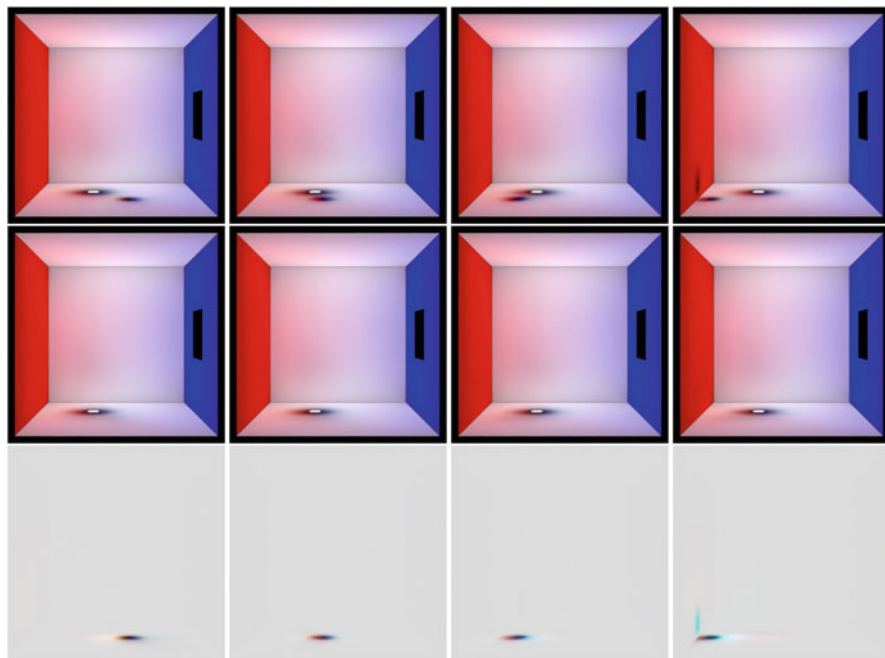


Fig. 3 First row: the 1st, 50th, 100th, and 150th frame of the original video sequence from Benning et al. (2007). Second row: the same frames of the computed low-rank part. Third row: the same frames of the computed sparse part

In Fig. 3 we see the 1st, 50th, 100th, and 150th frame of the original Cornell box video sequence from Benning et al. (2007), together with a low-rank approximation and a sparse component computed with the accelerated linearized Bregman iteration.

Deep Learning

Ever since Alexnet entered the scene in 2012 (Krizhevsky et al. 2012), thwarting then state-of-the-art image classification approaches in terms of accuracy in the process, deep neural networks (DNNs) have been central to research in computer vision and imaging. In this section, we merely want to support the analogy between incremental Bregman proximal methods and DNNs as shown in section “Deep Neural Networks” with a practical example, rather than engaging in a discussion of when and why DNNs based on (30) should be used or what advantages or shortcomings they possess compared to other neural network architectures. For a

comprehensive overview over developments in deep learning, we refer the reader to Goodfellow et al. (2016).

In this example, we set up a DNN-based auto-encoder for dimensionality reduction and compare it to classical dimensionality reduction via singular value decomposition. The auto-encoder is of the form

$$\begin{aligned} x^k &= (I + \partial \|\cdot\|_1)^{-1} (A_k x^{k-1} + b_k), \\ &= S_1 (A_k x^{k-1} + b_k), \end{aligned}$$

for $k \in \{1, 2, 3, 4\}$ and $x^0 = x$, where x denotes the input of the network, $A_k := \frac{1}{2}(M_k + M_k^T) \circ H_k$ for matrices $M_k \in \mathbb{R}^{m_k \times m_k}$ dimensions $m_1 = 196$, $m_2 = 49$, $m_3 = 196$, and $m_4 = 784$, and where H_1 and H_2 are two-dimensional average pooling operators with window size 2×2 and H_3 and H_4 are nearest-neighbor interpolation operators that upscale by a factor of two. The vectors $\{b_k\}_{k=1}^4$ are bias vectors of dimensions $\{m_k\}_{k=1}^4$, and the operator S_1 is the soft-shrinkage operator as described in section “[The Sparse Kaczmarz Method](#).” Please note that this auto-encoder architecture is of the form (30) and represents a parametrized mapping Φ_Θ from \mathbb{R}^{784} to \mathbb{R}^{784} , where $\Theta = (\{M_k\}_{k=1}^4, \{b_k\}_{k=1}^4)$ denotes the collection of parameters. We train the auto-encoder by minimizing the empirical risk based on the mean-squared error for a set of samples $\{x_i\}_{i=1}^s$, $s = 60000$, via stochastic gradient descent (which is the randomized version of (24)), i.e., we approximately estimate optimal parameters $\hat{\Theta}$ via

$$\hat{\Theta} = \arg \min_{\Theta} \frac{1}{2s} \sum_{i=1}^s (\Phi_\Theta(x_i) - x_i)^2.$$

We emphasize that the soft-thresholding activation function S_1 leaves Φ_Θ as not differentiable, which is why the application of (24) is technically a stochastic subgradient method. We train the auto-encoder with the help of PyTorch for a fixed number of epochs (500) and fixed step size $\tau = 2$ with batch size 100 on the MNIST training dataset (LeCun et al. 1998). In Fig. 4, we visualized several samples and the corresponding transformed outputs of the auto-encoder. In Fig. 5, we have visualized random images from the same dataset in comparison to their truncated singular value decomposition reconstructions where all but the first 49 singular values are cut off. As to be expected, nonlinear dimensionality reduction can outperform linear dimensionality reduction, achieving visually superior results for the same subspace dimensionality.

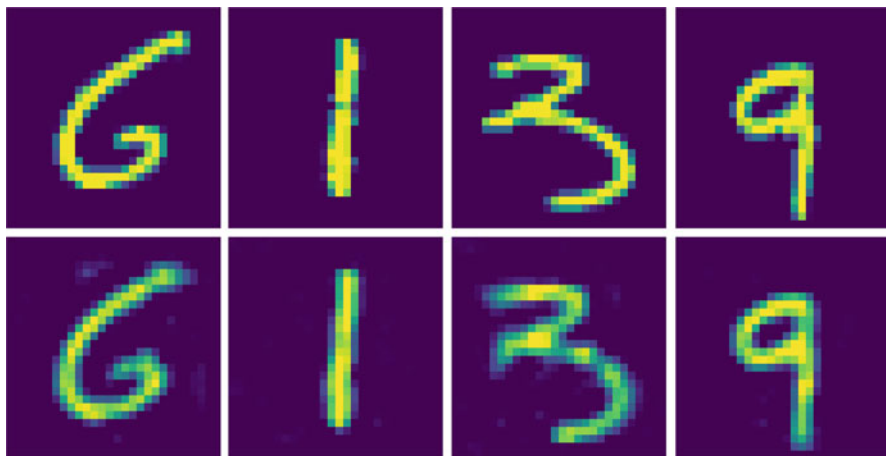


Fig. 4 Top row: random samples from the MNIST dataset. Bottom row: the corresponding approximations with the trained auto-encoder

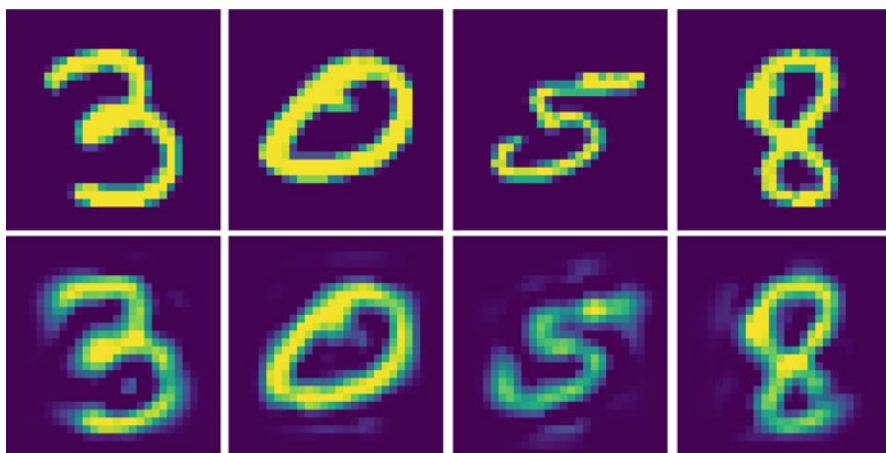


Fig. 5 Top row: random samples from the MNIST dataset. Bottom row: the corresponding approximations with the first 49 singular vectors

Student-t Regularized Image Denoising

In what follows, we apply BIA methods for solving a nonsmooth, nonconvex image denoising model, previously presented in Ochs et al. (2014). A priori knowledge of the noise distribution allows the use of Bregman functions $J(x)$ that exploit sparsity structures of the problem. As we will see, this yields significantly improved convergence rates in comparison with the default Itoh–Abe scheme

(i.e., $J(x) = \|x\|^2/2$). The application of the BIA method for this example was previously presented in Benning et al. (2020).

The objective function is given by

$$F : \mathbb{R}^n \rightarrow \mathbb{R}, \quad F(x) := \sum_{i=1}^N \varphi_i \Phi(K_i x) + \|x - x^\delta\|_1. \tag{52}$$

Here $\{K_i\}_{i=1}^N$ is a collection of linear filters, $(\varphi_i)_{i=1}^N \subset [0, \infty)$ are coefficients, $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is the nonconvex function based on the student-t distribution, defined as

$$\Phi(x) := \sum_{j=1}^n \psi(x_j), \quad \psi(x) := \log(1 + x^2),$$

and x^δ is an image corrupted by impulse noise (salt and pepper noise).

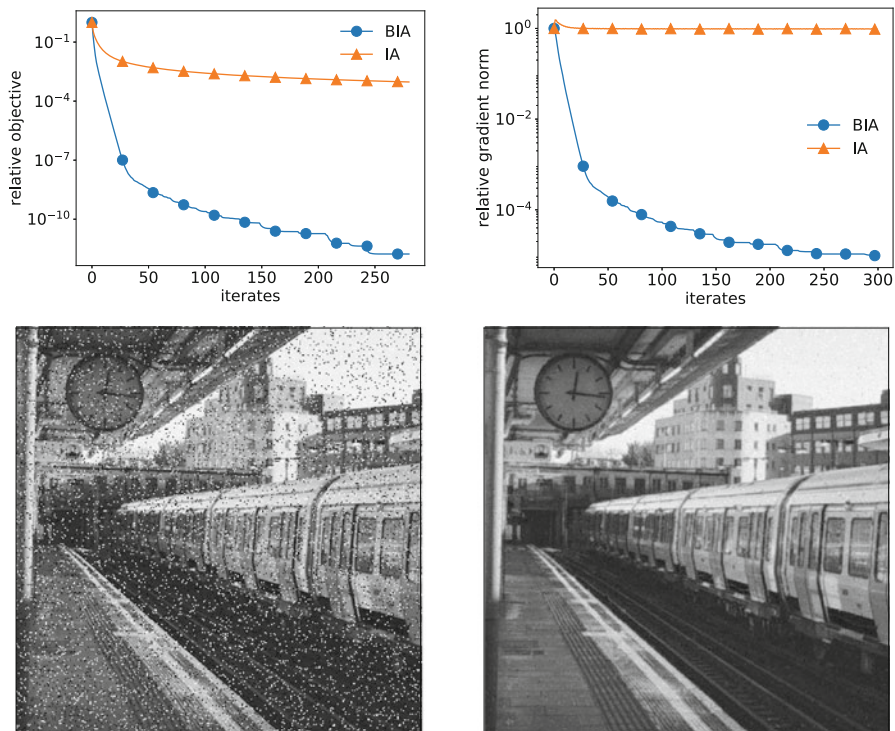


Fig. 6 Comparison of BIA and IA methods, for student-t regularized image denoising. First: convergence rate for relative objective. Second: convergence rate for relative gradient norm. Third: input data. Fourth: reconstruction

As impulse noise only affects a fraction of pixels, we use the data fidelity term $x \mapsto \|x - x^\delta\|_1$ to promote sparsity of $x^* - x^\delta$ for $x^* \in \arg \min F(x)$. As linear filters, we consider the simple case of finite difference approximations to first-order derivatives of x . We note that by applying a gradient flow to this regularization function, we observe a similarity to Perona–Malik diffusion (Perona and Malik 1990).

For the BIA method, we consider the Bregman function

$$J(x) := \frac{1}{2} \|x\|^2 + \gamma \|x - x^\delta\|_1,$$

to account for the sparsity of the residual $x^* - x^\delta$ and compare the method to the regular Itoh–Abe discrete gradient method (abbreviated to IA).

We set the starting point $x^0 = x^\delta$ and the parameters to $\tau_k = 1$ for all k , $\gamma = 0.5$, and $\varphi_i = 2$, $i = 1, 2$. For the impulse noise, we use a noise density of 10%. In the case where x_i^{k+1} is not set to x_i^δ , we use the scalar root solver *scipy.optimize.brenth* on Python. Otherwise, the updates are in closed form.

See Fig. 6 for numerical results. By gradient norm, we mean $\text{dist}(\partial^C F(x^k), 0)$.

Conclusions and Outlook

In this review paper, we gave a selective overview on a range of topics concerning adaptations of Bregman algorithms suited for large-scale problems in imaging. In particular, we discussed Nesterov accelerations of the Bregman (proximal) gradient or linearized Bregman iteration, incremental variants of Bregman methods, and coordinate descent-type Bregman algorithms with a particular focus on a Bregman Itoh–Abe scheme.

Despite the variety of numerous adaptations, a lot of research on Bregman algorithms is yet to be done. We conclude this chapter by discussing some open problems as well as ongoing directions of research.

Examples of open problems are adaptations for nonconvex objectives (following recent advances in papers such as Ahookhosh et al. 2019), extensions to nonlinear inverse problems (Bachmayr and Burger 2009) or inverse problems with non-quadratic data fidelity terms (Benning and Burger 2011) and the closer analysis and numerical realization of neural network architectures inspired by Bregman algorithms. We also want to emphasize that Bregman variants of incremental or stochastic variants of ADMM or the PDHG method in the spirit of Ouyang et al. (2013) and Chambolle et al. (2018) are still open problems.

Another important topic of ongoing research is to understand the scope for and limitations of accelerated Bregman methods, as stated by Teboulle (2018). Dragomir et al. (2019) point out the open problem of whether accelerated Bregman methods are possible if one makes further assumptions on the objective and Bregman functions or by allowing access to second-order information. Another interesting approach is to consider ODEs – see, e.g., Krichene et al. (2015) in which Krichene et

al. investigate accelerating mirror descent via the ODE interpretation of Nesterov's acceleration (Su et al. 2016).

Going from optimization to sampling, some recent papers consider methods for sampling of distributions which incorporate elements of mirror descent in the underlying dynamics. Hsieh et al. (2018) propose a framework for sampling from constrained distributions, termed *mirrored Langevin dynamics*. In a similar vein, Zhang et al. (2020) propose a Mirror Langevin Monte Carlo algorithm, to improve the smoothness and convexity properties for the distribution.

Acknowledgments MB thanks Queen Mary University of London for their support. ESR acknowledges support from the London Mathematical Society.

References

- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Ahooikhosh, M., Hien, L.T.K., Gillis, N., Patrinos, P.: Multi-block Bregman proximal alternating linearized minimization and its application to sparse orthogonal nonnegative matrix factorization. *arXiv preprint arXiv:1908.01402* (2019)
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numerica* **28**, 1–174 (2019)
- Attouch, H., Buttazzo, G., Michaille, G.: Variational analysis in Sobolev and BV spaces: applications to PDEs and optimization. *SIAM* (2014)
- Azizan, N., Hassibi, B.: Stochastic gradient/mirror descent: Minimax optimality and implicit regularization. *arXiv preprint arXiv:1806.00952* (2018)
- Bachmayr, M., Burger, M.: Iterative total variation schemes for nonlinear inverse problems. *Inverse Prob.* **25**(10), 105004 (2009)
- Bauschke, H.H., Bolte, J., Teboulle, M.: A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Math. Oper. Res.* **42**(2), 330–348 (2017)
- Bauschke, H.H., Borwein, J.M., Combettes, P.L.: Bregman monotone optimization algorithms. *SIAM J. Control. Optim.* **42**(2), 596–636 (2003)
- Beck, A.: *First-Order Methods in Optimization*, Vol. 25. *SIAM* (2017)
- Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.* **31**(3), 167–175 (2003)
- Beck, A., Tsetuashvili, L.: On the convergence of block coordinate descent type methods. *SIAM J. Optim.* **23**(4), 2037–2060 (2013)
- Ben-Tal, A., Margalit, T., Nemirovski, A.: The ordered subsets mirror descent optimization method with applications to tomography. *SIAM J. Optim.* **12**(1), 79–108 (2001)
- Benning, M., Betcke, M., Ehrhardt, M., Schönlieb, C.-B.: Gradient descent in a generalised Bregman distance framework. In: *Geometric Numerical Integration and its Applications*, Vol. 74, pp. 40–45. *MI Lecture Notes series of Kyushu University* (2017)
- Benning, M., Betcke, M.M., Ehrhardt, M.J., Schönlieb, C.-B.: Choose your path wisely: gradient descent in a Bregman distance framework. *SIAM Journal on Imaging Sciences (SIIMS)*. *arXiv preprint arXiv:1712.04045* (2017)
- Benning, M., Burger, M.: Error estimates for general fidelities. *Electron. Trans. Numer. Anal.* **38**(44–68), 77 (2011)
- Benning, M., Burger, M.: Modern regularization methods for inverse problems. *Acta Numerica* **27**, 1–111 (2018)
- Benning, M., Knoll, F., Schönlieb, C.-B., Valkonen, T.: Preconditioned ADMM with nonlinear operator constraint. In: *IFIP Conference on System Modeling and Optimization*, pp. 117–126. Springer (2015)

- Benning, M., Lee, E., Pao, H., Yacoubou-Djima, K., Wittman, T., Anderson, J.: Statistical filtering of global illumination for computer graphics. IPAM Research in Industrial Projects for Students (RIPS) Report (2007)
- Benning, M., Riis, E.S., Schönlieb, C.-B.: Bregman Itoh–Abe methods for sparse optimisation. In print: *J. Math. Imaging Vision* (2020)
- Bertocchi, C., Chouzenoux, E., Corbineau, M.-C., Pesquet, J.-C., Prato, M.: Deep unfolding of a proximal interior point method for image restoration. *Inverse Prob.* **36**, 034005 (2019)
- Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optim. Mach. Learn.* **2010**(1–38), 3 (2011)
- Bertsekas, D.P.: Incremental proximal methods for large scale convex optimization. *Math. Program.* **129**(2), 163 (2011)
- Blatt, D., Hero, A.O., Gauchman, H.: A convergent incremental gradient method with a constant step size. *SIAM J. Optim.* **18**(1), 29–51 (2007)
- Bohte, J., Sabach, S., Teboulle, M., Vaisbourd, Y.: First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM J. Optim.* **28**(3), 2131–2151 (2018)
- Bonettini, S., Rebegoldi, S., Ruggiero, V.: Inertial variable metric techniques for the inexact forward–backward algorithm. *SIAM J. Sci. Comput.* **40**(5), A3180–A3210 (2018)
- Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: A family of variable metric proximal methods. *Math. Program.* **68**(1–3), 15–47 (1995)
- Bouwmans, T., Javed, S., Zhang, H., Lin, Z., Otazo, R.: On the applications of robust PCA in image and video processing. *Proc. IEEE* **106**(8), 1427–1457 (2018)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* **3**(1), 1–122 (2011)
- Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.* **7**(3), 200–217 (1967)
- Brunton, S.L., Kutz, J.N.: *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press (2019)
- Burbea, J., Rao, C.: On the convexity of higher order Jensen differences based on entropy functions (corresp.). *IEEE Trans. Inf. Theory* **28**(6), 961–963 (1982)
- Burbea, J., Rao, C.: On the convexity of some divergence measures based on entropy functions. *IEEE Trans. Inf. Theory* **28**(3), 489–495 (1982)
- Burger, M.: Bregman distances in inverse problems and partial differential equations. In: *Advances in Mathematical Modeling, Optimization and Optimal Control*, pp. 3–33. Springer (2016)
- Burger, M., Frick, K., Osher, S., Scherzer, O.: Inverse total variation flow. *Multiscale Model. Simul.* **6**(2), 366–395 (2007)
- Burger, M., Gilboa, G., Moeller, M., Eckardt, L., Cremers, D.: Spectral decompositions using one-homogeneous functionals. *SIAM J. Imag. Sci.* **9**(3), 1374–1408 (2016)
- Burger, M., Gilboa, G., Osher, S., Xu, J.: Nonlinear inverse scale space methods. *Commun. Math. Sci.* **4**(1), 179–212 (2006)
- Burger, M., Moeller, M., Benning, M., Osher, S.: An adaptive inverse scale space method for compressed sensing. *Math. Comput.* **82**(281), 269–299 (2013)
- Burger, M., Osher, S.: Convergence rates of convex variational regularization. *Inverse Prob.* **20**(5), 1411 (2004)
- Burger, M., Resmerita, E., He, L.: Error estimation for Bregman iterations and inverse scale space methods in image restoration. *Computing* **81**(2–3), 109–135 (2007)
- Cai, J.-F., Osher, S., Shen, Z.: Convergence of the linearized Bregman iteration for ℓ^1 -norm minimization. *Math. Comput.* **78**(268), 2127–2136 (2009)
- Cai, J.-F., Osher, S., Shen, Z.: Linearized Bregman iterations for compressed sensing. *Math. Comput.* **78**(267), 1515–1536 (2009)
- Cai, J.-F., Osher, S., Shen, Z.: Linearized Bregman iterations for frame-based image deblurring. *SIAM J. Imag. Sci.* **2**(1), 226–252 (2009)

- Calatroni, L., Garrigos, G., Rosasco, L., Villa, S.: Accelerated iterative regularization via dual diagonal descent. arXiv preprint arXiv:1912.12153 (2019)
- Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 11 (2011)
- Censor, Y., Lent, A.: An iterative row-action method for interval convex programming. J. Optim. Theory Appl. **34**(3), 321–353 (1981)
- Censor, Y., Stavros Zenios, A.: Proximal minimization algorithm with d -functions. J. Optim. Theory Appl. **73**(3), 451–464 (1992)
- Chambolle, A., Dossal, C.: On the convergence of the iterates of the “fast iterative shrinkage/thresholding algorithm. J. Optim. Theory Appl. **166**(3), 968–982 (2015)
- Chambolle, A., Ehrhardt, M.J., Richtárik, P., Carola-Schonlieb, B.: Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. SIAM J. Optim. **28**(4), 2783–2808 (2018)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vision **40**(1), 120–145 (2011)
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. Acta Numerica **25**, 161–319 (2016)
- Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal–dual algorithm. Math. Prog. **159**(1–2), 253–287 (2016)
- Chavent, G., Kunisch, K.: Regularization of linear least squares problems by total bounded variation. ESAIM Control Optim. Calc. Var. **2**, 359–376 (1997)
- Chen, G., Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using Bregman functions. SIAM J. Optim. **3**(3), 538–543 (1993)
- Chouzenoux, E., Pesquet, J.-C., Repetti, A.: Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. J. Optim. Theory Appl. **162**(1), 107–132 (2014)
- Clarke, F.H.: Optimization and Nonsmooth Analysis. Classics in Applied Mathematics, 1st edn. SIAM, Philadelphia (1990)
- Clason, C., Mazurenko, S., Valkonen, T.: Acceleration and global convergence of a first-order primal-dual method for nonconvex problems. SIAM J. Optim. **29**(1), 933–963 (2019)
- Clason, C., Valkonen, T.: Primal-dual extragradient methods for nonlinear nonsmooth PDE-constrained optimization. SIAM J. Optim. **27**(3), 1314–1339 (2017)
- Combettes, P.L., Pesquet, J.-C.: Deep neural network structures solving variational inequalities. arXiv preprint arXiv:1808.07526 (2018)
- Combettes, P.L., Vũ, B.C.: Variable metric forward–backward splitting with applications to monotone inclusions in duality. Optimization **63**(9), 1289–1318 (2014)
- Corona, V., Benning, M., Ehrhardt, M.J., Gladden, L.F., Mair, R., Recí, A., Sederman, A.J., Reichelt, S., Schönlieb, C.-B.: Enhancing joint reconstruction and segmentation with non-convex Bregman iteration. Inverse Prob. **35**(5), 055001 (2019)
- Corona, V., Benning, M., Gladden, L.F., Recí, A., Sederman, A.J., Schoenlieb, C.-B.: Joint phase reconstruction and magnitude segmentation from velocity-encoded MRI data. arXiv preprint arXiv:1908.05285 (2019)
- Doan, T.T., Bose, S., Nguyen, D.H., Beck, C.L.: Convergence of the iterates in mirror descent methods. IEEE Control Syst. Lett. **3**(1), 114–119 (2018)
- Dragomir, R.-A., Taylor, A., d’Aspremont, A., Bolte, J.: Optimal complexity and certification of Bregman first-order methods. arXiv preprint arXiv:1911.08510 (2019)
- Duchi, J.C., Agarwal, A., Johansson, M., Jordan, M.I.: Ergodic mirror descent. SIAM J. Optim. **22**(4), 1549–1578 (2012)
- Eckstein, J.: Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming. Math. Oper. Res. **18**(1), 202–226 (1993)
- Ehrhardt, M.J., Riis, E.S., Ringholm, T., Schönlieb, C.-B.: A geometric integration approach to smooth optimisation: Foundations of the discrete gradient method. ArXiv e-prints (2018)

- Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.* **3**(4), 1015–1046 (2010)
- Frankel, P., Garrigos, G., Peypouquet, J.: Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *J. Optim. Theory Appl.* **165**(3), 874–900 (2015)
- Frerix, T., Möllenhoff, T., Moeller, M., Cremers, D.: Proximal backpropagation. arXiv preprint arXiv:1706.04638 (2017)
- Frick, K., Scherzer, O.: Convex inverse scale spaces. In: *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 313–325. Springer (2007)
- Gabay, D.: Chapter ix applications of the method of multipliers to variational inequalities. In: *Studies in Mathematics and Its Applications*, Vol. 15, pp. 299–331. Elsevier (1983)
- Gao, T., Lu, S., Liu, J., Chu, C.: Randomized Bregman coordinate descent methods for non-Lipschitz optimization. arXiv preprint arXiv:2001.05202 (2020)
- Garrigos, G., Rosasco, L., Villa, S.: Iterative regularization via dual diagonal descent. *J. Math. Imaging Vision* **60**(2), 189–215 (2018)
- Gilboa, G., Moeller, M., Burger, M.: Nonlinear spectral analysis via one-homogeneous functionals: Overview and future prospects. *J. Math. Imaging Vision* **56**(2), 300–319 (2016)
- Goldstein, T., Li, M., Yuan, X.: Adaptive primal-dual splitting methods for statistical learning and image processing. In: *Advances in Neural Information Processing Systems*, pp. 2089–2097 (2015)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016)
- Gordon, R., Bender, R., Herman, G.T.: Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theor. Biol.* **29**(3), 471–481 (1970)
- Gower, R.M., Richtárik, P.: Randomized iterative methods for linear systems. *SIAM J. Matrix Anal. Appl.* **36**(4), 1660–1690 (2015)
- Grimm, V., McLachlan, R.I., McLaren, D.I., Quispel, G.R.W., Schönlieb, C.-B.: Discrete gradient methods for solving variational image regularisation models. *J. Phys. A* **50**(29), 295201 (2017)
- Gutman, D.H., Peña, J.F.: A unified framework for Bregman proximal methods: subgradient, gradient, and accelerated gradient schemes. arXiv preprint arXiv:1812.10198 (2018)
- Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Vol. 31, 2nd edn. Springer Science & Business Media, Berlin (2006)
- Hanzely, F., Richtárik, P., Xiao, L.: Accelerated Bregman proximal gradient methods for relatively smooth convex optimization. arXiv preprint arXiv:1808.03045 (2018)
- Hellinger, E.: Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik (Crelles Journal)* **1909**(136), 210–271 (1909)
- Hiriart-Urruty, J.-B., Lemaréchal, C.: *Convex analysis and minimization algorithms I: Fundamentals*, volume 305 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, 2nd edn. Springer, Berlin (1993)
- Hohage, T., Homann, C.: A generalization of the chambolle-pock algorithm to Banach spaces with applications to inverse problems. arXiv preprint arXiv:1412.0126 (2014)
- Hsieh, Y.-P., Kavis, A., Rolland, P., Cevher, V.: Mirrored Langevin dynamics. In: *Advances in Neural Information Processing Systems*, pp. 2878–2887 (2018)
- Hua, X., Yamashita, N.: Block coordinate proximal gradient methods with variable Bregman functions for nonsmooth separable optimization. *Math. Program.* **160**(1–2), 1–32 (2016)
- Huang, B., Ma, S., Goldfarb, D.: Accelerated linearized Bregman method. *J. Sci. Comput.* **54**(2–3), 428–453 (2013)
- Itakura, F.: Analysis synthesis telephony based on the maximum likelihood method. In: *The 6th International Congress on Acoustics*, 1968, pp. 280–292 (1968)
- Itoh, T., Abe, K.: Hamiltonian-conserving discrete canonical equations based on variational difference quotients. *J. Comput. Phys.* **76**(1), 85–102 (1988)

- Juditsky, A., Nemirovski, A., et al.: First order methods for nonsmooth convex large-scale optimization, I: General purpose methods. *Optim. Mach. Learn.* 121–148 (2011). <https://doi.org/10.7551/mitpress/8996.003.0007>
- Kaczmarz, M.S.: Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin Internationale de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* **35**, 355–357 (1937)
- Kiwiel, K.C.: Free-steering relaxation methods for problems with strictly convex costs and linear constraints. *Math. Oper. Res.* **22**(2), 326–349 (1997)
- Kiwiel, K.C.: Proximal minimization methods with generalized Bregman functions. *SIAM J. Control. Optim.* **35**(4), 1142–1168 (1997)
- Kobler, E., Klatzer, T., Hammernik, K., Pock, T.: Variational networks: connecting variational methods and deep learning. In: *German Conference on Pattern Recognition*, pp. 281–293. Springer (2017)
- Krichene, W., Bayen, A., Bartlett, P.L.: Accelerated mirror descent in continuous and discrete time. In: *Advances in Neural Information Processing Systems*, pp. 2845–2853 (2015)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
- LeCun, Y., Cortes, C., Burges, C.J.C.: The mnist database of handwritten digits (1998). <http://yann.lecun.com/exdb/mnist> 10:34 (1998)
- Lee, K.-C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 684–698 (2005)
- Li, G., Pong, T.K.: Global convergence of splitting methods for nonconvex composite optimization. *SIAM J. Optim.* **25**(4), 2434–2460 (2015)
- Li, H., Schwab, J., Antholzer, S., Haltmeier, M.: Nett: Solving inverse problems with deep neural networks. *Inverse Prob.* **36**, 065005 (2020)
- Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16**(6), 964–979 (1979)
- Lorenz, D.A., Schöpfer, F., Wenger, S.: The linearized Bregman method via split feasibility problems: Analysis and generalizations. *SIAM J. Imag. Sci.* **7**(2), 1237–1262 (2014)
- Lorenz, D.A., Wenger, S., Schöpfer, F., Magnor, M.: A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing. *arXiv e-prints* (2014)
- Prasanta, P.C.: On the generalized distance in statistics. *National Institute of Science of India* (1936)
- Matet, S., Rosasco, L., Villa, S., Vu, B.L.: Don't relax: Early stopping for convex regularization. *arXiv preprint arXiv:1707.05422* (2017)
- McLachlan, R.I., Quispel, G.R.W.: Six lectures on the geometric integration of ODEs, pp. 155–210. *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge (2001)
- McLachlan, R.I., Quispel, G.R.W., Robidoux, N.: Geometric integration using discrete gradients. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **357**(1754), 1021–1045 (1999)
- Miyatake, Y., Sogabe, T., Zhang, S.-L.: On the equivalence between SOR-type methods for linear systems and the discrete gradient methods for gradient systems. *J. Comput. Appl. Math.* **342**, 58–69 (2018)
- Moeller, M., Benning, M., Schönlieb, C., Cremers, D.: Variational depth from focus reconstruction. *IEEE Trans. Image Process.* **24**(12), 5369–5378 (2015)
- Möllenhoff, T., Strelakovsky, E., Moeller, M., Cremers, D.: The primal-dual hybrid gradient method for semiconvex splittings. *SIAM J. Imag. Sci.* **8**(2), 827–857 (2015)
- Moreau, J.-J.: Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France* **93**, 273–299 (1965)
- Morozov, V.A.: Regularization of incorrectly posed problems and the choice of regularization parameter. *USSR Comput. Math. Math. Phys.* **6**(1), 242–251 (1966)
- Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814 (2010)

- Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.* **19**(4), 1574–1609 (2009)
- Nemirovsky, A.S., Yudin, D.B.: Problem complexity and method efficiency in optimization (1983)
- Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$. In: *Doklady AN USSR*, Vol. 269, pp. 543–547 (1983)
- Nesterov, Y.: Primal-dual subgradient methods for convex problems. *Math. Program.* **120**(1), 221–259 (2009)
- Neubauer, A.: On Nesterov acceleration for Landweber iteration of linear ill-posed problems. *J. Inverse Ill-posed Prob.* **25**(3), 381–390 (2017)
- Nielsen, F., Boltz, S.: The Burbea-Rao and Bhattacharyya centroids. *IEEE Trans. Inf. Theory* **57**(8), 5455–5466 (2011)
- Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM J. Imag. Sci.* **7**(2), 1388–1419 (2014)
- Ochs, P., Ranftl, R., Brox, T., Pock, T.: Bilevel optimization with nonsmooth lower level problems. In: *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 654–665. Springer (2015)
- Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.* **4**(2), 460–489 (2005)
- Oswald, P., Zhou, W.: Convergence analysis for Kaczmarz-type methods in a Hilbert space framework. *Linear Algebra Appl.* **478**, 131–161 (2015)
- Ouyang, H., He, N., Tran, L., Gray, A.: Stochastic alternating direction method of multipliers. In: *International Conference on Machine Learning*, pp. 80–88 (2013)
- Parikh, N., Boyd, S., et al.: Proximal algorithms. *Found. Trends@ Optim.* **1**(3), 127–239 (2014)
- Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990)
- Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 1133–1140. IEEE (2009)
- Resmerita, E., Scherzer, O.: Error estimates for non-quadratic regularization and the relation to enhancement. *Inverse Prob.* **22**(3), 801 (2006)
- Riis, E.S., Ehrhardt, M.J., Quispel, G.R.W., Schönlieb, C.-B.: A geometric integration approach to nonsmooth, nonconvex optimisation. *Foundations of Computational Mathematics (FOCM)*. ArXiv e-prints (2018)
- Ringholm, T., Lazić, J., Schönlieb, C.-B.: Variational image regularization with Euler’s elastica using a discrete gradient scheme. *SIAM J. Imag. Sci.* **11**(4), 2665–2691 (2018)
- Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**, 400–407 (1951)
- Scherzer, O., Groetsch, C.: Inverse scale space theory for inverse problems. In: *International Conference on Scale-Space Theories in Computer Vision*, pp. 317–325. Springer (2001)
- Marie Schmidt, F., Benning, M., Schönlieb, C.-B.: Inverse scale space decomposition. *Inverse Prob.* **34**(4), 179–212 (2018)
- Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. *Math. Program.* **162**(1–2), 83–112 (2017)
- Schöpfer, F., Lorenz, D.A.: Linear convergence of the randomized sparse Kaczmarz method. *Math. Program.* **173**(1), 509–536 (2019)
- Schuster, T., Kaltenbacher, B., Hofmann, B., Kazimierski, K.S.: Regularization methods in Banach spaces, Vol. 10. Walter de Gruyter (2012)
- Su, W., Boyd, S., Candes, E.J.: A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.* **17**(153), 1–43 (2016)
- Teboulle, M.: Entropic proximal mappings with applications to nonlinear programming. *Math. Oper. Res.* **17**(3), 670–690 (1992)
- Teboulle, M.: A simplified view of first order methods for optimization. *Math. Program.* **170**(1), 67–96 (2018)
- Teboulle, M., Chen, G.: Convergence analysis of a proximal-like minimization algorithm using Bregman function. *SIAM J. Optim.* **3**(3), 538–543 (1993)

- Valkonen, T.: A primal–dual hybrid gradient method for nonlinear operators with applications to MRI. *Inverse Prob.* **30**(5), 055012 (2014)
- Wang, H., Banerjee, A.: Bregman alternating direction method of multipliers. In: *Advances in Neural Information Processing Systems*, pp. 2816–2824 (2014)
- Widrow, B., Hoff, M.E.: Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs (1960)
- Wright, S.J.: Coordinate descent algorithms. *Math. Program.* **1**(151), 3–34 (2015)
- Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.* **11**, 2543–2596 (2010)
- Yin, W.: Analysis and generalizations of the linearized Bregman method. *SIAM J. Imag. Sci.* **3**(4), 856–877 (2010)
- Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing. *SIAM J. Imag. Sci.* **1**(1), 143–168 (2008)
- Yosida, K.: *Functional Analysis*. Springer (1964)
- Young, D.M.: *Iterative Solution of Large Linear Systems*. Computer Science and Applied Mathematics, 1st edn. Academic Press, Inc., Orlando (1971)
- Zhang, H., Dai, Y.-H., Guo, L., Peng, W.: Proximal-like incremental aggregated gradient method with linear convergence under Bregman distance growth conditions. arXiv preprint arXiv:1711.01136 (2017)
- Zhang, K.S., Peyré, G., Fadili, J., Pereyra, M.: Wasserstein control of mirror Langevin Monte Carlo. arXiv e-prints (2020)
- Zhang, X., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.* **46**(1), 20–46 (2011)
- Zhou, Z., Mertikopoulos, P., Bambos, N., Boyd, S., Glynn, P.W.: Stochastic mirror descent in variationally coherent optimization problems. In: *Advances in Neural Information Processing Systems*, pp. 7040–7049 (2017)
- Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report* 34 (2008)



Fast Iterative Algorithms for Blind Phase Retrieval: A Survey

4

Huibin Chang, Li Yang, and Stefano Marchesini

Contents

Introduction	140
Mathematical Formula and Nonlinear Optimization Model for BPR	141
Mathematical Formula	141
Optimization Problems and Proximal Mapping	145
Fast Iterative Algorithms	147
Alternating Projection (AP) Algorithms	147
ePIE-Type Algorithms	149
Proximal Algorithms	151
ADMM	153
Convex Programming	156
Second-Order Algorithm Using Hessian	160
Subspace Method	162
Discussions	167
Experimental Issues	167
Theoretical Analysis	168
Further Discussions	169
Conclusions	170
References	171

Abstract

In nanoscale imaging technique and ultrafast laser, the reconstruction procedure is normally formulated as a blind phase retrieval (BPR) problem, where one has to recover both the sample and the probe (pupil) jointly from phaseless data. This

H. Chang (✉) · L. Yang

School of Mathematical Sciences, Tianjin Normal University, Tianjin, China

S. Marchesini

SLAC National Laboratory, Menlo Park, CA, USA

© Springer Nature Switzerland AG 2023

K. Chen et al. (eds.), *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, https://doi.org/10.1007/978-3-030-98661-2_116

139

survey first presents the mathematical formula of BPR and related nonlinear optimization problems and then gives a brief review of the recent iterative algorithms. It mainly consists of three types of algorithms, including the operator-splitting-based first-order optimization methods, second-order algorithm with Hessian, and subspace methods. The future research directions for experimental issues and theoretical analysis are further discussed.

Introduction

Phase retrieval (PR) plays a key role in nanoscale imaging technique (Pfeiffer 2018; Elser et al. 2018; Zheng et al. 2021; Gürsoy et al. 2022) and ultrafast laser (Trebino et al. 1997). Retrieving the images of the sample from phaseless data is a long-standing problem. Generally speaking, designing fast and reliable algorithms is challenging since directly solving the quadratic polynomials of PR is NP hard and the involved optimization problem is nonconvex and possibly nonsmooth. Thus, it has drawn the attentions of researchers for several decades (Luke 2005; Shechtman et al. 2015; Grohs et al. 2020; Fannjiang and Strohmer 2020). Among the general PR problems, besides the recovery of the sample, it is also of great importance to reconstruct the probes. The motivation of blind recovery is twofold: (1) characteristics of the probe (wave front sensing) and (2) improving the reconstruction quality of the sample. Essentially in practice, as the probe is almost never completely known, one has to solve such blind phase retrieval (BPR) problem, e.g., in coherent diffractive imaging (CDI) (Thibault and Guizar-Sicairos 2012), convention ptychography imaging (Thibault et al. 2009; Maiden and Rodenburg 2009), Fourier ptychography (Zheng et al. 2013; Ou et al. 2014), convolutional PR (Ahmed et al. 2018), frequency-resolved optical gating (Trebino et al. 1997), and others.

An early work by Chapman (1996) to solve the blind problem used the Wigner-distribution deconvolution method to retrieve the probe. In the optics community, alternating projection (AP) algorithms are very popular for nonblind PR problems (Marchesini 2007; Elser et al. 2018). Some AP algorithms have also been applied to BPR problems, e.g., Douglas-Rachford (DR)-based algorithm (Thibault et al. 2009), extended ptychographic engine (ePIE) and variants (Maiden and Rodenburg 2009; Maiden et al. 2017), and relaxed averaged alternating reflection (Luke 2005)-based projection algorithm (Marchesini et al. 2016). More advanced first-order optimization method includes proximal algorithms, Hesse et al. (2015), Yan (2020), and Huang et al. (2021), alternating direction of multiplier methods (ADMMs) (Chang et al. 2019a; Fannjiang and Zhang 2020), and convex programming method (Ahmed et al. 2018). To further accelerate the first-order optimization, several second-order algorithms utilizing the Hessian have also been developed (Qian et al. 2014; Yeh et al. 2015; Ma et al. 2018; Gao and Xu 2017; Kandel et al. 2021). Moreover, the subspace methods (Xin et al. 2021) were successfully applied to the BPR as Thibault and Guizar-Sicairos (2012), Chang et al. (2019a), and Fung and Wendy (2020).

The purpose of the survey is to give a brief review of the recent iterative algorithms for BPR problem, so as to provide instructions for practical use and draw attentions of applied mathematician for further improvement. The remainder of the survey is organized as follows: Section “[Mathematical Formula and Nonlinear Optimization Model for BPR](#)” gives the mathematical formula for BPR and related nonlinear optimization models, as well as the closed-form expression of the proximal mapping. Fast iterative algorithms are reviewed in Section “[Fast Iterative Algorithms](#)”. Section “[Discussions](#)” further discusses the experimental issues and theoretical analysis. Section “[Conclusions](#)” summarizes this survey.

Mathematical Formula and Nonlinear Optimization Model for BPR

First, introduce the general nonblind PR problem in the discrete setting. By introducing a linear operator $A \in \mathbb{C}^{m,n}$, for the sample of interest $u \in \mathbb{C}^n$, experimental instruments usually collect the quadratic phaseless data $f \in \mathbb{R}^m$ as below:

$$f = |Au|^2, \quad (1)$$

in the ideal situation. However, noise contamination is evitable in practice (Chang et al. 2018b) as

$$f_{\text{noise}} = \text{Poi}(|Au|^2), \quad (2)$$

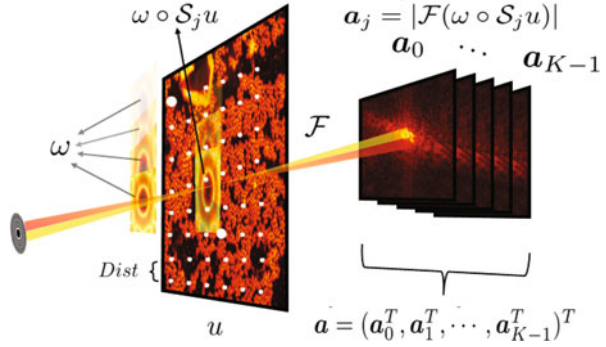
where Poi denotes the random variable following i.i.d Poisson distribution. See more advanced models for practical noise as outliers and structured and randomly distributed uncorrelated noise sources in Godard et al. (2012), Reinhardt et al. (2017), Wang et al. (2017), Odstrčil et al. (2018), Chang et al. (2019b), and references therein.

Mathematical Formula

State the BPR problem starting from convention ptychography (Rodenburg 2008), since the principle of other BPR problems can be explained in a similar manner, all of which can be unified as the blind recovery problem.

As shown in Fig. 1, a detector in the far field measures a series of phaseless intensities, by letting a localized coherent X-ray probe w scan through the sample u . Let the 2D image and the localized 2D probe denote as $u \in \mathbb{C}^n$ with $\sqrt{n} \times \sqrt{n}$ pixels and $w \in \mathbb{C}^{\bar{m}}$ with $\sqrt{\bar{m}} \times \sqrt{\bar{m}}$ pixels, respectively. Here both the sample and the probe are rewritten as vectors by a lexicographical order. Let $f_j^P \in \mathbb{R}_+^{\bar{m}} \forall 0 \leq j \leq J - 1$ denote the phaseless measurements satisfying

Fig. 1 Ptychographic PR (far field): A stack of phaseless data $f_j := \mathbf{a}_j^2$ is collected, with w being the localized coherent probe and u being the image of interest (sample). The white dots represent the scanning lattice points, with $Dist$ denoting the sliding distance between centers of two adjacent frames



$$f_j^P = |\mathcal{F}(w \circ \mathcal{S}_j u)|^2, \quad (3)$$

where the symbols $|\cdot|$, $(\cdot)^2$, and \circ represent the element-wise absolute value and square of a vector and the element-wise multiplication of two vectors, respectively, the symbol $\mathcal{S}_j \in \mathbb{R}^{\tilde{m} \times n}$ represents a matrix with binary elements extracting a patch (with the index j and size \tilde{m}) from the entire sample, and the symbol \mathcal{F} denotes the normalized discrete Fourier transformation (DFT). In practice, to get an accurate estimate of the probe, one has to solve a blind ptychographic PR problem. Note that the coherent CDI problem (Thibault and Guizar-Sicairos 2012) can be interpreted as a special blind ptychography problem with only one scanned frame ($J = 1$).

A recent super-resolution technique based on visible light called as the Fourier ptychography method (FP) has been developed by Zheng et al. (2013) and quickly spreads out for fruitful applications (Zheng et al. 2021). Letting w and u (here reuse the notations for simplicity) be the point spread function (PSF) of the imaging system and the sample of interest, the collected phaseless data f_j^{FP} of FP can be expressed as

$$f_j^{FP} = |\mathcal{F}^{-1}(\bar{w} \circ \mathcal{S}_j \bar{u})|^2 \text{ for } 0 \leq j \leq J - 1$$

with $\bar{w} := \mathcal{F}w$ and $\bar{u} := \mathcal{F}u$.

Some similar problems dubbed as “convolutional PR” were recently studied (Qu et al. 2017, 2019; Ahmed et al. 2018). Given the sample u and the convolution kernel κ , the phaseless measurement f^{Cov} is given as

$$f^{\text{Cov}} = |\kappa \circledast u|^2 \quad \text{Qu et al. (2017, 2019)}$$

or

$$f^{\text{Cov}} = |\mathcal{F}(\kappa \circledast u)|^2 \quad \text{Ahmed et al. (2018),} \quad (4)$$

where the symbol \circledast denotes the convolution.

Other interesting blind problem for full characterization of ultrashort optical pulses is to use frequency-resolved optical gating (FROG) (Trebino et al. 1997; Bendory et al. 2017; Kane and Vakhtin 2021). The phaseless measurement for a typical SHG-FROG can be obtained as

$$f_j^{\text{FROG}} = |\mathcal{F}(u \circ \mathcal{T}_j u)|^2,$$

where the symbol \mathcal{T}_j denotes the translation. From the measurement $\{f_j^{\text{FROG}}\}_j$, one may also formulate it by BPR if assuming the element-wise multiplication for two independent variables.

All the mentioned problems can be unified as the BPR problem, i.e., to recover the probe (pupil, convolution kernel, or the signal itself) and the sample jointly. Essentially the relation between these two variables is bilinear. For conventional ptychography, the bilinear operators $\mathcal{A} : \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \rightarrow \mathbb{C}^m$ and $\mathcal{A}_j : \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \rightarrow \mathbb{C}^m \forall 0 \leq j \leq J - 1$ are denoted as follows:

$$\mathcal{A}(w, u) := (\mathcal{A}_0^T(w, u), \mathcal{A}_1^T(w, u), \dots, \mathcal{A}_{J-1}^T(w, u))^T, \quad (5)$$

with

$$\mathcal{A}_j(w, u) := \mathcal{F}(w \circ \mathcal{S}_j u)$$

and

$$f := (f_0^T, f_1^T, \dots, f_{J-1}^T)^T \in \mathbb{R}_+^m.$$

Actually for all BPR problems, the bilinear operators can be unified as

$$\mathcal{A}_j(w, u) := \begin{cases} \mathcal{F}(w \circ \mathcal{S}_j u); & \text{Case I: CDI and ptychography} \\ \mathcal{F}^{-1}(\mathcal{F}w \circ \mathcal{S}_j(\mathcal{F}u)); & \text{Case II: Fourier ptychography} \\ \mathcal{F}(w \circ \mathcal{T}_j u); & \text{Case III: FROG} \\ w \circledast u, \text{ or } \mathcal{F}(w \circledast u); & \text{Case IV: Convolution PR} \end{cases} \quad (6)$$

where there are totally one frame as $J = 1$ for the last case for convolution PR. Hence, by introducing the general bilinear operator $\mathcal{A}(\cdot, \cdot)$, the BPR can be given below:

$$\text{BPR: To find the "probe" } w \text{ and the sample } u, \text{ s.t. } |\mathcal{A}(w, u)|^2 = f, \quad (7)$$

where \mathcal{A} is denoted as (5) and (6) and the per frame of phaseless measurements f_j represents the measurement from four cases. Note that the BPR problem is not limited to the cases with forward propagation as (6).

Denote two linear operators A_w, A_u as below:

$$\begin{aligned} A_w u &= \mathcal{A}(w, u) \forall u; \\ A_u w &= \mathcal{A}(w, u) \forall w; \end{aligned} \quad (8)$$

Then one can obtain the conjugate operators

$$A_w^* z = \begin{cases} \sum_j \mathcal{S}_j^T (\text{conj}(w) \circ \mathcal{F}^{-1} z_j); & \text{Case I} \\ \mathcal{F}^{-1} \sum_j \mathcal{S}_j^T (\text{conj}(\mathcal{F}w) \circ \mathcal{F}z_j); & \text{Case II} \\ \sum_j \mathcal{T}_j^T (\text{conj}(w) \circ \mathcal{F}^{-1} z_j); & \text{Case III} \\ \text{conj}(w) \otimes z, \text{ or } \text{conj}(w) \otimes \mathcal{F}^{-1} z; & \text{Case IV} \end{cases} \quad (9)$$

and

$$A_u^* z = \begin{cases} \sum_j (\text{conj}(\mathcal{S}_j u) \circ \mathcal{F}^{-1} z_j); & \text{Case I} \\ \mathcal{F}^{-1} \sum_j (\text{conj}(\mathcal{S}_j \mathcal{F}u) \circ \mathcal{F}z_j); & \text{Case II} \\ \sum_j (\text{conj}(\mathcal{T}_j u) \circ \mathcal{F}^{-1} z_j); & \text{Case III} \\ \text{conj}(u) \otimes z, \text{ or } \text{conj}(u) \otimes \mathcal{F}^{-1} z; & \text{Case IV} \end{cases} \quad (10)$$

$\forall z = (z_1^T, z_2^T, \dots, z_{J-1}^T)^T \in \mathbb{C}^m$. Here \sum_j is a simplified form of $\sum_{j=0}^{J-1}$. Consequently, one obtains

$$A_w^* A_w u = \begin{cases} (\sum_j \mathcal{S}_j^T |w|^2) \circ u; & \text{Case I} \\ \mathcal{F}^{-1} ((\sum_j \mathcal{S}_j^T |\mathcal{F}w|^2) \circ \mathcal{F}u); & \text{Case II} \\ (\sum_j \mathcal{T}_j^T |w|^2) \circ u; & \text{Case III} \\ \text{conj}(w) \otimes w \otimes u; & \text{Case IV} \end{cases} \quad (11)$$

and

$$A_u^* A_u w = \begin{cases} \left(\sum_j \mathcal{S}_j |u|^2 \right) \circ w; & \text{Case I} \\ \mathcal{F}^{-1} \left(\left(\sum_j \mathcal{S}_j |\mathcal{F}u|^2 \right) \circ \mathcal{F}w \right); & \text{Case II} \\ \left(\sum_j \mathcal{T}_j |u|^2 \right) \circ w; & \text{Case III} \\ \text{conj}(u) \circledast u \circledast w. & \text{Case IV} \end{cases} \quad (12)$$

Optimization Problems and Proximal Mapping

Solving a nonblind problem may be NP hard if knowing w or u in advance. Other than directly solving equations as (7), one can solve the following nonlinear optimization problems in order to determine the underlying image u and probe w from noisy measurements f :

$$\min_{w,u} \mathcal{M}(|\mathcal{A}(w, u)|^2, f), \quad (13)$$

where the symbol $\mathcal{M}(\cdot, \cdot)$ represents the error between the unknown intensity $|\mathcal{A}(w, u)|^2$ and collected phaseless data f . Various metrics proposed under different noise settings include amplitude-based metric for Gaussian measurements (AGM) (Wen et al. 2012; Chang et al. 2016), intensity-based metric for Poisson measurements (IPM) (Thibault and Guizar-Sicairos 2012; Chen and Candes 2015; Chang et al. 2018b), and intensity-based metric for Gaussian measurements (IGM) (Qian et al. 2014; Candes et al. 2015; Sun et al. 2016), all of which can be expressed as

$$\mathcal{M}(g, f) := \begin{cases} \frac{1}{2} \|\sqrt{g} - \sqrt{f}\|^2; & \text{(AGM)} \\ \frac{1}{2} \langle g - f \circ \log(g), \mathbf{1} \rangle; & \text{(IPM)} \\ \frac{1}{2} \|g - f\|^2; & \text{(IGM)} \end{cases} \quad (14)$$

where the operations on vectors such as $\sqrt{\cdot}$, $\log(\cdot)$, $|\cdot|$, $(\cdot)^2$ are all defined pointwisely in this survey, $\mathbf{1}$ denotes a vector whose entries all equal to ones, and $\|\cdot\|$ denotes the ℓ^2 norm in Euclidean space.

The proximal mapping for functions defined on complex Euclidean space is introduced below.

Definition 1. Given function $h : \mathbb{C}^N \rightarrow \mathbb{R} \cup \{+\infty\}$, the proximal mapping $\text{Prox}_{h;\mu} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ of h is defined by

$$\text{Prox}_{h;\beta}(v) = \arg \min_x \left(h(x) + \frac{\beta}{2} \|x - v\|^2 \right), \quad (15)$$

with the symbol $\|\cdot\|$ denoted as the ℓ^2 norm of a complex vector on complex Euclidean space (use the same notation for real and complex spaces).

Namely, the proximal operator for the function $\mathcal{M}(|\cdot|^2, f)$ defined in (14) has a closed-form formula (Chang et al. 2018c) as below:

$$\text{Prox}_{\mathcal{M}(|\cdot|^2, f); \beta}(z) = \begin{cases} \frac{\sqrt{f} + \beta|z|}{1 + \beta} \circ \text{sign}(z), & \text{for AGM;} \\ \frac{\beta|z| + \sqrt{(\beta|z|)^2 + 4(1 + \beta)f}}{2(1 + \beta)} \circ \text{sign}(z), & \text{for IPM;} \\ \varpi_{\beta}(|z|) \circ \text{sign}(z), & \text{for IGM;} \end{cases} \quad (16)$$

where $\forall z \in \mathbb{C}^m$, $(\text{sign}(z))(t) := \text{sign}(z(t)) \forall 0 \leq t \leq m - 1$, $\text{sign}(x)$ for a scalar $x \in \mathbb{C}$ is denoted as $\text{sign}(x) = \frac{x}{|x|}$ if $x \neq 0$, otherwise $\text{sign}(0) := c$ with an arbitrary constant $c \in \mathbb{C}$ with unity length, and

$$\varpi_{\beta}(|z|)(t) = \begin{cases} \sqrt[3]{\frac{\beta|z(t)|}{4} + \sqrt{D(t)}} + \sqrt[3]{\frac{\beta|z(t)|}{4} - \sqrt{D(t)}}, & \text{if } D(t) \geq 0; \\ 2\sqrt{\frac{f(t) - \frac{\beta}{2}}{3}} \cos\left(\arccos\frac{\theta(t)}{3}\right), & \text{otherwise,} \end{cases} \quad (17)$$

for $0 \leq t \leq m - 1$, with $D(t) = \frac{(\frac{\beta}{2} - f(t))^3}{27} + \frac{\beta^2|z(t)|^2}{16}$, and $\theta(t) = \frac{\beta|z(t)|}{4\sqrt{\frac{(f(t) - \frac{\beta}{2})^3}{27}}}$.

Note that the alternating direction method of multipliers (ADMM) was adopted in Wen et al. (2012) and Chang et al. (2016, 2018b) to solve the variational PR model in (13). However, due to the lack of the globally Lipschitz differentiable terms in the objective function, it seems difficult to guarantee its convergence. Some other variants of the metric have been recently proposed, such as the penalized metrics $\mathcal{M}(|\cdot|^2 + \epsilon \mathbf{1}, f + \epsilon \mathbf{1})$ by adding a small positive scalar ϵ as Guizar-Sicairos and Fienup (2008), Chang et al. (2019a), and Gao et al. (2020). Although it has simple form, the technique will make the related proximal mapping not have closed-form expression, such that additional computation cost as an inner loop may have to be introduced (Chang et al. 2019a). By cutting off the AGM near the origin, and then adding back a smooth function, one can keep the global minimizer unchanged. Hence, a novel smooth truncated AGM (ST-AGM) $\mathcal{G}_{\epsilon}(\cdot; f)$ with truncation parameter $\epsilon > 0$ (Chang et al. 2021) was designed below:

$$\mathcal{M}_\epsilon(z, f) := \sum_j M_\epsilon(z(j), f(j)), \quad (18)$$

where $\forall x \in \mathbb{C}, b \in \mathbb{R}^+$,

$$M_\epsilon(x, b) := \begin{cases} \frac{1-\epsilon}{2} \left(b - \frac{1}{\epsilon}|x|^2 \right), & \text{if } |x| < \epsilon\sqrt{b}; \\ \frac{1}{2}||x| - \sqrt{b}|^2, & \text{otherwise.} \end{cases} \quad (19)$$

Readily its closed form of the corresponding proximal mapping can be found in Chang et al. (2021). More other elaborate metrics can be found in Luke (2005), Cai et al. (2021), and references therein.

Fast Iterative Algorithms

In this section, the main iterative algorithms for BPR will be introduced. Note that each algorithm may be designed originally for a specific case of (6). Hence, the basic idea based on the original case will be explained first, and the possible extensions to other cases will be discussed then.

Alternating Projection (AP) Algorithms

First consider BPR defined in (7) in the case of convention ptychography.

Given the exit wave in the far field $\Psi := (\psi_0^T, \psi_1^T, \dots, \psi_{J-1}^T)^T \in \mathbb{C}^m$, with

$$\psi_j := \mathcal{F}(w \circ S_j u) \quad \forall 0 \leq j \leq J-1,$$

the optimal exit wave Ψ^* lies in the intersection of two following sets, i.e.,

$$\Psi^* \in \widehat{\mathcal{X}}_1 \cap \widehat{\mathcal{X}}_2,$$

with

$$\begin{aligned} \widehat{\mathcal{X}}_1 &:= \{\Psi := (\psi_0^T, \psi_1^T, \dots, \psi_{J-1}^T)^T \in \mathbb{C}^m : |\psi_j| = \sqrt{f_j} \quad \forall 0 \leq j \leq J-1\}, \\ \widehat{\mathcal{X}}_2 &:= \{\Psi \in \mathbb{C}^m : \exists w \in \mathbb{C}^{\bar{m}}, u \in \mathbb{C}^n, \text{ s.t. } w \circ S_j u = \mathcal{F}^{-1} \psi_j \quad \forall 0 \leq j \leq J-1\}. \end{aligned} \quad (20)$$

The AP algorithm determining this intersection alternatively calculates the projections onto these two sets $\widehat{\mathcal{X}}_1$ and $\widehat{\mathcal{X}}_2$. Regarding the projection onto $\widehat{\mathcal{X}}_1$ as

$$\widehat{\mathcal{P}}_1(\Psi) := \arg \min_{\widehat{\Psi} \in \widehat{\mathcal{X}}_1} \|\widehat{\Psi} - \Psi\|^2,$$

one readily gets a closed-form solution

$$\widehat{\mathcal{P}}_1(\Psi) := ((\widehat{\mathcal{P}}_1^0(\Psi))^T, \dots, (\widehat{\mathcal{P}}_1^{J-1}(\Psi))^T)^T,$$

with

$$\widehat{\mathcal{P}}_1^j(\Psi) = \sqrt{f_j} \circ \text{sign}(\Psi_j) \quad 0 \leq j \leq J-1.$$

For the projection onto $\widehat{\mathcal{X}}_2$, given Ψ^k as the solution in the k th iteration, one gets

$$\begin{aligned} \widehat{\mathcal{P}}_2(\Psi^k) := & ((\mathcal{F}(w^{k+1} \circ \mathcal{S}_0 u^{k+1}))^T, (\mathcal{F}(w^{k+1} \circ \mathcal{S}_1 u^{k+1}))^T, \dots, \\ & (\mathcal{F}(w^{k+1} \circ \mathcal{S}_{J-1} u^{k+1}))^T)^T, \end{aligned}$$

where

$$(w^{k+1}, u^{k+1}) = \arg \min_{w, u} F(w, u, \Psi^k) := \frac{1}{2} \sum_j \|\mathcal{F}^{-1} \Psi_j^k - w \circ \mathcal{S}_j u\|^2. \quad (21)$$

Unfortunately, it does not have a closed-form solution. One can solve (21) by alternating minimization (with T steps) as below:

$$\begin{aligned} w_{l+1} &= \arg \min_w F(w, u_l, \Psi^k), \\ u_{l+1} &= \arg \min_u F(w_{l+1}, u, \Psi^k) \quad \forall l = 0, 1, \dots, T-1. \end{aligned} \quad (22)$$

Readily one has

$$\begin{aligned} w_{l+1} &\approx \frac{\sum_j \text{conj}(\mathcal{S}_j u_l) \circ \mathcal{F}^{-1} \Psi_j^k}{\sum_j |\mathcal{S}_j u_l|^2 + \bar{\alpha}_1}; \\ u_{l+1} &\approx \frac{\sum_j \mathcal{S}_j^T (\text{conj}(w_{l+1}) \circ \mathcal{F}^{-1} \Psi_j^k)}{\sum_j (\mathcal{S}_j^T |w_{l+1}|^2) + \bar{\alpha}_2} \quad \forall l = 0, 1, \dots, T-1, \end{aligned} \quad (23)$$

where the parameters $0 < \bar{\alpha}_1, \bar{\alpha}_2 \ll 1$ are introduced in order to avoid dividing by zeros.

Letting Ψ^k be iterative solution in the k th iteration, the standard AP for BPR can be directly given as below:

- (1) Compute $\widehat{\Psi}^k$ by $\widehat{\Psi}_j^k = \mathcal{F}(w^{k+1} \circ \mathcal{S}_j u^{k+1})$, where the pair (w^{k+1}, u^{k+1}) is approximately solved by (23).
- (2) Compute Ψ^{k+1} by $\Psi^{k+1} = \widehat{\mathcal{P}}_1(\widehat{\Psi}^k)$.

The DR algorithm for BPR can be formulated in two steps (Thibault et al. 2009), as follows:

- (1) Compute $\widehat{\Psi}^k$ as the first step of AP.
- (2) Compute Ψ^{k+1} by

$$\Psi^{k+1} = \Psi^k + \widehat{\mathcal{P}}_1(2\widehat{\Psi}^k - \Psi^k) - \widehat{\Psi}^k. \quad (24)$$

Note that the formula (24) utilizing DR operator is essentially Fienup's hybrid input-output map, which can also be derived with proper parameters from difference map (Elser 2003).

Since the fixed point of DR iteration may not exist, Marchesini et al. (2016) adopted the relaxed version of DR (dubbed as RAAR by Luke 2005) to further improve the stability of the reconstruction from noisy measurements, which simply takes weighted average of right term of (24) and $\widehat{\Psi}^k$ with a tunable parameter $\delta \in (0, 1)$ as

$$\Psi^{k+1} = \delta(\Psi^k + \widehat{\mathcal{P}}_1(2\widehat{\Psi}^k - \Psi^k) - \widehat{\Psi}^k) + (1 - \delta)\widehat{\Psi}^k,$$

with $\widehat{\Psi}^k$ determined in a same manner as the first step of AP.

At the end of this part, extension of AP to general BPR problems will be discussed. Similarly as for the ptychography, introduce Ψ as

$$\Psi = \mathcal{A}(w, u), \text{ and } \Psi_j = \mathcal{A}_j(w, u).$$

In the same manner, one can define two constraint sets and establish the AP algorithms for the four cases of BPR. The only differences lie in the calculations of the projections onto the bilinear constraint set. As (21), consider

$$\min_{w, u} \|\Psi^k - \mathcal{A}(w, u)\| \quad (25)$$

by alternating minimization, where Ψ^k is the iterative solution. Then the scheme is given below:

$$\begin{aligned} w_{l+1} &\approx (A_{u_l}^* A_{u_l} + \bar{\alpha}_1 \mathbf{I})^{-1} A_{u_l}^* \Psi^k, \\ u_{l+1} &\approx (A_{w_{l+1}}^* A_{w_{l+1}} + \bar{\alpha}_2 \mathbf{I})^{-1} A_{w_{l+1}}^* \Psi^k \quad \forall l = 0, 1, \dots, T-1. \end{aligned} \quad (26)$$

The detailed forms of these operators can be found in (9), (10), (11), and (12). Notably the inverse in (26) can be efficiently solved by pointwise division or DFT.

ePIE-Type Algorithms

This iterative algorithm can be expressed as an AP method for convention ptychography as follows: To find $\Psi_{n_k}^*$ belonging to the intersection as

$$\Psi_{n_k}^* \in \{|\Psi_{n_k}| = \sqrt{f_{n_k}}\} \cap \{\Psi_{n_k} : \exists w \in \mathbb{C}^m, u \in \mathbb{C}^n, \text{ s.t. } w \circ \mathcal{S}_{n_k} u = \mathcal{F}^{-1} \Psi_{n_k}\},$$

with a random frame index n_k . Let w^k, u^k be the iterative solutions in the k^{th} iteration. By first computing the projection of $\psi_{n_k}^k := \mathcal{F}(w^k \circ \mathcal{S}_{n_k} u^k)$ by $\widehat{\mathcal{P}}_1^{n_k}(\psi_{n_k}^k)$, and then updating w^{k+1} and u^{k+1} by the gradient descent algorithm (inexact projection) for (21), the ePIE algorithm proposed by Maiden and Rodenburg (2009) can be expressed by updating w^{k+1} and u^{k+1} in parallel as

$$\begin{cases} w^{k+1} = w^k - \frac{d_2}{\|\mathcal{S}_{n_k} u^k\|_\infty^2} \mathcal{S}_{n_k} \text{conj}(u^k) \circ \mathcal{F}^{-1}(\Psi_{n_k}^k - \widehat{\mathcal{P}}_1^{n_k}(\Psi_{n_k}^k)) \\ u^{k+1} = u^k - \frac{d_1}{\|\mathcal{S}_{n_k}^T w^k\|_\infty^2} \mathcal{S}_{n_k}^T \left(\text{conj}(w^k) \circ \mathcal{F}^{-1}(\Psi_{n_k}^k - \widehat{\mathcal{P}}_1^{n_k}(\Psi_{n_k}^k)) \right), \end{cases} \quad (27)$$

with frame index $n_k \in \{0, 1, \dots, J-1\}$ generated randomly and positive parameters d_1 and d_2 (default values are ones) and $\|w\|_\infty := \max_t |w(t)|$.

The regularized PIE (rPIE) was further proposed by Maiden et al. (2017) as

$$\begin{cases} w^{k+1} = w^k - \frac{\mathbf{1}}{\delta \|\mathcal{S}_{n_k} u^k\|_\infty^2 + (1-\delta) \mathcal{S}_{n_k} |u^k|^2} \\ \quad \circ \mathcal{S}_{n_k} \text{conj}(u^k) \circ \mathcal{F}^{-1}(\Psi_{n_k}^k - \widehat{\mathcal{P}}_1^{n_k}(\Psi_{n_k}^k)), \\ u^{k+1} = u^k - \frac{\mathbf{1}}{\delta \|\mathcal{S}_{n_k}^T w^k\|_\infty^2 + (1-\delta) \mathcal{S}_{n_k}^T |w^k|^2} \\ \quad \circ \mathcal{S}_{n_k}^T \left(\text{conj}(w^k) \circ \mathcal{F}^{-1}(\Psi_{n_k}^k - \widehat{\mathcal{P}}_1^{n_k}(\Psi_{n_k}^k)) \right), \end{cases} \quad (28)$$

with the scalar constant $\delta \in (0, 1)$. It can be interpreted as a hybrid scheme for the stepsize of gradient descent, which takes the weighted average of the denominator of the ePIE scheme (27) and first term in the denominator of AP scheme (23). The rPIE algorithm was further accelerated by momentum (Maiden et al. 2017).

One can directly get the ePIE and rPIE schemes for FP (Zheng et al. 2021) by replacing the variables w and u by $\mathcal{F}w$ and $\mathcal{F}u$. The ePIE-type algorithms are very popular in optics community, since it is enough to implement the algorithm if one knows how to calculate the gradient of the objective functions, and the memory footprint is much smaller than more advanced AP algorithm including DR and RAAR. However, it tends to unstable when the data redundancy is insufficient (e.g., noisy data, big-step scan) as reported in Chang et al. (2019a). Moreover, the theoretical convergence is unknown and seems challenging due to the relation with nonsmooth objective functions.

Note that if with totally $J = 1$ frame as CDI, the differences between the ePIE (with $d_1 = d_2 = 1$) and standard AP lie in the preconditioning matrices: AP utilizes the spatial weighted diagonal matrices $A_u^* A_u$ and $A_w^* A_w$, while ePIE utilizes

the spatial-independent constant determined by the maximum of their diagonal matrices.

Proximal Algorithms

Proximal Heterogeneous Block Implicit-Explicit (PHeBIE) For convention ptychography, consider an optimization problem (to get rid of introducing redundant notations in this survey, slightly modify the constraint set of Ψ in Hesse et al. (2015) as $\widehat{\mathcal{X}}_1$ and adjust the notation of the first term of the following model accordingly in order to present an equivalent form) (Hesse et al. 2015) as follows:

$$\min_{w,u,\Psi} F(w, u, \Psi) + \mathbb{I}_{\widehat{\mathcal{X}}_1}(\Psi) + \mathbb{I}_{\mathcal{X}_1}(w) + \mathbb{I}_{\mathcal{X}_2}(u), \quad (29)$$

with $F(w, u, \Psi)$ and $\widehat{\mathcal{X}}_1$ denoted in (21) and (20), respectively, and the indicator function $\mathbb{I}_{\mathcal{X}}$ denoted as

$$\mathbb{I}_{\mathcal{X}}(\Psi) := \begin{cases} 0, & \text{if } \Psi \in \mathcal{X}, \\ +\infty, & \text{otherwise,} \end{cases}$$

where the amplitude constraints of the probe and image are incorporated (in Hesse et al. (2015), the authors further considered the compact support condition of the probes and image), where

$$\begin{aligned} \mathcal{X}_1 &:= \{w \in \mathbb{C}^m : \|w\|_\infty \leq C_w\}; \\ \mathcal{X}_2 &:= \{u \in \mathbb{C}^n : \|u\|_\infty \leq C_u\} \end{aligned} \quad (30)$$

with two positive constants C_w, C_u . The projection operator onto \mathcal{X}_1 is readily obtained as

$$\text{Proj}(w; C_w) := \min\{C_w, |w|\} \circ \text{sign}(w) \quad \forall w,$$

which is the closed-form expression for the minimizer to the problem

$$\min_{\|\tilde{w}\|_\infty \leq C_w} \frac{1}{2} \|\tilde{w} - w\|^2.$$

Similarly, one gets the projection onto \mathcal{X}_2 as $\text{Proj}(u; C_u)$.

Hesse et al. (2015) further adopted the proximal alternating linearized minimization (PALM) method (Bolte et al. 2014) for the BPR problem in the case of convention ptychography, such that the proximal heterogeneous block implicit-explicit (PHeBIE) (see Hesse et al. 2015, Algorithm 2.1) consists of two steps with three positive parameters d_1, d_2 , and γ :

(1) Calculate w^{k+1}, u^{k+1} sequentially as

$$\begin{cases} w^{k+1} = \text{Proj}\left(w^k - \frac{1}{d_1^k} \sum_j \mathcal{S}_j \text{conj}(u^k) \circ \mathcal{F}^{-1}(w^k \circ \mathcal{S}_j u^k - \Psi_j^k); C_w\right); \\ u^{k+1} = \text{Proj}\left(u^k - \frac{1}{d_2^k} \sum_j \mathcal{S}_j^T (\text{conj}(w^{k+1}) \circ \mathcal{F}^{-1}(w^{k+1} \circ \mathcal{S}_j u^k - \Psi_j^k)), C_u\right), \end{cases} \quad (31)$$

with $d_1^k := d_1 \|\sum_j |\mathcal{S}_j u^k|\|_\infty^2$, $d_2^k := d_2 \|\sum_j \mathcal{S}_j^T |w^{k+1}|\|_\infty^2$.

(2) Calculate Ψ^{k+1} by

$$\Psi^{k+1} = \widehat{\mathcal{P}}_1\left(\frac{1}{1+\gamma}(\widehat{\Psi}^{k+1} + \gamma\Psi^k)\right),$$

with

$$\widehat{\Psi}_j^{k+1} := \mathcal{F}(w^{k+1} \circ \mathcal{S}_j u^{k+1}).$$

Under the assumption of boundedness of iterative sequences, the convergence of PALM to stationary points of (29) was proved (Hesse et al. 2015). To the knowledge, it is the first iterative algorithm with convergence guarantee for BPR problem. The PHeBIE has multiple similarities with ePIE. The main differences between them are that, for ePIE, only the gradient of F w.r.t. a randomly selected single frame is adopted to update w and u per outer loop as (27), while, for PHeBIE, each block of w and u can be updated in parallel by employing the gradient as (31) w.r.t. all adjacent frames. Therefore, PHeBIE is more stable than ePIE numerically.

One readily knows that the convergence rate relies on the Lipschitz constant of partial derivative of F . In order to get smaller constant, a direct way is to employ the derivative of a small block for unknowns. Hence, based on the partition of the sample and the probe, the parallel version of PHeBIE was also provided (Hesse et al. 2015) with convergence guarantee.

For more extensions to other cases of BPR, one can introduce a generalized nonlinear optimization model:

$$\min_{w, u, \Psi} \|\Psi - \mathcal{A}(w, u)\|^2 + \mathbb{I}_{\widehat{\mathcal{X}}_1}(\Psi) + \mathbb{I}_{\mathcal{X}_1}(w) + \mathbb{I}_{\mathcal{X}_2}(u),$$

where one adopts the same form as (25) for the first term. The detailed algorithms are omitted, since one only needs to update the gradient of first term following (9), (10), (11), and (12).

Variant of Proximal Algorithm Here introduce a general constraint set for the bilinear relation as

$$X := \{\Psi \in \mathbb{C}^m : \exists w \in \mathbb{C}^{\bar{m}}, u \in \mathbb{C}^n, s.t. \mathcal{A}(w, u) = \Psi\}. \quad (32)$$

Consider the optimization problem as

$$\min_z \mathbb{I}_{\widehat{\mathcal{X}}_1}(z) + \mathbb{I}_X(z). \quad (33)$$

By replacing the indicator function by the metrics, and further combining the alternating minimization with proximal algorithms, Han Yan (2020) derived a new proximal algorithm for the convention ptychography problem. Specifically, the proposed algorithm with a generalized form for BPR has the following steps:

$$\begin{aligned} \text{Step 1: } z^{k+1} &= \arg \min_z \mathcal{M}(|z|^2; f) + \frac{\beta}{2} \|z - \mathcal{A}(w^k, u^k)\|^2; \\ \text{Step 2: } w^{k+1} &= \arg \min_w \|z^{k+1} - A_{u^k} w\|^2. \\ \text{Step 3: } u^{k+1} &= \arg \min_u \|z^{k+1} - A_{w^{k+1}} u\|^2. \end{aligned} \quad (34)$$

Here the last two steps can be solved in a same manner as (26). The above algorithm has deep connections with the ADMM (Chang et al. 2019a). If removing the constraint of boundedness of two variables, and setting the penalization parameter to zero in (35), then by solving the constraint problem (37) by adding a penalization term $\|z - \mathcal{A}(w, u)\|^2$ without introducing the multiplier Λ , one can get exactly the same iterative scheme as (34). Besides, it was further improved by accelerated proximal gradient method in Yan (2020) and recently by stochastic gradient descent (Huang et al. 2021) for FP.

ADMM

As a typical operator-splitting algorithm, ADMM is very flexible and successfully applied to inverse and imaging problems (Wu and Tai 2010; Boyd et al. 2011), which is also adopted for classical and ptychographic PR problems (Wen et al. 2012; Chang et al. 2019a).

Consider the metrics using penalized-AGM (pAGM) and penalized-IPM (pIPM) as to measure the error of recovered intensity and the targets. A nonlinear optimization model (Chang et al. 2019a) was given as

$$\min_{w \in \mathbb{C}^{\bar{m}}, u \in \mathbb{C}^n} \mathcal{G}(\mathcal{A}(w, u)) + \mathbb{I}_{\mathcal{X}_1}(w) + \mathbb{I}_{\mathcal{X}_2}(u), \quad (35)$$

with $\mathcal{G}(z) := \mathcal{M}(|z|^2 + \epsilon \mathbf{1}, f + \epsilon \mathbf{1})$ and the constraint sets defined in (30). The authors further leveraged the additional data $\mathbf{c} \in \mathbb{R}_+^{\bar{m}}$ to eliminate structural artifacts caused by grid scan and therefore obtained the following variant:

$$\min_{w \in \mathbb{C}^{\bar{m}}, u \in \mathbb{C}^n} \mathcal{G}(\mathcal{A}(w, u)) + \mathbb{I}_{\mathcal{X}_1}(w) + \mathbb{I}_{\mathcal{X}_2}(u) + \tau \widehat{\mathcal{G}}(\mathcal{F}w), \quad (36)$$

where τ is a positive parameter, the additional measurement \mathbf{c} is the diffraction pattern (absolute value of Fourier transform of the probe) as $\mathbf{c} := |\mathcal{F}u|$, and $\widehat{\mathcal{G}}(z) := \mathcal{B}(|z|^2 + \epsilon \mathbf{1}, \mathbf{c}^2 + \epsilon \mathbf{1})$. For simplicity, assume that \mathcal{G} and $\widehat{\mathcal{G}}$ adopt the same metric.

As the procedures for solving the two above models are quite similar using ADMM, only details for solving the first optimization model (35) are given below. By introducing an auxiliary variable $z = \mathcal{A}(w, u) \in \mathbb{C}^m$, an equivalent form of (35) is formulated below:

$$\min_{w, u, z} \mathcal{G}(z) + \mathbb{I}_{\mathcal{D}_1}(w) + \mathbb{I}_{\mathcal{D}_2}(u), \quad s.t. \quad z - \mathcal{A}(w, u) = 0. \quad (37)$$

The corresponding augmented Lagrangian reads

$$\begin{aligned} \Upsilon_\beta(w, u, z, \Lambda) := & \mathcal{G}(z) + \mathbb{I}_{\mathcal{D}_1}(w) + \mathbb{I}_{\mathcal{D}_2}(u) + \Re(\langle z - \mathcal{A}(w, u), \Lambda \rangle) \\ & + \frac{\beta}{2} \|z - \mathcal{A}(w, u)\|^2, \end{aligned} \quad (38)$$

with the multiplier $\Lambda \in \mathbb{C}^m$ and a positive parameter β , where $\Re(\cdot)$ denotes the real part of a complex number. Then one considers the following problem:

$$\max_{\Lambda} \min_{w, u, z} \Upsilon_\beta(w, u, z, \Lambda). \quad (39)$$

Given the approximated solution $(w^k, u^k, z^k, \Lambda^k)$ in the k th iteration, the four-step iteration by the generalized ADMM (only the subproblems *w.r.t.* w or u have proximal terms) is given as follows:

$$\left\{ \begin{array}{l} \text{Step 1: } w^{k+1} = \arg \min_w \Upsilon_\beta(w, u^k, z^k, \Lambda^k) + \frac{\alpha_1}{2} \|w - w^k\|_{M_1^k}^2 \\ \text{Step 2: } u^{k+1} = \arg \min_u \Upsilon_\beta(w^{k+1}, u, z^k, \Lambda^k) + \frac{\alpha_2}{2} \|u - u^k\|_{M_2^k}^2 \\ \text{Step 3: } z^{k+1} = \arg \min_z \Upsilon_\beta(w^{k+1}, u^{k+1}, z, \Lambda^k) \\ \text{Step 4: } \Lambda^{k+1} = \Lambda^k + \beta(z^{k+1} - \mathcal{A}(w^{k+1}, u^{k+1})), \end{array} \right. \quad (40)$$

with diagonal positive semidefinite matrices $M_1^k \in \mathbb{R}_+^{\bar{m} \times \bar{m}}$ and $M_2^k \in \mathbb{R}_+^{n \times n}$ and two penalization parameters $\alpha_1, \alpha_2 > 0$, where $\|w\|_{M_1^k}^2 := \langle M_1^k w, w \rangle$ and $\|u\|_{M_2^k}^2 := \langle M_2^k u, u \rangle$.

Detailed algorithms will be given focusing on convention ptychography as Chang et al. (2019a). Note that these two matrices M_1^k, M_2^k are assumed to be diagonal so that subproblems in Step 1 and Step 2 have closed-form solutions. Roughly speaking, based on splitting technique of proximal ADMM, subproblems of u, w , and z are element-wise optimization problems with closed-form solutions, such that each subproblem can be fast solved. In practice, these two matrices are chosen by hand, and an adaptive strategy was presented in Chang et al. (2019a) in order to guarantee the convergence. Letting

$$\hat{z}^k := z^k + \frac{\Lambda^k}{\beta}$$

and the diagonal matrices M_1^k and M_2^k satisfy

$$\begin{cases} \min_t \sum_j \left| (\mathcal{S}_j u^k)(t) \right|^2 + \frac{\alpha_1}{\beta} \text{diag}(M_1^k)(t) > 0, \\ \min_t \sum_j \left| (\mathcal{S}_j^T w^{k+1})(t) \right|^2 + \frac{\alpha_2}{\beta} \text{diag}(M_2^k)(t) > 0, \end{cases} \quad (41)$$

the closed-form solutions of Step 1 and Step 2 are given as

$$\begin{cases} w^{k+1} = \text{Proj} \left(\frac{\beta \sum_j \text{conj}(\mathcal{S}_j u^k) \circ (\mathcal{F}^{-1} z_j^k) + \alpha_1 \text{diag}(M_1^k) \circ w^k}{\beta \sum_j |\mathcal{S}_j u^k|^2 + \alpha_1 \text{diag}(M_1^k)}; C_w \right); \\ u^{k+1} = \text{Proj} \left(\frac{\beta \sum_j \mathcal{S}_j^T (\text{conj}(w^{k+1}) \circ \mathcal{F}^{-1} z_j^k) + \alpha_2 \text{diag}(M_2^k) \circ u^k}{\beta \sum_j (\mathcal{S}_j^T |w^{k+1}|^2) + \alpha_2 \text{diag}(M_2^k)}; C_u \right). \end{cases} \quad (42)$$

For Step 3, denoting

$$z^+ = \mathcal{A}(w^{k+1}, u^{k+1}) - \frac{\Lambda^k}{\beta},$$

one has

$$z^{k+1} = \arg \min_z \frac{1}{2} (|z|^2 + \varepsilon \mathbf{1}_m - (f + \varepsilon \mathbf{1}_m) \circ \log(|z|^2 + \varepsilon \mathbf{1}_m), \mathbf{1}_m) + \frac{\beta}{2} \|z - z^+\|^2.$$

The solution can be expressed as

$$z^{k+1} = \rho^* \circ \text{sign}(z^+), \quad (43)$$

where $\rho^*(t)$ was solved by the gradient projection scheme expressed as

$$x_{l+1} = \max \left\{ 0, x_l - \delta \left((1 + \beta - \frac{f(t) + \varepsilon}{|x_l|^2 + \varepsilon}) x_l - \beta z^+(t) \right) \right\}, \forall l = 0, 1, \dots, \quad (44)$$

if using the pIPM, or

$$x_{l+1} = \max \left\{ 0, x_l - \delta \left((1 + \beta - \frac{\sqrt{f(t) + \varepsilon}}{\sqrt{|x_l|^2 + \varepsilon}}) x_l - \beta z^+(t) \right) \right\}, \forall l = 0, 1, \dots, \quad (45)$$

if using the pAGM with the stepsize $\delta > 0$, and $x_0 := |z^k(t)|$. Note that with the penalization parameter $\varepsilon = 0$, one can directly get the closed-form solution by (16) as Wen et al. (2012) and Chang et al. (2018b).

Under the condition of sufficient overlapping scan, and bounded preconditioning matrices M_1^k and M_2^k , the convergence of the ADMM can be derived on the sense that the iterative sequence generated by above algorithm converges to a stationary point of the augmented Lagrangian by letting the parameter β sufficiently large.

From the point of view of fixed point analysis, for nonblind problems (knowing the probe w), the authors Fannjiang and Zhang (2020) presented a variant ADMM to solve the following optimization problem:

$$\min_z \mathcal{M}(|z|^2; f) + \mathbb{I}_X(z), \quad (46)$$

with $X \subset \mathbb{C}^m$ defined in (32). By introducing the auxiliary variable $\bar{z} = z$ and decomposing the objective functions, the ADMM was proposed in Fannjiang and Zhang (2020) to solve

$$\min_{z, \bar{z}} \mathcal{M}(|z|^2; f) + \mathbb{I}_X(\bar{z}), \quad s.t. \quad z - \bar{z} = 0. \quad (47)$$

To further apply the idea to the BPR, alternating minimization was further adopted as

$$\begin{aligned} z^{k+1/2} &:= \arg \min_z \mathcal{M}(|z|^2; f) + \mathbb{I}_{X_1^k}(z); \\ z^{k+1} &:= \arg \min_z \mathcal{M}(|z|^2; f) + \mathbb{I}_{X_2^k}(z); \end{aligned} \quad (48)$$

where these two subproblems can be solved via ADMM as inner loop. Here one has to adjust the constraint sets with the update probe and sample, i.e.,

$$\begin{aligned} X_1^k &:= \{z : z = \mathcal{A}(w^k, u) \forall u \in \mathbb{C}^n\}, \\ X_2^k &:= \{z : z = \mathcal{A}(w, u^{k+1}) \forall w \in \mathbb{C}^{\bar{m}}\}. \end{aligned}$$

Note that the probe and sample can be readily determined by solving the least squares problem as

$$\begin{aligned} u^{k+1} &= \arg \min_u \|z^{k+1/2} - \mathcal{A}(w^k, u)\|^2, \\ w^{k+1} &= \arg \min_w \|z^{k+1} - \mathcal{A}(w, u^{k+1})\|^2, \end{aligned} \quad (49)$$

which can be solved by (23). Although the algorithms worked well with suitable initialization as reported in Fannjiang and Zhang (2020), the theoretical convergence for the blind recovery is still open.

Convex Programming

Ahmed et al. (2018) proposed a convex relaxation based on a lifted matrix recovery formulation that allows a nontrivial convex relaxation of the convolution PR.

Consider the convolution PR as

$$f^{\text{Cov}} = |\mathcal{F}\kappa \circ \mathcal{F}u|^2.$$

One basic assumption for unique recovery is that the variables κ and u belong to the subspace of \mathbb{C}^n , i.e.,

$$\kappa = \mathbf{B}\mathbf{h}, \quad u = \mathbf{C}\mathbf{m},$$

where $\mathbf{h} \in \mathbb{C}^{k_1}$ and $\mathbf{m} \in \mathbb{C}^{k_2}$ with known matrices $\mathbf{B} \in \mathbb{C}^{n,k_1}$ and $\mathbf{C} \in \mathbb{C}^{n,k_2}$ ($k_1, k_2 \ll n$). Then one is concerned with the following problem with \mathbf{h}, \mathbf{m} as unknowns:

$$f^{\text{Cov}} = \frac{1}{\sqrt{n}} |\hat{\mathbf{B}}\mathbf{h} \circ \hat{\mathbf{C}}\mathbf{m}|^2 \quad (50)$$

with $\hat{\mathbf{B}} := \sqrt{n}\mathcal{F}\mathbf{B}$, $\hat{\mathbf{C}} := \sqrt{n}\mathcal{F}\mathbf{C}$. Further by the lifting technique in semidefinite programming (SDP), the above problem reduces to

$$f^{\text{Cov}}(l) = \frac{1}{n} \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle, \quad (51)$$

where $\mathbf{H} := \mathbf{h}\mathbf{h}^*$, $\mathbf{M} := \mathbf{m}\mathbf{m}^*$ (rank 1 matrices), and $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product (trace of multiplication of two matrices). Here \mathbf{b}_l^* and \mathbf{c}_l^* are the rows of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$, respectively. By using a nuclear-norm minimization, to convexify the rank of matrix and further transform (51) to a convex constraint, then the following convex optimization model can be derived as

$$\begin{aligned} \min_{\mathbf{H} \succeq \mathbf{0}, \mathbf{M} \succeq \mathbf{0}} \quad & \text{Tr}(\mathbf{H}) + \text{Tr}(\mathbf{M}) \\ \text{s.t.} \quad & \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle \geq \bar{f}(l), \quad 0 \leq l \leq n-1, \end{aligned} \quad (52)$$

with $\bar{f} := n f^{\text{Cov}}$.

An ADMM scheme was further developed (Ahmed et al. 2018) to solve (52). By introducing the convex constraint set

$$\mathcal{C} := \{(\mathbf{v}_1, \mathbf{v}_2) : \mathbf{v}_1(l)\mathbf{v}_2(l) \geq \bar{f}(l), \mathbf{v}_1(l) \geq 0, \mathbf{v}_2(l) \geq 0 \forall 0 \leq l \leq n-1\}$$

and $\mathbf{H}' = \mathbf{H}$, $\mathbf{M}' = \mathbf{M}$, an equivalent form can be given as

$$\begin{aligned} \min_{\mathbf{H}, \mathbf{H}', \mathbf{M}, \mathbf{M}', \mathbf{v}_1, \mathbf{v}_2} \quad & \mathbb{I}_{\mathcal{C}}(\mathbf{v}_1, \mathbf{v}_2) + \text{Tr}(\mathbf{H}) + \text{Tr}(\mathbf{M}) \\ & + \mathbb{I}_{\{X \succeq \mathbf{0}\}}(\mathbf{H}') + \mathbb{I}_{\{X \succeq \mathbf{0}\}}(\mathbf{M}'), \\ \text{s.t.} \quad & \mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle = 0, \quad \mathbf{v}_2(l) - \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle = 0, \\ & \mathbf{H}' - \mathbf{H} = 0, \quad \mathbf{M}' - \mathbf{M} = 0. \end{aligned}$$

With the multipliers Λ_k for $k = 1, 2, 3, 4$ for the totally four constraints, the augmented Lagrangian with scalar form has the following form:

$$\begin{aligned}
& \mathcal{L}_c(\mathbf{H}, \mathbf{H}', \mathbf{M}, \mathbf{M}', \mathbf{v}_1, \mathbf{v}_2; \{\Lambda_k\}_{k=1}^4) \\
& := \mathbb{I}_{\mathcal{C}}(\mathbf{v}_1, \mathbf{v}_2) + \text{Tr}(\mathbf{H}) + \text{Tr}(\mathbf{M}) + \mathbb{I}_{\{X \succeq \mathbf{0}\}}(\mathbf{H}') + \mathbb{I}_{\{X \succeq \mathbf{0}\}}(\mathbf{M}') \\
& \quad + \beta_1 \sum_l \left(\langle \Lambda_1(l), \mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle \rangle + \frac{1}{2} \|\mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle\|^2 \right) \\
& \quad + \beta_1 \sum_l \left(\langle \Lambda_2(l), \mathbf{v}_2(l) - \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle \rangle + \frac{1}{2} \|\mathbf{v}_2(l) - \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle\|^2 \right) \\
& \quad + \beta_2 \langle \Lambda_3, \mathbf{H}' - \mathbf{H} \rangle + \frac{\beta_2}{2} \|\mathbf{H}' - \mathbf{H}\|^2 \\
& \quad + \beta_2 \langle \Lambda_4, \mathbf{M}' - \mathbf{M} \rangle + \frac{\beta_2}{2} \|\mathbf{M}' - \mathbf{M}\|^2, \tag{53}
\end{aligned}$$

with two positive scalar parameters β_1, β_2 . Then with alternating minimization and update of dual variables Λ_k , the iterative scheme is obtained. First, one can optimize the variables \mathbf{H} and \mathbf{M} in parallel and only consider

$$\begin{aligned}
\mathbf{H}^* & := \arg \min_{\mathbf{H}} \text{Tr}(\mathbf{H}) + \beta_1 \sum_l \langle \Lambda_1(l), \mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle \rangle \\
& \quad + \frac{\beta_1}{2} \|\mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle\|^2 + \beta_2 \langle \Lambda_3, \mathbf{H}' - \mathbf{H} \rangle + \frac{\beta_2}{2} \|\mathbf{H}' - \mathbf{H}\|^2.
\end{aligned}$$

By considering the first-order optimality condition (taking the derivative of the objective function w.r.t. \mathbf{H}), one obtains

$$\text{vec}(\mathbf{H}^*) = \mathbf{T}_1^{-1} \text{vec} \left(\beta_1 \sum_l (\mathbf{v}_1(l) + \Lambda_1(l)) \mathbf{b}_l \mathbf{b}_l^* + \beta_2 (\mathbf{H}' - \Lambda_3) - \mathbf{I} \right),$$

with

$$\mathbf{T}_1 := \beta_1 \sum_l \text{vec}(\mathbf{b}_l \mathbf{b}_l^*) \text{vec}(\mathbf{b}_l \mathbf{b}_l^*)^* + \beta_2 \mathbf{I}.$$

Similarly, one can determine the optimal \mathbf{M}^* for the subproblem w.r.t. \mathbf{M} by

$$\text{vec}(\mathbf{M}^*) = \mathbf{T}_2^{-1} \text{vec} \left(\beta_1 \sum_l (\mathbf{v}_2(l) + \Lambda_2(l)) \mathbf{c}_l \mathbf{c}_l^* + \beta_2 (\mathbf{M}' - \Lambda_4) - \mathbf{I} \right),$$

with

$$\mathbf{T}_2 := \beta_1 \sum_l \text{vec}(\mathbf{c}_l \mathbf{c}_l^*) \text{vec}(\mathbf{c}_l \mathbf{c}_l^*)^* + \beta_2 \mathbf{I}.$$

For the \mathbf{H}' -subproblem, denoting $\tilde{\mathbf{H}} := \mathbf{H} - \Lambda_3$, one considers the problem

$$\mathbf{H}'^* := \arg \min_{\mathbf{H}'} \mathbb{I}_{\{X \succ \mathbf{0}\}}(\mathbf{H}') + \frac{1}{2} \|\mathbf{H}' - \tilde{\mathbf{H}}\|^2, \quad (54)$$

with the Hermitian matrix $\tilde{\mathbf{H}}$ (if initializing the multipliers Λ_3 and Λ_4 with Hermitian matrices, it can be readily guaranteed that all iterative sequences of these two multipliers are Hermitian). The closed-form solution of (54) can be directly given as

$$\mathbf{H}'^* = \text{Proj}_+(\tilde{\mathbf{H}}),$$

with the operator Proj_+ defined as

$$\text{Proj}_+(\tilde{\mathbf{H}}) := \mathbf{U} \text{diag}(\max\{\text{diag}(\boldsymbol{\Sigma}), 0\}) \mathbf{U}^*$$

and $\tilde{\mathbf{H}}$ has the eigen-decomposition as $\tilde{\mathbf{H}} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{U}^*$ with unitary matrix \mathbf{U} and diagonal matrix $\boldsymbol{\Sigma}$.

Similarly,

$$\mathbf{M}'^* := \arg \min_{\mathbf{M}'} \mathbb{I}_{\{X \succ \mathbf{0}\}}(\mathbf{M}') + \frac{1}{2} \|\mathbf{M}' - (\mathbf{M} - \Lambda_3)\|^2. \quad (55)$$

One can directly get the closed-form solution

$$\mathbf{M}'^* = \text{Proj}_+(\mathbf{M} - \Lambda_3).$$

The subproblems w.r.t the variables $\mathbf{v}_1, \mathbf{v}_2$ can be solved in an element-wise manner, due to the independence of the optimization problem for each element of these two variables. Since they can be derived with standard discussion based on Karush-Kuhn-Tucker optimality conditions, the details here are omitted. Hence, all procedures to get the iterative scheme are summarized by further combining with the update of the multipliers as

$$\begin{aligned} \Lambda_1(l) &\leftarrow \Lambda_1(l) + \mathbf{v}_1(l) - \langle \mathbf{b}_l \mathbf{b}_l^*, \mathbf{H} \rangle; \\ \Lambda_2(l) &\leftarrow \Lambda_2(l) + \mathbf{v}_2(l) - \langle \mathbf{c}_l \mathbf{c}_l^*, \mathbf{M} \rangle; \\ \Lambda_3 &\leftarrow \Lambda_3 + \mathbf{H}' - \mathbf{H}; \\ \Lambda_4 &\leftarrow \Lambda_4 + \mathbf{M}' - \mathbf{M}. \end{aligned}$$

Please see more details in the appendix of Ahmed et al. (2018).

As reported in Ahmed et al. (2018), this convex method showed excellent agreement with the theorem in the case of random subspaces. However, it was less effective on deterministic subspaces, including partial discrete cosine transforms or partial discrete wavelet transforms. One should also notice that although the model

is convex, the lifting technique increased the dimension of original nonconvex optimization problem greatly, at the order of square of the original dimension, causing huge memory requirement as well as computational complexity. That may limit practical applications, especially for reconstructing 2D images or volumes.

It seems rather difficult to adopt the same convex method for other cases of BPR, since they cannot be rewritten as the same form as (50). Convexifying a general BPR problem should be an interesting research direction in the future.

Second-Order Algorithm Using Hessian

The second-order algorithms relying on the Hessian of the nonlinear optimization problems have also been developed for PR problem, such as using Newton method (NT) (Qian et al. 2014; Yeh et al. 2015), Levenberg-Marquardt method (LM) (Ma et al. 2018; Kandel et al. 2021), or Gauss-Newton algorithm (GN) (Gao and Xu 2017). Consider the following problem by rewriting (13)

$$\min_u Q(u), \quad (56)$$

with $Q(u) := \mathcal{M}(|A_w u|^2, f)$. Given the initial guess u^0 ,

$$Q(u) \approx Q(u^0) + \Re(\nabla_u Q(u^0), u - u^0) + \frac{1}{2} \Re(\langle \nabla_u^2 Q(u^0)(u - u^0), u - u^0 \rangle), \quad (57)$$

where ∇_u^2 denotes the Hessian operator. Then a new estimate u^1 for the stationary point can be obtained by solving the following systems:

$$\nabla_u^2 Q(u_0)(u^1 - u^0) = -\nabla_u Q(u^0).$$

Assuming the Hessian matrix is nonsingular, the iterative scheme by NT is derived as

$$\text{Newton method: } u^{k+1} = u^k - (\nabla_u^2 Q(u^k))^{-1} \nabla_u Q(u^k) \quad \forall k. \quad (58)$$

The gradient is given below:

$$\nabla_u Q(u) = \begin{cases} A_w^* (A_w u - \frac{\sqrt{f}}{|A_w u|} \circ A_w u); & \text{(AGM)} \\ A_w^* (A_w u - \frac{f}{|A_w u|^2} \circ A_w u); & \text{(IPM)} \\ 2A_w^* (|A_w u|^2 \circ A_w u - f \circ A_w u); & \text{(IGM)} \end{cases} \quad (59)$$

where the objective function $Q(u)$ is rewritten as $Q(u) = \mathcal{M}(|A_w u|^2, f)$ by denoting the matrix A_w as (8), and the detailed forms of the operators can be found

in (9), (10), (11), and (12). The Hessian matrices for three metrics are complicated, and please see Appendix A of Yeh et al. (2015).

More efficient algorithms including LM and GN were developed, concerned with the nonlinear least squares problems (NLS) (56) with the AGM and IPM metrics (please see (14)). Namely, by denoting the residual function

$$\mathbf{r}(u) = \begin{cases} |A_w u| - \sqrt{f}; & \text{(AGM)} \\ |A_w u|^2 - f; & \text{(IGM)} \end{cases}$$

consider the NLS problem below:

$$\min_u \mathcal{Q}(u) = \frac{1}{2} \|\mathbf{r}(u)\|^2.$$

Then with Jacobian matrix as

$$\mathbf{J}(u) := \nabla_u \mathbf{r}(u) = \begin{cases} \text{diag}(\text{sign}(\text{conj}(A_w u))) A_w; & \text{(AGM)} \\ \text{diag}(\text{conj}(A_w u)) A_w & \text{(IGM);} \end{cases}$$

the GN method considered

$$GN(u) := \mathbf{J}^*(u) \mathbf{J}(u),$$

as an estimate of the Hessian matrix, that leads to the following iterative scheme:

$$\begin{aligned} \text{Gauss-Newton method: } u^{k+1} &= (\mathbf{J}^*(u^k) \mathbf{J}(u^k))^{-1} (u^k - \nabla_u \mathcal{Q}(u^k)) \\ &= (\mathbf{J}^*(u^k) \mathbf{J}(u^k))^{-1} (u^k - \mathbf{J}^*(u^k) \mathbf{r}(u^k)) \quad \forall k. \end{aligned} \quad (60)$$

Gao and Xu (2017) further proposed a global convergent GN algorithm with resampling for PR problem, which partial phaseless data was used to reformulate the GN matrix and the gradient per loop.

The Hessian matrix or the GN matrix cannot be guaranteed to be nonsingular practically. Hence, the LM method interpreted as a regularized variant of GN was proposed as

$$\text{LM method: } u^{k+1} = (\mathbf{J}^*(u^k) \mathbf{J}(u^k) + \mu^k \mathbf{I})^{-1} (u^k - \mathbf{J}^*(u^k) \mathbf{r}(u^k)) \quad \forall k \quad (61)$$

with the adaptive parameter μ^k . Readily one knows μ^k cannot be too large; otherwise, the Hessian information is useless. To obtain fast convergence, Marquardt (1963) proposed the following strategy for μ^k depending on the diagonal matrix of GN matrix as

$$\mu^k = \mu_0 \mathbf{D}_g(\mathbf{J}^*(u^k) \mathbf{J}(u^k)),$$

with $\mathbf{D}_g(A)$ denoting the diagonal matrix with the elements from the main diagonal of the matrix A . Yamashita and Fukushima (2001) and Fan and Yuan (2005) proposed the scheme depending on the objective function value below:

$$\mu^k = (\mathcal{Q}(u^k))^{\frac{\nu}{2}} \quad (62)$$

with $\nu \in [1, 2]$. Ma et al. (2018) further improved the scheme (62) as choosing a larger value when the iterative solution u^k is far away from the global minimizer, i.e.,

$$\mu^k = \text{Thresh}(u^k)(\mathcal{Q}(u^k))^{\frac{\nu}{2}},$$

with

$$\text{Thresh}(u) = \begin{cases} \tau, & \text{if } \mathcal{Q}(u^k) \geq c_0 \|u^k\|^2, \\ 1, & \text{otherwise,} \end{cases}$$

with $\tau \gg 1$ and parameter $c_0 > 0$.

The mentioned algorithms including Qian et al. (2014), Gao and Xu (2017), and Ma et al. (2018) focused on nonblind PR. With the generalized GN method and automatic differentiation, Kandel et al. (2021) proposed a variant LM algorithm for blind recovery, where, especially for IPM-based metric, it employed the generalized GN (GGN) as

$$GGN(u) := \mathbf{J}^*(u^k) \nabla_g^2 \mathcal{M}(|A_{u^k} w|^2, f) \mathbf{J}(u^k)$$

with $\mathcal{M}(g, f)$ defined in (14). Following the same manner with alternating minimization, one can easily derive the second-order algorithm for the blind problem as

$$\begin{aligned} u^{k+1} &= \arg \min_u \mathcal{M}(|A_{w^k} u|^2, f); \\ w^{k+1} &= \arg \min_w \mathcal{M}(|A_{u^{k+1}} w|^2, f); \end{aligned} \quad (63)$$

where both two subproblems are solved by NT, GN, or LM algorithms.

Subspace Method

The subspace method (Saad 2003; Xin et al. 2021) is a very powerful algorithm, iteratively refining the variable in the subspace of solution, which includes the Krylov subspace method as well-known conjugate gradient method, domain decomposition method, and multigrid method. It originally focused on solving the linear equations or least squares problems and now has been successfully extended to nonlinear

equations or nonlinear optimization problems. In this part, the subspace methods for the PR and BPR problems will be reviewed.

Nonlinear Conjugate Gradient Algorithm Consider the following optimization problem:

$$\min f(\mathbf{x}).$$

By the nonlinear conjugate gradient (NLCG) algorithm, the iterative scheme can be given below:

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha^k \mathbf{d}^k; \\ \mathbf{d}^k &= -\nabla_{\mathbf{x}} f(\mathbf{x}^k) + \beta^{k-1} \mathbf{d}^{k-1} \quad \forall k \geq 1, \end{aligned} \quad (64)$$

with the stepsize α^k and weight β^{k-1} , where \mathbf{d}^k is the search direction. One may notice that the search direction \mathbf{d}^k in NLCG is the combination of the gradient and the search direction \mathbf{d}^{k-1} with the weight β^{k-1} . To get optimal parameters, the stepsize α^k is selected by the monotone line search procedures, while the weight β^k is determined based on the gradient $\nabla_{\mathbf{x}} f(\mathbf{x}^{k-1})$, $\nabla_{\mathbf{x}} f(\mathbf{x}^k)$ and the search direction \mathbf{d}^{k-1} (typically five different formulas (Xin et al. 2021)).

The NLCG has been successfully applied to the BPR problem (Thibault and Guizar-Sicairos 2012; Qian et al. 2014). For example, Thibault and Guizar-Sicairos (2012) adopted the NLCG to solve the CDI problem. The iterative scheme can be given below:

$$\begin{aligned} (w^{k+1}, u^{k+1}) &= (w^k, u^k) + \alpha^k \mathbf{\Delta}^k; \\ \mathbf{\Delta}^k &= -\mathbf{g}^k + \beta^{k-1} \mathbf{\Delta}^{k-1} \quad \forall k \geq 1, \end{aligned} \quad (65)$$

with the gradient $\mathbf{g}^k := (\nabla_w \mathcal{M}(|\mathcal{A}(w^k, u^k)|^2, f), \nabla_u \mathcal{M}(|\mathcal{A}(w^k, u^k)|^2, f))$ calculated by (59) and $\mathbf{\Delta} := (\mathbf{\Delta}_w, \mathbf{\Delta}_u)$. The weight β^{k-1} is derived by the Polak-Ribère formula as

$$\beta^{k-1} = \frac{\langle \mathbf{g}^k, \mathbf{g}^k \rangle - \Re(\langle \mathbf{g}^k, \mathbf{g}^{k-1} \rangle)}{\langle \mathbf{g}^{k-1}, \mathbf{g}^{k-1} \rangle}.$$

To further get α^k , by estimating $\mathcal{M}(|\mathcal{A}(w^k + \alpha \mathbf{\Delta}_w, u^k + \alpha \mathbf{\Delta}_u)|^2, f)$ by the low-order polynomial as

$$\mathcal{M}(|\mathcal{A}(w^k + \alpha \mathbf{\Delta}_w, u^k + \alpha \mathbf{\Delta}_u)|^2, f) \approx \sum_{t=0}^8 c_t \alpha^t,$$

the authors adopted the Newton–Raphson algorithm in order to minimize the following one-dimensional problem:

$$\alpha^k := \arg \min_{\alpha} \sum_{t=0}^8 c_t \alpha^t.$$

Domain Decomposition Method The domain decomposition methods (DDMs) allow for highly parallel computing with good load balance, by decomposing the equations on whole domain to the problems on relatively small subdomains with information synchronization on the partition interfaces. They have played a great role in solving partial differential equations numerically and recently been successfully extended to large-scale image restoration, image reconstruction, and other inverse problems, e.g., Xu et al. (2010), Chang et al. (2015, 2021), Langer and Gaspoz (2019), Lee et al. (2019), and references therein. For ptychography imaging, several parallel algorithms (Nashed et al. 2014; Guizar-Sicairos et al. 2014; Marchesini et al. 2016; Enfedaque et al. 2019; Chang et al. 2021) have been developed. Specially, for convention ptychography, Chang et al. (2021) proposed an overlapping DDM with the ST-AGM as defined in (18), with fewer communication cost and theoretical convergence guarantee.

First, give the domain decomposition. Denote the whole region $\Omega := \{0, 1, 2, \dots, n - 1\}$ in the discrete setting. There exists the two-subdomain overlapping DD $\{\Omega_d\}_{d=1}^2$, such that

$$\Omega = \bigcup_{d=1}^2 \Omega_d$$

with $\Omega_d := \{l_0^d, l_1^d, \dots, l_{n_d-1}^d\}$, and the overlapping region is denoted as

$$\Omega_{1,2} := \Omega_1 \cap \Omega_2 = \{\hat{l}_0, \hat{l}_1, \dots, \hat{l}_{\hat{n}-1}\}.$$

Here consider a special overlapping DD as shown in Fig. 2. Denote the restriction operators R_1, R_2 as

$$R_d u = u|_{\Omega_d}, \quad R_{1,2} u = u|_{\Omega_{1,2}},$$

i.e.,

$$(R_d u)(j) = u(l_j^d) \forall 0 \leq j \leq n_d - 1,$$

$$(R_{1,2} u)(j) = u(\hat{l}_j) \forall 0 \leq j \leq \hat{n} - 1.$$

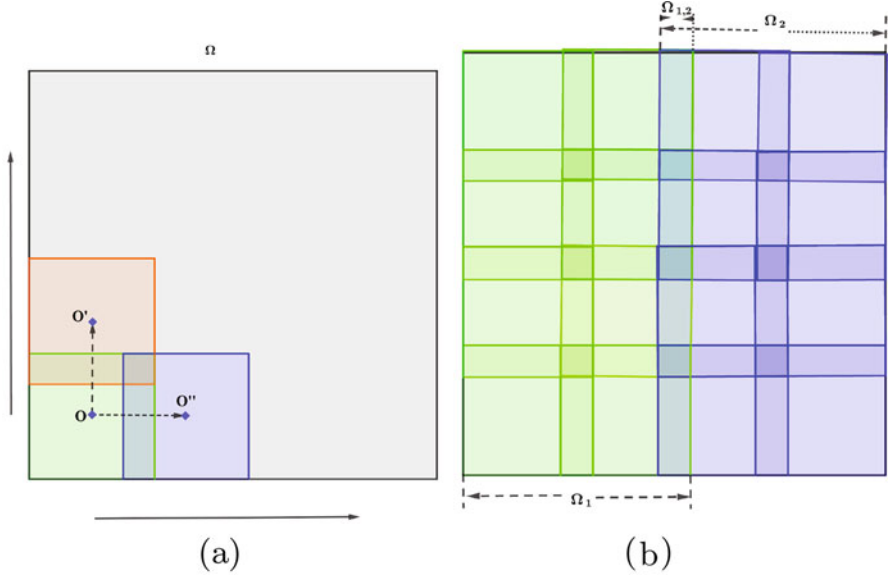


Fig. 2 (a) Ptychography scan in the domain Ω (grid scan): the starting scan centers at point \mathbf{O} and then moves up (or to the right) with the center point \mathbf{O}' (or \mathbf{O}''); (b) two-subdomain DD (totally 4×4 frames): The subdomains Ω_1, Ω_2 are generated by two 4×2 scans, and the overlapping region $\Omega_{1,2} = \Omega_1 \cap \Omega_2$

Then two groups of localized shift operators can be introduced $\{\mathcal{S}_{j_d}^d\}_{j_d=0}^{J_d-1}$ for $d = 1, 2$ with $\sum_d J_d = J$.

For nonblind problem, denote the linear operators A_1, A_2 on the subdomains as

$$A_d u_d := ((\mathcal{F}(w \circ \mathcal{S}_0^d u_d))^T, (\mathcal{F}(w \circ \mathcal{S}_1^d u_d))^T, \dots, (\mathcal{F}(w \circ \mathcal{S}_{J_d-1}^d u_d))^T)^T, \quad (66)$$

for $d = 1, 2$. Based on the continuity on the overlapping regions, one has

$$\pi_{1,2} u_1 = \pi_{2,1} u_2, \quad (67)$$

where the operators $\pi_{1,2}$ (restriction from Ω_1 into $\Omega_{1,2}$) and $\pi_{2,1}$ (restriction from Ω_2 into $\Omega_{1,2}$) are denoted as

$$\pi_{1,2} u_1 := R_{1,2} R_1^T u_1,$$

and

$$\pi_{2,1} u_2 := R_{1,2} R_2^T u_2.$$

Naturally, the measurement f is also decomposed to two nonoverlapping parts f_1, f_2 , i.e.,

$$f_d := |A_d R_d u|^2.$$

Hereafter, consider the nonlinear optimization problem with ST-AGM. In order to enable the parallel computing of u_1 and u_2 , introduce an auxiliary variable v which is only defined in the overlapping region $\Omega_{1,2}$, and then is concerned with the following model:

$$\begin{aligned} \min_{u_1, u_2, v} \quad & \sum_{d=1}^2 \mathcal{G}_\epsilon(A_d u_d; f_d), \\ \text{s.t.} \quad & \pi_{d,3-d} u_d - v = 0, \quad d = 1, 2. \end{aligned} \quad (68)$$

In order to develop an iterative scheme without inner loop as well as with fast convergence for large-step scan, two auxiliary variables z_1, z_2 are introduced below:

$$\begin{aligned} \min_{u_1, u_2, v, z_1, z_2} \quad & \sum_{d=1}^2 \mathcal{G}_\epsilon(z_d; f_d), \\ \text{s.t.} \quad & \pi_{d,3-d} u_d - v = 0, \quad A_d u_d - z_d = 0, \quad d = 1, 2. \end{aligned} \quad (69)$$

Then it is quite standard to solve the saddle point problem by ADMM. The details are omitted here, and please see more details in Chang et al. (2021).

Then for blind recovery, in order to reduce the grid pathology (Chang et al. 2019a) (ambiguity derived by the multiplication of any periodical function and the true solution) due to grid scan, introduce the support set constraint of the probe, i.e., $\mathcal{O} := \{w : (\mathcal{F}w)(j) = 0, j \in \mathcal{J}\}$, with the support set \mathcal{J} denoted as the complement of the set $\bar{\mathcal{J}}$ (index set for zero values for the Fourier transform of the probe). Then consider the blind ptychography problem for two-subdomain DD:

$$\begin{aligned} \min_{\{w, u_1, u_2, v\}} \quad & \sum_{d=1}^2 \mathcal{G}_\epsilon(\mathcal{A}_d(w, u_d); f_d) + \mathbb{I}_{\mathcal{O}}(w), \\ \text{s.t.} \quad & \pi_{d,3-d} u_d - v = 0, \quad d = 1, 2, \end{aligned}$$

where the bilinear mapping $\mathcal{A}_d(w, u_d)$ is denoted as

$$(\mathcal{A}_d)_{j_d}(w, u_d) := \mathcal{F}(w \circ (S_{j_d}^d u_d)) \quad \forall 0 \leq j_d \leq J_d - 1,$$

with $\sum_{d=1}^2 J_d = J$, and the indicator function $\mathbb{I}_{\mathcal{O}}$. To enable parallel computing, consider the following constraint optimization problems:

$$\begin{aligned}
& \min_{\{w, w_1, w_2, u_1, u_2, v, z_1, z_2\}} \sum_{d=1}^2 \mathcal{G}_\epsilon(z_d; f_d) + \mathbb{I}_{\mathcal{O}}(w) \\
& \text{s.t.} \quad \pi_{1,2}u_1 - v = 0, \quad \pi_{2,1}u_2 - v = 0, \\
& \quad \quad w_d = w, \quad z_d = \mathcal{A}_d(w_d, u_d), \quad d = 1, 2,
\end{aligned}$$

which was also efficiently solved by ADMM.

Multigrid Methods The multigrid method (MG) is a standard framework in order to accelerate solving partial differential equations (Hackbusch 1985), large-scale linear equations (Xu and Zikatanov 2017), and related optimization problems (Borzi and Schulz 2009) with the full approximation scheme (FAS) (Brandt and Livne 2011). A multigrid-based optimization framework based on Nash (2000) to reduce the computational for nonblind ptychographic PR was proposed by Fung and Wendy (2020), which utilized the hierarchical structures of the measured data.

Consider the following feasible problem (Fung and Wendy 2020) as

$$\min_u \sum_j \|\mathcal{F}^*(\sqrt{f_j} \circ \text{sign}(\mathcal{F}(w \circ \mathcal{S}_j u))) - w \circ \mathcal{S}_j u\|^2, \quad (70)$$

which is equivalent to the problem

$$\min_u \mathcal{M}(|\mathcal{A}(w, u)|^2, f)$$

using the AGM metric. Then the multigrid optimization framework based on FAS was further developed, where the coarse-grid subproblem was interpreted as a first-order approximation to the fine-grid problem. However, it is unclear how to extend the current algorithm to the blind problem.

Discussions

Experimental Issues

Probe Drift Probe drift happens in ptychography, when the data is very noisy. The mass center of the iterative probe will eventually touch the boundary such that the iterative algorithms fail eventually. Hence, the joint reconstruction will cause instability of the iterative algorithms from noisy experimental data. One simple strategy proposed by Marchesini et al. (2016) is to shift the probe to the mass center of the complex image periodically. Other possible way is to consider the compact support condition for the probe, or to get additional measurement for the probe by letting the light go through the vacuum as Marchesini et al. (2016) and Chang

et al. (2019a). The related numerical stability shall be investigated, and one can refer to Huang and Xu (2020, 2021) for nonblind PR.

Flat Samples When the sample is nearly flat (such as weak absorption or scattering for biological specimens using hard X-ray sources), there will be no sufficient diversity of the measured phaseless data even by very dense scan. In such case, the iterative algorithms mentioned in this survey will become slow, and the recovered image quality gets worse. Acquiring of scattering map by linearization for large features of the sample (Dierolf et al. 2010b) or modeling with additional Kramers-Kronig relation (KKR) (Hirose et al. 2017) was exploited to improve the reconstruction quality. Besides, pairwise relations between adjacent frames were considered in Marchesini and Wu (2014) to accelerate projection algorithms for the flat sample.

Background Retrieval Parasitic scattering termed as background often happens experimentally, which may come from any element along the beam path other than the sample and the optical elements desired harmonic order (Chang et al. 2019b). Direct reconstruction without background removal will introduce structural artifacts to the reconstruction images. Several methods were designed, such as preconditioned gradient descent (Marchesini et al. 2013), preprocessing method (Wang et al. 2017), and ADMM for nonlinear optimization method with framewise-invariant background (Chang et al. 2019b). It is still a challenging problem since the practical background is sophisticated and cannot be assumed to be framewise invariant.

High-Dimensional Problems The formula for all four cases for BPR holds for a thin (2D) object in paraxial approximation. For thick samples, the linear propagation as (7) will cause obvious errors, and one has to consider the nonlinear transform as Dierolf et al. (2010a). Other than the 3D imaging, high-dimensional problems may result from the spectromicroscopy (Maiden et al. 2013), multimode decomposition of partial coherence (Thibault and Menzel 2013; Chang et al. 2018a), and dichroic ptychography (Chang et al. 2020; Lo et al. 2021). Such strong nonlinearity coupling with the high-dimensional optimization causes difficulties for designing the stable and high-throughput algorithm.

Theoretical Analysis

Convergence of Iterative Algorithms Other than the projection onto nonconvex modulus constraint for nonblind PR, APs (Thibault et al. 2009; Marchesini et al. 2016) for BPR involve the bilinear constraint set. Some progress has been made for the general PR problem using projection algorithms (Hesse and Luke 2013; Marchesini et al. 2015; Chen and Fannjiang 2016). However, the corresponding convergence theories for BPR are still unclear. Moreover, only the PHeBIE- (Hesse et al. 2015) and ADMM-based algorithm (Chang et al. 2019a) for BPR provided

rigorous convergence analysis. Hence, it is of great importance to either study the convergence of existing algorithms or develop new algorithms with clear convergence guarantee in the future.

Uniqueness Analysis Uniqueness can be guaranteed for 1D nonblind ptychographic PR for nonvanishing signals with the probe of proper size (Jaganathan et al. 2016). It can also be guaranteed for BPR (Bendory et al. 2019). By letting two signals lie in low-dimensional random subspaces, the uniqueness was obtained (Ahmed et al. 2018) with sufficient measurements. For 2D imaging problems, with a randomly phased probe, the uniqueness can be proved for the measurements which is strongly connected and possesses an anchor. See more discussions on more general cases together with sparse signals in Grohs et al. (2020). Readily for ptychography, nontrivial ambiguity including periodical function and linear phase exists for raster scan. Rigorous analysis about more general ambiguity was given (Fannjiang 2019). Experimentally more flexible spiral or random scan (Huang et al. 2014) has been exploited for stable recovery.

Further Discussions

Recently, some efficient algorithms have been developed for nonblind PR, such as the second-order algorithms including Ma et al. (2018), Gao and Xu (2017), and the multigrid method (Fung and Wendy 2020); however, it is not clear how they can be applied to the blind problem. Hence, we only list the algorithms for the BPR problem included in this survey, and please see the overview in Table 1.

Then we discuss the advantages and disadvantages of all listed algorithms. As the unique convex method, the convex programming (Ahmed et al. 2018) provided a convex relaxation such that it can gain the global minimizer. The dimension of the lifted matrix is much higher than that of the original form leading to the iterative algorithm with high complexity, and therefore it seems more impractical for real experimental analysis. Moreover, it is limited to the convolutional PR as the special case of BPR, since it relies on the structure as (50). All other listed algorithms designed based on the nonconvex optimization problem work well for perfect data (smaller scan stepsizes to guarantee enough redundancy and long exposure with high signal-to-noise ratio (SNR)). The AP, ePIE-type, proximal, and ADMM algorithms are of the first order and have closed-form expression for all iterative steps, all of which have already been efficiently implemented for practical ptychography and Fourier ptychography imaging instrument with low computational complexity. As reported in Chang et al. (2019a), the ePIE algorithm may get unstable for noisy measurements, and it seems more sensitive to the scan stepsizes for ptychography imaging, while the ADMM algorithm (Chang et al. 2019a) for ptychography imaging can offer promising performance even for noisy and insufficient data. The second-order algorithms utilizing the Hessian usually requires much more computation cost, and it can be accelerated by Gauss-Newton

or Levenberg-Marquardt methods without direct calculation of the Hessian. Further requirement of parallel computing may consider the DDM (Chang et al. 2021).

Conclusions

In this survey, a short review of the iterative algorithms is provided for the nonlinear optimization problem arising from the BPR problem, mainly consisting of three types of algorithms as the first-order operator-splitting algorithms and second-order algorithms and subspace methods. There still exist sophisticated experimental issues and challenging theoretical analysis, which are further discussed in the last part. Learning-based methods have been a powerful tool for solving inverse problem and PR problems, which are not included in this survey.

This survey focuses on the BPR problems with forms expressed as (6). However, not all the BPR problem belongs to the categories of (6). Very recently, a resolution-enhanced parallel coded ptychography (CP) technique (Jiang et al. 2021, 2022) was reported which achieves the highest numerical aperture. With the sample u and the transmission profile of the engineered surface w , the phaseless data was generated as

$$f_j^{CP} = |(w \circ (\mathcal{S}_j u \otimes \kappa_1)) \otimes \kappa_2|^2,$$

with κ_1, κ_2 as two known PSFs. Such advanced cases should be further investigated.

Table 1 Overview of all iterative algorithms for the blind phase retrieval (BPR) problem in this survey. “Y” and “N” are short for “yes” and “no”, respectively

Name	Refs	Convex(Y/N)	Convergence(Y/N)
Alternating projection	Thibault et al. (2009); Marchesini et al. (2016)	N	N
ePIE-type algorithms	Maiden and Rodenburg (2009); Maiden et al. (2017)	N	N
Proximal algorithms	Hesse et al. (2015); Yan (2020)	N	Y (Hesse et al. 2015)
ADMM	Chang et al. (2019a); Fannjiang and Zhang (2020)	N	Y (Chang et al. 2019a)
Convex programming	Ahmed et al. (2018)	Y	Y
Second-order algorithms	Yeh et al. (2015); Kandel et al. (2021)	N	N
Subspace method	Thibault and Guizar-Sicairos (2012); Qian et al. (2014); Chang et al. (2021)	N	Y (Chang et al. 2021)

Acknowledgments The work of the first author was partially supported by the NSFC (Nos. 11871372, 11501413) and Natural Science Foundation of Tianjin (18JCYBJC16600). The authors would like to thank Prof. Guoan Zheng for the helpful discussions.

References

- Ahmed, A., Aghasi, A., Hand, P.: Blind deconvolutional phase retrieval via convex programming (2018). *NeurIPS* (arXiv:1806.08091)
- Bendory, T., Sidorenko, P., Eldar, Y.C.: On the uniqueness of frog methods. *IEEE Sig. Process. Lett.* **24**(5), 722–726 (2017)
- Bendory, T., Edidin, D., Eldar, Y.C.: Blind phaseless short-time fourier transform recovery. *IEEE Trans. Inf. Theory* **66**(5), 3232–3241 (2019)
- Bohte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **146**(1–2), 459–494 (2014)
- Borzi, A., Schulz, V.: Multigrid methods for pde optimization. *SIAM Rev.* **51**(2), 361–395 (2009)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011)
- Brandt, A., Livne, O.E.: *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, Revised Edition. SIAM, Philadelphia (2011)
- Cai, J.-F., Huang, M., Li, D., Wang, Y.: The global landscape of phase retrieval II: quotient intensity models (2021). arXiv preprint arXiv:2112.07997
- Candes, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Trans. Inf. Theory* **61**(4), 1985–2007 (2015)
- Chang, H., Tai, X.-C., Wang, L.-L., Yang, D.: Convergence rate of overlapping domain decomposition methods for the Rudin-Osher-Fatami model based on a dual formulation. *SIAM J. Image Sci.* **8**, 564–591 (2015)
- Chang, H., Lou, Y., Ng, M.K., Zeng, T.: Phase retrieval from incomplete magnitude information via total variation regularization. *SIAM J. Sci. Comput.* **38**(6), A3672–A3695 (2016)
- Chang, H., Enfedaque, P., Lou, Y., Marchesini, S.: Partially coherent ptychography by gradient decomposition of the probe. *Acta Crystallogr. Sect. A: Found. Adv.* **74**(3), 157–169 (2018a)
- Chang, H., Lou, Y., Duan, Y., Marchesini, S.: Total variation–based phase retrieval for Poisson noise removal. *SIAM J. Imaging Sci.* **11**(1), 24–55 (2018b)
- Chang, H., Marchesini, S., Lou, Y., Zeng, T.: Variational phase retrieval with globally convergent preconditioned proximal algorithm. *SIAM J. Imaging Sci.* **11**(1), 56–93 (2018c)
- Chang, H., Enfedaque, P., Marchesini, S.: Blind ptychographic phase retrieval via convergent alternating direction method of multipliers. *SIAM J. Imaging Sci.* **12**(1), 153–185 (2019a)
- Chang, H., Enfedaque, P., Zhang, J., Reinhardt, J., Enders, B., Yu, Y.-S., Shapiro, D., Schroer, C.G., Zeng, T., Marchesini, S.: Advanced denoising for x-ray ptychography. *Opt. Express* **27**(8), 10395–10418 (2019b)
- Chang, H., Marcus, M.A., Marchesini, S.: Analyzer-free linear dichroic ptychography. *J. Appl. Crystallogr.* **53**(5), 1283–1292 (2020)
- Chang, H., Glowinski, R., Marchesini, S., Tai, X.-C., Wang, Y., Zeng, T.: Overlapping domain decomposition methods for ptychographic imaging. *SIAM J. Sci. Comput.* **43**(3), B570–B597 (2021)
- Chapman, H.N.: Phase-retrieval x-ray microscopy by wigner-distribution deconvolution. *Ultramicroscopy* **66**(3), 153–172 (1996)
- Chen, Y., Candes, E.: Solving random quadratic systems of equations is nearly as easy as solving linear systems. In: *Advances in Neural Information Processing Systems*, pp. 739–747 (2015)
- Chen, P., Fannjiang, A.: Fourier phase retrieval with a single mask by douglas–rachford algorithms. *Appl. Comput. Harmon. Anal.* **44**(3), 665–699 (2016)

- Dierolf, M., Menzel, A., Thibault, P., Schneider, P., Kewish, C.M., Wepf, R., Bunk, O., Pfeiffer, F.: Ptychographic x-ray computed tomography at the nanoscale. *Nature* **467**(7314), 436–439 (2010a)
- Dierolf, M., Thibault, P., Menzel, A., Kewish, C.M., Jefimovs, K., Schlichting, I., von König, K., Bunk, O., Pfeiffer, F.: Ptychographic coherent diffractive imaging of weakly scattering specimens. *New J. Phys.* **12**(3), 035017 (2010b)
- Elser, V.: Phase retrieval by iterated projections. *J. Opt. Soc. Am. A* **20**(1), 40–55 (2003)
- Elser, V., Lan, T.-Y., Bendory, T.: Benchmark problems for phase retrieval. *SIAM J. Imaging Sci.* **11**(4), 2429–2455 (2018)
- Enfedaque, P., Chang, H., Enders, B., Shapiro, D., Marchesini, S.: High performance partial coherent x-ray ptychography. In: *International Conference on Computational Science*, pp. 46–59. Springer (2019)
- Fan, J.-Y., Yuan, Y.-X.: On the quadratic convergence of the levenberg-marquardt method without nonsingularity assumption. *Computing* **74**(1), 23–39 (2005)
- Fannjiang, A.: Raster grid pathology and the cure. *Multiscale Model. Simul.* **17**(3), 973–995 (2019)
- Fannjiang, A., Strohmer, T.: The numerics of phase retrieval. *Acta Numer.* **29**, 125–228 (2020)
- Fannjiang, A., Zhang, Z.: Fixed point analysis of douglas–rachford splitting for ptychography and phase retrieval. *SIAM J. Imaging Sci.* **13**(2), 609–650 (2020)
- Fung, S.W., Wendy, Z.: Multigrid optimization for large-scale ptychographic phase retrieval. *SIAM J. Imaging Sci.* **13**(1), 214–233 (2020)
- Gao, B., Xu, Z.: Phaseless recovery using the Gauss–Newton method. *IEEE Trans. Sig. Process.* **65**(22), 5885–5896 (2017)
- Gao, B., Wang, Y., Xu, Z.: Solving a perturbed amplitude-based model for phase retrieval. *IEEE Trans. Sig. Process.* **68**, 5427–5440 (2020)
- Godard, P., Allain, M., Chamard, V., Rodenburg, J.: Noise models for low counting rate coherent diffraction imaging. *Opt. Express* **20**(23), 25914–25934 (2012)
- Grohs, P., Koppensteiner, S., Rathmair, M.: Phase retrieval: uniqueness and stability. *SIAM Rev.* **62**(2), 301–350 (2020)
- Guizar-Sicairos, M., Fienup, J.R.: Phase retrieval with transverse translation diversity: a nonlinear optimization approach. *Opt. Express* **16**(10), 7264–7278 (2008)
- Guizar-Sicairos, M., Johnson, I., Diaz, A., Holler, M., Karvinen, P., Stadler, H.-C., Dinapoli, R., Bunk, O., Menzel, A.: High-throughput ptychography using eiger: scanning x-ray nano-imaging of extended regions. *Opt. Express* **22**(12), 14859–14870 (2014)
- Gürsoy, D., Chen-Wiegart, Y.-C.K., Jacobsen, C.: Lensless x-ray nanoimaging: revolutions and opportunities. *IEEE Sig. Process. Mag.* **39**(1), 44–54 (2022)
- Hackbusch, W.: *Multi-grid Methods and Applications*, Springer, Berlin, Heidelberg (1985)
- Hesse, R., Luke, D.R.: Nonconvex notions of regularity and convergence of fundamental algorithms for feasibility problems. *SIAM J. Optim.* **23**(4), 2397–2419 (2013)
- Hesse, R., Luke, D.R., Sabach, S., Tam, M.K.: Proximal heterogeneous block implicit-explicit method and application to blind ptychographic diffraction imaging. *SIAM J. Imaging Sci.* **8**(1), 426–457 (2015)
- Hirose, M., Shimomura, K., Burdet, N., Takahashi, Y.: Use of Kramers-Kronig relation in phase retrieval calculation in x-ray spectro-ptychography. *Opt. Express* **25**(8), 8593–8603 (2017)
- Huang, M., Xu, Z.: The estimation performance of nonlinear least squares for phase retrieval. *IEEE Trans. Inf. Theory* **66**(12), 7967–7977 (2020)
- Huang, M., Xu, Z.: Uniqueness and stability for the solution of a nonlinear least squares problem (2021). arXiv preprint arXiv:2104.10841
- Huang, X., Yan, H., Harder, R., Hwu, Y., Robinson, I.K., Chu, Y.S.: Optimization of overlap uniformness for ptychography. *Opt. Express* **22**(10), 12634–12644 (2014)
- Huang, Y., Jiang, S., Wang, R., Song, P., Zhang, J., Zheng, G., Ji, X., Zhang, Y.: Ptychography-based high-throughput lensless on-chip microscopy via incremental proximal algorithms. *Opt. Express* **29**(23), 37892–37906 (2021)

- Jaganathan, K., Eldar, Y.C., Hassibi, B.: Stft phase retrieval: uniqueness guarantees and recovery algorithms. *IEEE J. Sel. Top. Sig. Process.* **10**(4), 770–781 (2016)
- Jiang, S., Guo, C., Song, P., Zhou, N., Bian, Z., Zhu, J., Wang, R., Dong, P., Zhang, Z., Liao, J. et al.: Resolution-enhanced parallel coded ptychography for high-throughput optical imaging. *ACS Photon.* **8**(11), 3261–3271 (2021)
- Jiang, S., Guo, C., Bian, Z., Wang, R., Zhu, J., Song, P., Hu, P., Hu, D., Zhang, Z., Hoshino, K. et al.: Ptychographic sensor for large-scale lensless microbial monitoring with high spatiotemporal resolution. *Biosens. Bioelectron.* **196**, 113699 (2022)
- Kandel, S., Maddali, S., Nashed, Y.S., Hruszkewycz, S.O., Jacobsen, C., Allain, M.: Efficient ptychographic phase retrieval via a matrix-free levenberg-marquardt algorithm. *Opt. Express* **29**(15), 23019–23055 (2021)
- Kane, D.J., Vakhtin, A.B.: A review of ptychographic techniques for ultrashort pulse measurement. *Progress Quantum Electron.* vol. 81, 100364 (2021)
- Langer, A., Gaspoz, F.: Overlapping domain decomposition methods for total variation denoising. *SIAM J. Numer. Anal.* **57**(3), 1411–1444 (2019)
- Lee, C.-O., Park, E.-H., Park, J.: A finite element approach for the dual Rudin–Osher–Fatemi model and its nonoverlapping domain decomposition methods. *SIAM J. Sci. Comput.* **41**(2), B205–B228 (2019)
- Lo, Y.H., Zhou, J., Rana, A., Morrill, D., Gentry, C., Enders, B., Yu, Y.-S., Sun, C.-Y., Shapiro, D.A., Falcone, R.W., Kapteyn, H.C., Murnane, M.M., Gilbert, P.U.P.A., Miao, J.: X-ray linear dichroic ptychography. *Proc. Natl. Acad. Sci.* **118**(3), 2019068118 (2021)
- Luke, D.R.: Relaxed averaged alternating reflections for diffraction imaging. *Inverse Probl.* **21**(1), 37–50 (2005)
- Ma, C., Liu, X., Wen, Z.: Globally convergent levenberg-marquardt method for phase retrieval. *IEEE Trans. Inf. Theory* **65**(4), 2343–2359 (2018)
- Maiden, A.M., Rodenburg, J.M.: An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* **109**(10), 1256–1262 (2009)
- Maiden, A., Morrison, G., Kaulich, B., Gianoncelli, A., Rodenburg, J.: Soft x-ray spectromicroscopy using ptychography with randomly phased illumination. *Nat. Commun.* **4**, 1669 (2013)
- Maiden, A., Johnson, D., Li, P.: Further improvements to the ptychographical iterative engine. *Optica* **4**(7), 736–745 (2017)
- Marchesini, S.: Invited article: a unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* **78**(1), 011301 (2007)
- Marchesini, S., Wu, H.-T.: Rank-1 accelerated illumination recovery in scanning diffractive imaging by transparency estimation (2014). arXiv preprint arXiv:1408.1922
- Marchesini, S., Schirotzek, A., Yang, C., Wu, H.-T., Maia, F.: Augmented projections for ptychographic imaging. *Inverse Probl.* **29**(11), 115009 (2013)
- Marchesini, S., Tu, Y.-C., Wu, H.-T.: Alternating projection, ptychographic imaging and phase synchronization. *Appl. Comput. Harmon. Anal.* **41**(3), 815–851 (2015)
- Marchesini, S., Krishnan, H., Shapiro, D.A., Perciano, T., Sethian, J.A., Daurer, B.J., Maia, F.R.: SHARP: a distributed, GPU-based ptychographic solver. *J. Appl. Crystallogr.* **49**(4), 1245–1252 (2016)
- Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **11**(2), 431–441 (1963)
- Nash, S.G.: A multigrid approach to discretized optimization problems. *Optim. Methods Softw.* **14**(1–2), 99–116 (2000)
- Nashed, Y.S., Vine, D.J., Peterka, T., Deng, J., Ross, R., Jacobsen, C.: Parallel ptychographic reconstruction. *Opt. Express* **22**(26), 32082–32097 (2014)
- Odstrčil, M., Menzel, A., Guizar-Sicairos, M.: Iterative least-squares solver for generalized maximum-likelihood ptychography. *Opt. Express* **26**(3), 3108–3123 (2018)
- Ou, X., Zheng, G., Yang, C.: Embedded pupil function recovery for fourier ptychographic microscopy. *Opt. Express* **22**(5), 4960–4972 (2014)
- Pfeiffer, F.: X-ray ptychography. *Nat. Photon* **12**, 9–17 (2018)

- Qian, J., Yang, C., Schirotzek, A., Maia, F., Marchesini, S.: Efficient algorithms for ptychographic phase retrieval. *Inverse Probl. Appl. Contemp. Math.* **615**, 261–280 (2014)
- Qu, Q., Zhang, Y., Eldar, Y.C., Wright, J.: Convolutional phase retrieval. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6088–6098 (2017)
- Qu, Q., Zhang, Y., Eldar, Y.C., Wright, J.: Convolutional phase retrieval via gradient descent. *IEEE Trans. Inf. Theory* **66**(3), 1785–1821 (2019)
- Reinhardt, J., Hoppe, R., Hofmann, G., Damsgaard, C.D., Patommel, J., Baumbach, C., Baier, S., Rochet, A., Grunwaldt, J.-D., Falkenberg, G., Schroer, C.G.: Beamstop-based low-background ptychography to image weakly scattering objects. *Ultramicroscopy* **173**, 52–57 (2017)
- Rodenburg, J.M.: Ptychography and related diffractive imaging methods. *Adv. Imaging Electron Phys.* **150**, 87–184 (2008)
- Saad, Y.: *Iterative Methods for Sparse Linear Systems*, 2nd edn. Society for Industrial and Applied Mathematics (2003)
- Shechtman, Y., Eldar, Y.C., Cohen, O., Chapman, H.N., Miao, J., Segev, M.: Phase retrieval with application to optical imaging: a contemporary overview. *Sig. Process. Mag. IEEE* **32**(3), 87–109 (2015)
- Sun, J., Qu, Q., Wright, J.: A geometric analysis of phase retrieval. In: 2016 IEEE International Symposium on Information Theory (ISIT), pp. 2379–2383. IEEE (2016)
- Thibault, P., Guizar-Sicairos, M.: Maximum-likelihood refinement for coherent diffractive imaging. *New J. Phys.* **14**(6), 063004 (2012)
- Thibault, P., Menzel, A.: Reconstructing state mixtures from diffraction measurements. *Nature* **494**(7435), 68–71 (2013)
- Thibault, P., Dierolf, M., Bunk, O., Menzel, A., Pfeiffer, F.: Probe retrieval in ptychographic coherent diffractive imaging. *Ultramicroscopy* **109**(4), 338–343 (2009)
- Trebino, R., DeLong, K.W., Fittinghoff, D.N., Sweetser, J.N., Krumbügel, M.A., Richman, B.A., Kane, D.J.: Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating. *Rev. Sci. Instrum.* **68**(9), 3277–3295 (1997)
- Wang, C., Xu, Z., Liu, H., Wang, Y., Wang, J., Tai, R.: Background noise removal in x-ray ptychography. *Appl. Opt.* **56**(8), 2099–2111 (2017)
- Wen, Z., Yang, C., Liu, X., Marchesini, S.: Alternating direction methods for classical and ptychographic phase retrieval. *Inverse Probl.* **28**(11), 115010 (2012)
- Wu, C., Tai, X.-C.: Augmented Lagrangian method, dual methods and split-Bregman iterations for ROF, vectorial TV and higher order models. *SIAM J. Imaging Sci.* **3**(3), 300–339 (2010)
- Xin, L., Zaiwen, W., Ya-Xiang, Y.: Subspace methods for nonlinear optimization. *CSIAM Trans. Appl. Math.* **2**(4), 585–651 (2021)
- Xu, J., Zikatanov, L.: Algebraic multigrid methods. *Acta Numer.* **26**, 591–721 (2017)
- Xu, J., Tai, X.-C., Wang, L.-L.: A two-level domain decomposition method for image restoration. *Inverse Probl. Imaging* **4**(3), 523–545 (2010)
- Yamashita, N., Fukushima, M.: On the rate of convergence of the levenberg-marquardt method. In: Alefeld, G., Chen, X. (eds.) *Topics in Numerical Analysis*, pp. 239–249. Springer, Vienna (2001)
- Yan, H.: Ptychographic phase retrieval by proximal algorithms. *New J. Phys.* **22**(2), 023035 (2020)
- Yeh, L.-H., Dong, J., Zhong, J., Tian, L., Chen, M., Tang, G., Soltanolkotabi, M., Waller, L.: Experimental robustness of fourier ptychography phase retrieval algorithms. *Opt. Express* **23**(26), 33214–33240 (2015)
- Zheng, G., Horstmeyer, R., Yang, C.: Wide-field, high-resolution fourier ptychographic microscopy. *Nat. Photon.* **7**, 739–745 (2013)
- Zheng, G., Shen, C., Jiang, S., Song, P., Yang, C.: Concept, implementations and applications of fourier ptychography. *Nat. Rev. Phys.* **3**(3), 207–223 (2021)



Modular ADMM-Based Strategies for Optimized Compression, Restoration, and Distributed Representations of Visual Data

Yehuda Dar and Alfred M. Bruckstein

Contents

Introduction	176
Modular ADMM-Based Optimization: General Construction and Guidelines	178
Unconstrained Lagrangian Optimizations via ADMM	178
Employing Black-Box Modules	180
Another Splitting Structure	181
Image Restoration Based on Denoising Modules	183
Modular Optimizations Based on Standard Compression Techniques	185
Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization	185
Restoration by Compression	189
Modular Strategies for Intricate Compression Problems	191
Distributed Representations Using Black-Box Modules	198
The General Framework	198
Modular Optimizations for Holographic Compression of Images	199
Conclusion	202
References	205

Abstract

Iterative techniques are a well-established tool in modern imaging sciences, allowing to address complex optimization problems via sequences of simpler computational processes. This approach has been significantly expanded in recent years by iterative designs where explicit solutions of optimization sub-problems were replaced by black-box applications of ready-to-use modules for

Y. Dar (✉)

Electrical and Computer Engineering Department, Rice University, Houston, TX, USA
e-mail: ydar@rice.edu

A. M. Bruckstein (✉)

Computer Science Department, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: freddy@cs.technion.ac.il

denoising or compression. These modular designs are conceptually simple, yet often achieve impressive results. In this chapter, we overview the concept of modular optimization for imaging problems by focusing on structures induced by the alternating direction method of multipliers (ADMM) technique and their applications to intricate restoration and compression problems. We start by emphasizing general guidelines independent of the module type used and only then derive ADMM-based structures relying on denoising and compression methods. The wide perspective on the topic should motivate extensions of the types of problems addressed and the kinds of black boxes utilized by the modular optimization. As an example for a promising research avenue, we present our recent framework employing black-box modules for distributed representations of visual data.

Keywords

Modular optimization · Alternating direction method of multipliers (ADMM) · Inverse problems · Signal compression · Distributed representations

Introduction

During the last several decades, significant attention and efforts were invested in establishing solutions for a wide variety of imaging problems. The proposed methods often rely on models and techniques adapted to visual signals and the relevant problem settings. Naturally, along the contemporary challenges and open questions of the field, there are excellent solutions to various fundamental problems that were extensively studied throughout the years. This situation suggests addressing currently open problems by exploring their relations to existing methods developed for basic tasks.

A lot of work has been devoted to fundamental problems such as denoising of a single image contaminated by additive white Gaussian noise and lossy compression of still images with respect to squared errors as quality assessment measures. Persistent and thorough studies of such basic problems (in their classical settings) led to excellent solutions that are believed to be nearly perfect (see, e.g., Chatterjee and Milanfar 2009). However, the techniques for many other imaging tasks are in various degrees of maturity that leave room for possibly considerable improvements. Examples for types of currently active research lines include jointly addressing multiple imaging tasks (Burger et al. 2018; Corona et al. 2019a,b; Dar et al. 2018a,b,c,d), restoration with uncertainty about the degradation operator (Lai et al. 2016; Bahat et al. 2017), image compression with respect to modern perceptual quality measures (Ballé et al. 2017; Laparra et al. 2017), and tasks (also fundamental ones such as denoising and compression) involving visual data beyond a single natural image (this includes video, hyperspectral, medical, etc.).

In this chapter, we overview a recent and fascinating approach for elegant utilization of existing knowledge and available imaging tools for complex problems of interest. The general idea is to define an optimization problem such that when

addressed using a specific iterative optimization technique, the resulting sequential algorithm calls for solving a subproblem corresponding to a fundamental task like denoising or compression. Then, the explicit solutions of the basic subproblem instances may be replaced by black-box applications of available methods, highly perfected over the years due to their prevalence and long-standing importance. Interestingly, the black-box modules utilized do not have to exactly match the formulation of the subproblems they replace, as long as they address the same fundamental task.

The main concept of *modular optimization strategies* described above was preceded by a line of optimization-based iterative algorithms including stages of *explicitly* solving regularized inverse problems, often associated with denoising or maximum a posteriori estimation tasks (see, e.g., Afonso et al. 2010; Zoran and Weiss 2011). Yet, actual employment of image denoisers as black boxes was explicitly proposed only later in the Plug-and-Play Priors framework (Venkatakrishnan et al. 2013; Sreehari et al. 2016), where the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) was used to form iterative structures based on denoising modules to solve inverse imaging problems (specifically, demonstrated by Venkatakrishnan et al. (2013) and Sreehari et al. (2016) for tomographic reconstruction based on the BM3D denoiser (Dabov et al. 2007)). The Plug-and-Play Priors framework (based on ADMM and denoisers) proved very useful to a variety of practical inverse problems (Dar et al. 2016b; Rond et al. 2016; Brifman et al. 2016; Chan et al. 2017; Buzzard et al. 2018; Kwan et al. 2018; Yazaki et al. 2019; Brifman et al. 2019; Ahmad et al. 2019), and its convergence was analyzed for several particular cases (Chan et al. 2017; Chan 2019). Another prominent approach based on denoising modules is the Regularization-by-Denoising (RED) framework (Romano et al. 2017; Hong et al. 2019; Brifman et al. 2019), proposing to regularize the basic problem using the black-box denoising function. Then an efficient sequential procedure based on iterative optimization techniques of ADMM or a fixed-point strategy is called upon, thereby clarifying that modular optimizations can be constructed not only based on ADMM. Other non-ADMM methods using denoisers for restoration or reconstruction problems were proposed based on FISTA for addressing nonlinear problems (Kamilov et al. 2017; Ahmad et al. 2019), primal-dual splitting (Ono 2017), backward projections (Tirer and Giryes 2018a,b, 2019), and ISTA for online updates (Sun et al. 2019a,b). All of these firmly established the wide applicability of denoising-based modular approaches for inverse problems addressing restoration and reconstruction of visual data.

We here consider the modular optimization strategy as a general concept beyond the extensively studied aspect of using denoisers for solving inverse problems. The deviation from the denoising-based modular optimizations started by Dar et al. (2016a, 2018c), and also the related work of Beygi et al. (2017a,b), where inverse problems were addressed based on compression techniques, essentially functioning as complexity regularizers. Specifically, image deblurring and inpainting problems were addressed by Dar et al. (2018c) using JPEG2000 and the state-of-the-art image coding method of the High Efficiency Video Coding (HEVC) standard. Moreover, a shift-invariant regularizer was proposed by Dar et al. (2018c) to amend the limitations of the regular compression-based prior. All these complex problem

structures were treated using the ADMM optimization tool in a Plug-and-Play manner.

Another important generalization is due to a recent research line (Dar et al. 2016a, 2018a,b,c,d), branching out from the original Plug-and-Plug Priors framework, suggesting to address intricate compression and restoration problems based on image and video compression modules applied as black boxes. Importantly, this framework shows that modularity is possible not only for priors and that the basic modules employed can be other than denoisers. Furthermore, using standard compression techniques in modular optimization frameworks extends the range of imaging problems addressed far outside the area of inverse problems. This extension is pursued in (Dar et al. 2018a,b,c,d) where systems involving acquisition, compression, and rendering processes are optimized based on ADMM and standard compression techniques. This established the ability to optimize complex systems while being compatible to prevalent compression standards and without using post-processing – thereby emphasizing on the usefulness of modular optimization strategies to much more than using denoisers as black-box priors or in ready-to-use modules. Specifically, the ADMM-based framework in (Dar et al. 2018a,b,c,d) also exhibits how to address intricate rate-distortion optimizations (a fundamental concept in modern compression techniques (Shoham and Gersho 1988; Ortega and Ramchandran 1998; Sullivan and Wiegand 1998)) by decoupling the challenging distortion metric from the actual compression task, consequently enabling the use of standard techniques as modules. Indeed, this idea inspired the nice work reported by Rott Shaham and Michaeli (2018) where an alternating minimization process is used to decouple a perceptual distortion metric from a standard compression technique – thus externally adding desired perceptual aspects into a standardized compression method.

We further point out here on a new direction of developing modular optimizations for distributed representations. In general, ADMM is a technique for distributed optimization, and, therefore, it is natural to utilize its valuable decoupling ability also for optimizations aimed at distributed representations. Specifically, we suggest to employ black-box modules for creating multiple descriptions of a given signal. Therefore, we overview our recent work (Dar and Bruckstein 2021) on holographic compression of images, where standard image compression techniques are adjusted to settings of duplication-based storage systems. The idea is to create a set of standard-compatible representations, all of them being equally important in refining the data reconstruction. We conclude by discussing the general implications of modular optimizations to distributed tasks.

Modular ADMM-Based Optimization: General Construction and Guidelines

Unconstrained Lagrangian Optimizations via ADMM

Consider an arbitrary optimization problem of an unconstrained Lagrangian form, namely,

$$\hat{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{M}}{\operatorname{argmin}} R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{v}) \quad (1)$$

to be solved for the optimization variable $\mathbf{v} \in \mathcal{M} \subset \mathbb{R}^N$, where \mathcal{M} is a (continuous or discrete) subset of the N -dimensional real space. Moreover, the optimization (1) is defined for a given column vector $\mathbf{x} \in \mathbb{R}^M$. In this section, we refer to general scalar-valued functions satisfying $R : \mathcal{M} \rightarrow \mathbb{R}$ and $D : \mathbb{R}^M \times \mathbb{R}^N \rightarrow \mathbb{R}$. In the sequel discussing the applications for restoration and compression tasks, the general definitions given here take the following form. For restoration tasks, posed as inverse problems, \mathcal{M} is set to be \mathbb{R}^N , and the functions R and D implement regularization and fidelity terms, respectively. In the case of compression, \mathcal{M} is a discrete set of decompressed signals supported by the compression architecture, and the functions R and D measure bit-cost and distortion, respectively. While restoration and compression problems introduce various mathematical forms to the general optimization (1), we here address this general structure.

Particular instances of the problem (1) often take challenging forms that require significant engineering and/or computational resources. Addressing a new problem may require the design and implementation of a complete algorithm from scratch, ignoring existing knowledge and tools from potentially related problems. Then, computational difficulties may arise due to high dimensionality of specific instances of (1) such that direct solutions become very costly or even impractical. Such reasons motivate the translation of (1) into a tractable procedure addressing the original task, sometimes in an approximated manner, while avoiding the complications mentioned above. A prominent approach for such designs is described next.

The alternating direction method of multipliers (ADMM) technique (Boyd et al. 2011) is a popular tool for addressing the potentially challenging problem (1). For this we start by splitting the optimization variable such that (1) becomes

$$\begin{aligned} \hat{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{M}, \mathbf{z} \in \mathbb{R}^N}{\operatorname{argmin}} & R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{z}) \\ & f \\ \text{subject to} & \mathbf{v} = \mathbf{z} \end{aligned} \quad (2)$$

where $\mathbf{z} \in \mathbb{R}^N$ is an auxiliary variable that is not directly constrained to the domain \mathcal{M} . Next, we apply the scaled form of the augmented Lagrangian and the method of multipliers (Boyd et al. 2011, Ch. 2) on (2) and obtain the iterative procedure

$$\left(\hat{\mathbf{v}}^{(t)}, \hat{\mathbf{z}}^{(t)} \right) = \underset{\mathbf{v} \in \mathcal{M}, \mathbf{z} \in \mathbb{R}^N}{\operatorname{argmin}} R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{z}) + \frac{\beta}{2} \left\| \mathbf{v} - \mathbf{z} + \mathbf{u}^{(t)} \right\|_2^2 \quad (3)$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)} \right), \quad (4)$$

where t denotes the iteration index, $\mathbf{u}^{(t)} \in \mathbb{R}^N$ is the scaled dual variable, and β is an auxiliary parameter introduced by the augmented Lagrangian. Then, the ADMM form of the problem is derived by applying one iteration of alternating minimization on (3), yielding a series of simpler optimizations:

$$\hat{\mathbf{v}}^{(t)} = \operatorname{argmin}_{\mathbf{v} \in \mathcal{M}} R(\mathbf{v}) + \frac{\beta}{2} \|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\|_2^2 \quad (5)$$

$$\hat{\mathbf{z}}^{(t)} = \operatorname{argmin}_{\mathbf{z} \in \mathbb{R}^N} \lambda D(\mathbf{x}, \mathbf{z}) + \frac{\beta}{2} \|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\|_2^2 \quad (6)$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)} \right) \quad (7)$$

where $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$ and $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$. Importantly, in the last ADMM-based structure, the possibly nontrivial domain \mathcal{M} and the related function R are decoupled from the second, perhaps intricate, function D . Accordingly, the new subtasks in the process are much simpler. Specifically, note that (6) is a continuous optimization problem over \mathbb{R}^N , regardless of the original domain of problem (1) that may be even discrete. Note that in the general case, where R , D , and \mathcal{M} can induce non-convexity and discreteness to the problem, there are no convergence guarantees corresponding to the ADMM process formulated above, and its usefulness should be evaluated empirically. However, this common practice has already provided many useful methods for various applications, and selected examples of those are presented in sections “[Image Restoration Based on Denoising Modules](#)” and “[Modular Optimizations Based on Standard Compression Techniques](#)”.

Employing Black-Box Modules

While the ADMM form in (5), (6), and (7) indeed seems easier to carry out than a complex instance of (1), the explicit definition and deployment of \mathcal{M} and/or R in the optimization stage (5) may still require some engineering efforts (such as design, implementation, etc.). In the case of restoration tasks, this means detailed definitions and implementations of regularization functions. For compression architectures, one should establish binary compressed representations matching signal-domain reconstructions. As explained next, the fundamental idea of using black-box modules is to avoid explicit treatment of such details and still achieve excellent, or even state-of-the-art, results with respect to the actual goal.

The main guideline when addressing a problem based on modular optimization strategies is to *formulate the initial optimization problem* (in our case, an instance of (1)) and *choose an iterative optimization technique* (here, ADMM) *such that the resulting sequential process includes:*

- A stage corresponding to a basic problem, having well-established solutions readily available to use. In the developments presented here, we ask to replace the optimization stage (5) with a module applied as a black box and, by that, encapsulating the various aspects of the original problem domain \mathcal{M} and function R . Now, if (5) can be identified as a prototype formulation corresponding to a

fundamental problem (e.g., denoising, compression), then one can replace the direct treatments of (5) with application of a module addressing the same basic problem – possibly based on another formulation or even an algorithm that does not correspond to an explicit mathematical expression. Such module is applied as a black box and denoted here as

$$\hat{\mathbf{v}}^{(t)} = \text{BlackBoxModule} \left(\tilde{\mathbf{z}}^{(t)}; \theta(\beta) \right) \quad (8)$$

where $\theta(\beta)$ (which will be denoted from now on as θ) is a parameter generalizing the role of the Lagrange multiplier β in determining the *implicit* trade-off between the components appeared in (5) before the replacement with the module. The generic method is summarized in Algorithm 1, where the number of parameters is reduced based on the relation $\tilde{\beta} \triangleq \frac{\beta}{2\lambda}$ such that only the parameters θ and $\tilde{\beta}$ are required as inputs for the method (for simplicity, we do not use the fact that both θ and $\tilde{\beta}$ originally depend on β).

- A subproblem considering the distance function D while having a form that can be practically solved. This refers here to subproblem (6). In many interesting applications, the distance function is a particular case of

$$D(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^K \alpha_j \|\mathbf{A}_j \mathbf{x} - \mathbf{B}_j \mathbf{z}\|_2^2 \quad (9)$$

for some positive integer K , positive real values $\{\alpha_j\}_{j=1}^K$, and matrices $\{\mathbf{A}_j\}_{j=1}^K \in \mathbb{R}^{\tilde{N} \times M}$, $\{\mathbf{B}_j\}_{j=1}^K \in \mathbb{R}^{\tilde{N} \times N}$. Then, for the form (9), the optimization step is a least squares problem that can be easily addressed for many structures of matrices $\{\mathbf{A}_j\}_{j=1}^K \in \mathbb{R}^{\tilde{N} \times M}$ and $\{\mathbf{B}_j\}_{j=1}^K \in \mathbb{R}^{\tilde{N} \times N}$.

One should note that the modular optimization process in Algorithm 1 provides a result that is an output of the black-box module applied in the last iteration. This eventual output can be the signal $\hat{\mathbf{v}}^{(t)} \in \mathcal{M}$ produced by the module at the last iteration and/or other relevant data possibly outputted by the module. This structure is useful, for example, in the case of compression where the important output is a binary compressed representation (i.e., a direct output of the module which is coupled with the signal $\hat{\mathbf{v}}^{(t)} \in \mathcal{M}$). Various applications may benefit from an alternative application that is described next.

Another Splitting Structure

We now turn to describe the construction of a process mirroring Algorithm 1 and utilized often for restoration and reconstruction problems. For the developments

Algorithm 1 General Modular Optimization – Type I: Overall Results Are Module Outputs

- 1: Inputs: $\mathbf{x}, \theta, \tilde{\beta}$.
 - 2: Initialize $t = 0, \hat{\mathbf{z}}^{(0)} = \mathbf{x}, \mathbf{u}^{(1)} = \mathbf{0}$.
 - 3: **repeat**
 - 4: $t \leftarrow t + 1$
 - 5: $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
 - 6: $\hat{\mathbf{v}}^{(t)} = \text{BlackBoxModule}(\tilde{\mathbf{z}}^{(t)}; \theta)$
 - 7: $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
 - 8: $\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z} \in \mathbb{R}^N}{\operatorname{argmin}} D(\mathbf{x}, \mathbf{z}) + \tilde{\beta} \|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\|_2^2$
 - 9: $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)})$
 - 10: **until** stopping criterion is satisfied
 - 11: Output: $\hat{\mathbf{v}}^{(t)}$ and/or other application-specific outputs of *BlackBoxModule*.
-

overviewed, here, we assume that the output domain of the basic optimization problem (1) satisfies $\mathcal{M} = \mathbb{R}^N$.

The alternative process stems from a delicate difference in the variable splitting applied on the basic problem, namely, instead of (2), we write

$$\begin{aligned} \hat{\mathbf{v}} = \underset{\mathbf{v} \in \mathbb{R}^N, \mathbf{z} \in \mathbb{R}^N}{\operatorname{argmin}} \quad & R(\mathbf{z}) + \lambda D(\mathbf{x}, \mathbf{v}) \\ \text{subject to} \quad & \mathbf{v} = \mathbf{z} \end{aligned} \quad (10)$$

where $\mathbf{z} \in \mathbb{R}^N$ is an auxiliary variable used here to replace the occurrence of \mathbf{v} as the argument of R , whereas the function D still refers to \mathbf{v} (note the difference from the variable splitting described in (2)). Then, similarly to section “[Unconstrained Lagrangian Optimizations via ADMM](#)”, further developing (10) using the scaled form of the augmented Lagrangian, the method of multipliers, and alternating minimization gives

$$\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v} \in \mathbb{R}^N}{\operatorname{argmin}} \lambda D(\mathbf{x}, \mathbf{v}) + \frac{\beta}{2} \|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\|_2^2 \quad (11)$$

$$\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z} \in \mathbb{R}^N}{\operatorname{argmin}} R(\mathbf{z}) + \frac{\beta}{2} \|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\|_2^2 \quad (12)$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}) \quad (13)$$

where $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$ and $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$. Note that the current procedure in (11), (12), and (13) includes the same subproblems as in (5), (6), and (7) but in a different order (and also up to the setting $\mathcal{M} = \mathbb{R}^N$ used in this subsection).

Like in section “[Employing Black-Box Modules](#)”, we identify the stage considering the function R , here in (12), as a solution to a fundamental problem that

can be replaced by an available black-box implementation. This yields the process described in Algorithm 2. Note that the result of the procedure is not a direct output of the black-box module. This delicate change with respect to Algorithm 1 may lead to improved results in various applications such as image restoration (where the black-box module is utilized for regularization purposes and, in practice, it is often better not to use its output directly as the result of the entire procedure).

Algorithm 2 General Modular Optimization – Type II: Overall Results Are Not Module Outputs

1: Inputs: \mathbf{x} , θ , $\tilde{\beta}$.
2: Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
3: **repeat**
4: $t \leftarrow t + 1$
5: $\hat{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
6: $\hat{\mathbf{v}}^{(t)} = \operatorname{argmin}_{\mathbf{v} \in \mathbb{R}^N} D(\mathbf{x}, \mathbf{v}) + \tilde{\beta} \|\mathbf{v} - \hat{\mathbf{z}}^{(t)}\|_2^2$
7: $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
8: $\hat{\mathbf{z}}^{(t)} = \text{BlackBoxModule}(\tilde{\mathbf{v}}^{(t)}; \theta)$
9: $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)})$
10: **until** stopping criterion is satisfied
11: Output: $\hat{\mathbf{v}}^{(t)}$.

Image Restoration Based on Denoising Modules

In the previous section, we presented the modular optimization approach in its general form, independent of the type of tasks addressed and modules utilized. In this section, we focus on the prevalent application of denoising-based modular optimizations to image restoration problems.

The problem setting considered in this section is defined as follows. A signal $\mathbf{v}_0 \in \mathbb{R}^N$ is going through a degradation process, resulting in the observation $\mathbf{x} \in \mathbb{R}^M$ satisfying

$$\mathbf{x} = \mathbf{H}\mathbf{v}_0 + \mathbf{n}, \quad (14)$$

where \mathbf{H} is a $M \times N$ real matrix and \mathbf{n} is a white Gaussian noise column-vector of length M (the noise components are zero mean and have variance σ_n^2). The restoration task is to estimate the unknown \mathbf{v}_0 , given \mathbf{x} and the knowledge of the degradation operator \mathbf{H} and the noise variance σ_n^2 . For the purpose of restoration, we define the function D from (1) as the fidelity term of the respective inverse problem, namely,

$$D(\mathbf{x}, \mathbf{v}) = \|\mathbf{x} - \mathbf{H}\mathbf{v}\|_2^2. \quad (15)$$

The ADMM optimization structure based on black-box denoisers, first proposed in the Plug-and-Play Priors design (Venkatakrishnan et al. 2013; Sreehari et al. 2016), mainly stems from associating the function $R : \mathbb{R}^N \rightarrow \mathbb{R}$ with a regularizer implemented (explicitly or implicitly) in a ready-to-use denoising process. Then, the optimization for the $\hat{\mathbf{z}}^{(t)}$, appearing in (12), can be interpreted as an inverse problem for denoising $\tilde{\mathbf{v}}^{(t)}$ using the regularizer R . One can also perceive (12) as a maximum a posteriori (MAP) estimation of a signal from its noisy version $\tilde{\mathbf{v}}^{(t)}$, i.e.,

$$\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z} \in \mathbb{R}^N}{\operatorname{argmax}} \log p_R(\mathbf{z}) + \log p_\eta(\tilde{\mathbf{v}}^{(t)} - \mathbf{z}) \quad (16)$$

where $p_R(\mathbf{z}) \triangleq \exp(-R(\mathbf{z}))$ is the prior probability function assumed for the clean signal and p_η is the probability density function of an additive Gaussian noise vector $\boldsymbol{\eta}$ with i.i.d. components having zero mean and $1/\beta$ variance. Accordingly, the correspondence of (12) to denoising problems motivates the usage of black-box denoisers as the modules applied at stage 8 of Algorithm 2. These denoisers should be set to remove noise having variance of $1/\beta$ from the signal $\tilde{\mathbf{v}}^{(t)}$. Importantly, the substitution of (12) with applications of Gaussian denoisers was experimentally shown beneficial also for denoisers that do not follow the MAP estimation form or the regularized inverse problem approach. Specifically, one can even employ algorithmic denoisers that were designed based on completely different mindsets. The denoising-based restoration procedure for an arbitrary degradation operator \mathbf{H} is summarized in Algorithm 3.

The decoupling induced by the ADMM structure leads to an additional conceptual simplification: stage 6 of Algorithm 3 can be interpreted as a ℓ_2 -constrained deconvolution problem (or ℓ_2 -regularized least squares computation) with respect to the degradation operator \mathbf{H} . Note that this is one of the simplest restoration formulations addressing the degradation process (14) from the regularized inverse-problem perspective. The corresponding analytic solution is

$$\hat{\mathbf{v}}^{(t)} = \left(\mathbf{H}^T \mathbf{H} + \tilde{\beta} \mathbf{I} \right)^{-1} \left(\mathbf{H}^T \mathbf{x} + \tilde{\beta} \tilde{\mathbf{z}}^{(t)} \right). \quad (17)$$

Alternatively, this computation can be numerically applied in various tractable ways (depending on the specific structure of \mathbf{H}). In summary, the overall modular restoration process relies on sequential application of conceptually simple tasks: Gaussian denoising and ℓ_2 -constrained deconvolution.

Figures 1 and 2 show typical results obtained using the Plug-and-Play method, implemented in the code published with Chan et al. (2017), based on the BM3D denoiser (Dabov et al. 2007). The deblurring settings (Fig. 1) include a blur operator corresponding to a 9×9 pixels convolution kernel (Gaussian with standard deviation 1.75) and additive white Gaussian noise of standard deviation 10. The inpainting experiment (Fig. 2) considers 80% missing pixels and additive white Gaussian noise of standard deviation 10. Specifically, note the improvement in the PSNR of the intermediate estimates, $\hat{\mathbf{v}}^{(t)}$, evolving throughout the process iterations until

Algorithm 3 Restoration Based on Denoising Modules

```

1: Inputs:  $\mathbf{x}, \theta, \tilde{\beta}$ .
2: Initialize  $t = 0, \hat{\mathbf{z}}^{(0)} = \mathbf{x}, \mathbf{u}^{(1)} = \mathbf{0}$ .
3: repeat
4:    $t \leftarrow t + 1$ 
5:    $\hat{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$ 
6:    $\hat{\mathbf{v}}^{(t)} = \operatorname{argmin}_{\mathbf{v} \in \mathbb{R}^N} \|\mathbf{x} - \mathbf{H}\mathbf{v}\|_2^2 + \tilde{\beta} \|\mathbf{v} - \hat{\mathbf{z}}^{(t)}\|_2^2$ 
7:    $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$ 
8:    $\hat{\mathbf{z}}^{(t)} = \text{Denoiser}(\tilde{\mathbf{v}}^{(t)}; \theta)$ 
9:    $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)})$ 
10: until stopping criterion is satisfied
11: Output:  $\hat{\mathbf{v}}^{(t)}$ .

```

practical convergence (Figs. 1d and 2d). See Chan et al. (2017) for details and analysis of the convergence appearing here.

Modular Optimizations Based on Standard Compression Techniques

In this section, we overview the utilization of compression modules for restoration and challenging compression purposes. The use of modules beyond denoisers further establishes the modularity property as a general idea, relevant to various tasks.

Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization

Consider a signal, $\mathbf{x} \in \mathbb{R}^N$, to be compressed and represented as a sequence of bits. We describe a lossy compression procedure as the function

$$C : \mathbb{R}^N \rightarrow \mathcal{B}, \quad (18)$$

mapping the N -dimensional signal domain to a discrete set \mathcal{B} of compressed representations in variable-length binary forms. The compression of \mathbf{x} is

$$\mathbf{b} = C(\mathbf{x}), \quad (19)$$

where $\mathbf{b} \in \mathcal{B}$ is the binary compressed data useful for storage or transmission. Then, a matching decompression process gets the compressed data \mathbf{b} as its input and reconstructs a signal-domain representation via

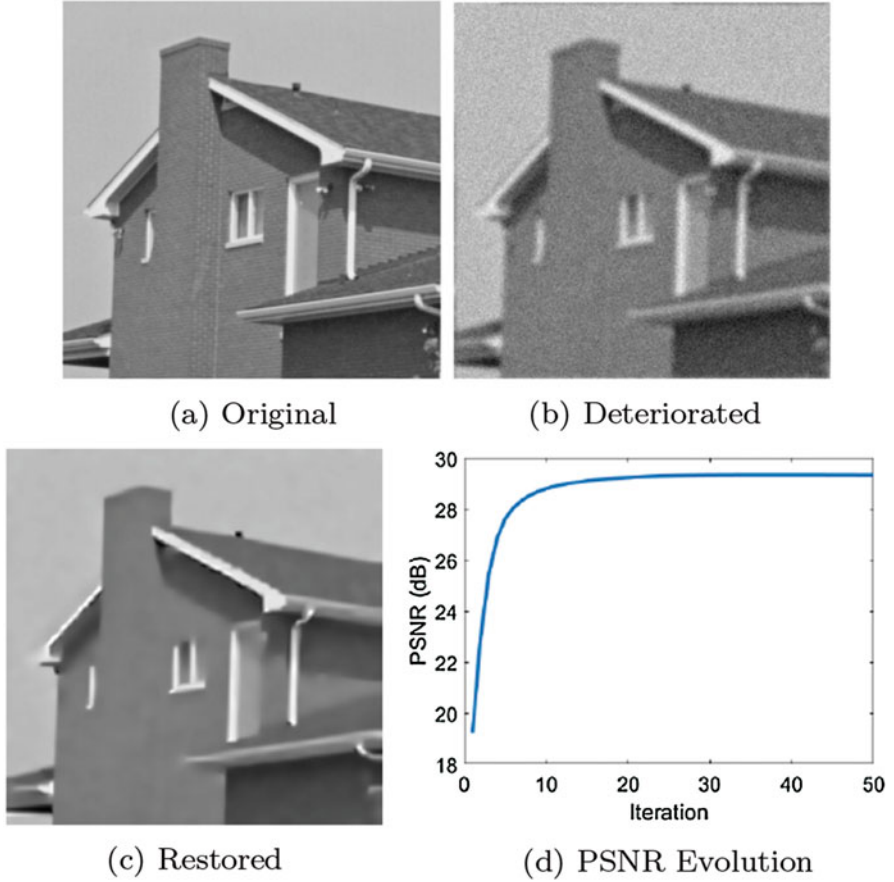


Fig. 1 Deblurring using denoising-based Plug-and-Play method (Chan et al. 2017). The utilized denoiser is BM3D (Dabov et al. 2007). The degradation includes a Gaussian blur (of 9×9 pixels kernel and 1.75 standard deviation), followed by additive noise with $\sigma_n = 10$, applied on the House image (256×256 pixels). (a) The original image. (b) Deteriorated image. (c) Restored image using the method by Chan et al. (2017) (29.33 dB). (d) The PSNR evolution of the intermediate estimate $\hat{\mathbf{v}}^{(t)}$ along the restoration-process iterations

$$\mathbf{v} = F(\mathbf{b}), \quad (20)$$

where

$$F: \mathcal{B} \rightarrow \mathcal{S} \quad (21)$$

maps the binary compressed representations in \mathcal{B} to their corresponding decompressed signals from the discrete set $\mathcal{S} \subset \mathbb{R}^N$. The decompressed signal \mathbf{v} can be

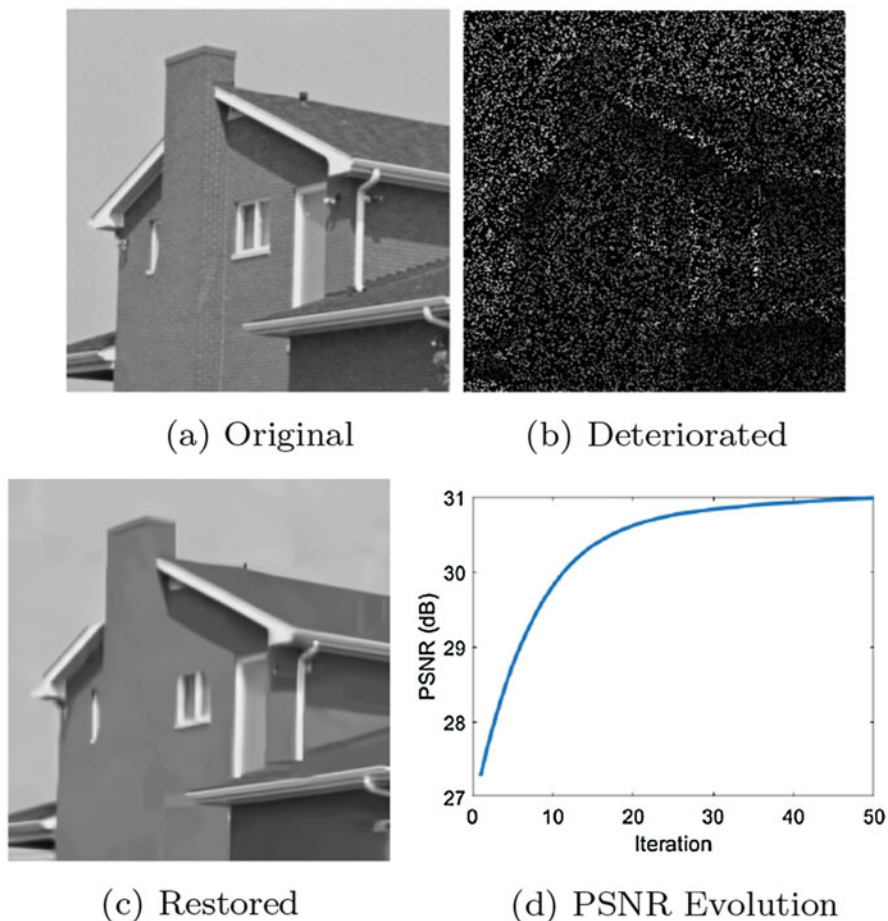


Fig. 2 Inpainting using denoising-based Plug-and-Play method (Chan et al. 2017). The employed denoiser is BM3D (Dabov et al. 2007). The degradation includes 80% missing pixels and additive noise with $\sigma_n = 10$, applied on the House image (256×256 pixels). (a) The original image. (b) Deteriorated image. (c) Restored image using the method by Chan et al. (2017) (30.98 dB). (d) The PSNR evolution of the intermediate estimate $\hat{v}^{(t)}$ along the iterations

further processed or outputted to a user. For example, in the case of visual signals, \mathbf{v} is usually displayed.

Modern compression architectures (see, e.g., Ortega and Ramchandran 1998; Sullivan and Wiegand 1998; Shukla et al. 2005; Sullivan et al. 2012) implement the compression function C using operational rate-distortion optimizations, a tool established by Shoham and Gersho (1988), Chou et al. (1989), and Ortega and Ramchandran (1998), and can be explained using our notions as follows. A given deterministic signal \mathbf{x} is compressed based on an optimization process searching for its best compressed representation $\mathbf{b} \in \mathcal{B}$, coupled with the decompressed signal

$\mathbf{v} \in \mathcal{S}$. The optimization trades off two opposing aspects of the representation: bit-cost and reconstruction quality. The bit-cost of the binary representation $\mathbf{b} \in \mathcal{B}$ is its length. Since, by (20), each $\mathbf{b} \in \mathcal{B}$ corresponds to one decompressed signal $\mathbf{v} \in \mathcal{S}$, we define the bit-cost of a decompressed signal $\mathbf{v} \in \mathcal{S}$ as the length of its binary representation $\mathbf{b} = F^{-1}(\mathbf{v})$. We also define the function $R_{\mathcal{S}}(\mathbf{v})$ to evaluate the bit-cost of the compressed binary representation associated with \mathbf{v} . Specifically, for $\mathbf{v} \in \mathcal{S}$ that satisfies $\mathbf{v} = F(\mathbf{b})$, the bit-cost is

$$R_{\mathcal{S}}(\mathbf{v}) \triangleq \text{length}\{\mathbf{b}\}, \quad (22)$$

where $\text{length}\{\cdot\}$ counts the length of a binary description. The second part of the trade-off is the reconstruction distortion, $D(\mathbf{x}, \mathbf{v})$, evaluating the distance between the compression input \mathbf{x} and its decompressed form \mathbf{v} . Note that the distortion value is real and nonnegative. Then, the optimization task including bit-cost constraints, corresponding to storage space or transmission bandwidth limitations, is

$$\begin{aligned} \hat{\mathbf{v}} &= \underset{\mathbf{v} \in \mathcal{S}}{\text{argmin}} D(\mathbf{x}, \mathbf{v}) \\ \text{subject to} \quad & R_{\mathcal{S}}(\mathbf{v}) \leq r \end{aligned} \quad (23)$$

where $r \geq 0$ is the maximal representation bit-cost allowed. Another relevant optimization problem, mirroring (23), is defined to minimize the compression bit-cost under a limited distortion amount, i.e.,

$$\begin{aligned} \hat{\mathbf{v}} &= \underset{\mathbf{v} \in \mathcal{S}}{\text{argmin}} R_{\mathcal{S}}(\mathbf{v}) \\ \text{subject to} \quad & D(\mathbf{x}, \mathbf{v}) \leq d \end{aligned} \quad (24)$$

where $d \geq 0$ is the tolerated distortion level. Without loss of generality, we consider here the optimization form in (24). The constrained optimization (24) is usually cast (see, e.g., Shoham and Gersho 1988, Chou et al. 1989, Ortega and Ramchandran 1998, Sullivan and Wiegand 1998, Shukla et al. 2005, and Sullivan et al. 2012) to its unconstrained Lagrangian form

$$\hat{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{S}}{\text{argmin}} R_{\mathcal{S}}(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{v}) \quad (25)$$

where $\lambda \geq 0$ is a Lagrange multiplier corresponding to a distortion constraint $d_{\lambda} \geq 0$. Such compression without a prespecified distortion level is common, e.g., in video coding (Sullivan et al. 2012).

When working with high-dimensional signals (large N values), the discrete set \mathcal{S} tends to be huge. Then, for arbitrarily structured distortion metrics $D(\mathbf{x}, \mathbf{v})$, one cannot directly solve the Lagrangian form in (25) via iterating over the elements in \mathcal{S} and evaluating their corresponding costs (recall that (25) is a discrete

optimization problem). Accordingly, compression methods are designed such that the combination of $D(\mathbf{x}, \mathbf{v})$, \mathcal{S} , and \mathcal{B} leads to a computationally tractable task. This is often obtained using architectures where nonoverlapping signal segments are independently compressed with respect to the squared-error distortion measure (see details in the Appendix). However, while such computationally efficient architectures that rely on squared-error metrics are prevalent (we also refer them as *standard* compression techniques), they are often too simple and limit the compression performance one could wish for in various settings of interest. This will be further demonstrated in section “[Modular Strategies for Intricate Compression Problems](#)”.

Restoration by Compression

Regularization of inverse problems based on complexity measures is a well-established approach for estimation tasks (see, e.g., Rissanen 2000). In a subclass of these methods, complexity is defined based on the number of bits required for the compressed representation of the candidate estimate. This motivated various studies of signal and image denoising using lossy compression techniques (see, for example, Natarajan 1995 and Liu and Moulin 2001). The extension of this idea to image restoration problems beyond Gaussian denoising was studied from a theoretical perspective by Moulin and Liu (2000), also including a limited experimental demonstration for Poisson denoising based on a particularly designed compression process. Implementing the compression-based approach for other image restoration problems (such as deblurring, inpainting, super resolution, etc.) was considered as impractical for a long while until the *Restoration by Compression* architecture (Dar et al. 2016a, 2018c) resolved the computational difficulties via ADMM-based modularity. Next, we overview the main construction and applicative aspects of the *Restoration by Compression* idea as a special case of the generic modular optimization designs presented above.

The core idea in the *Restoration by Compression* approach (Dar et al. 2016a, 2018c) is to exploit existing compression techniques such that their underlying signal models will be indirectly used for desired restoration purposes. For this, we define the function R in (1) as a complexity regularizer, measuring the likelihood of a signal based on its compression bit-cost (assuming that more probable signals receive shorter compressed representations). Specifically, the regularizer extends the bit-cost evaluation function (22) such that for any $\mathbf{z} \in \mathbb{R}^N$ it returns

$$R(\mathbf{z}) = \begin{cases} R_{\mathcal{S}}(\mathbf{z}) & \text{for } \mathbf{z} \in \mathcal{S} \\ \infty & \text{for } \mathbf{z} \notin \mathcal{S} \end{cases}, \quad (26)$$

where \mathcal{S} and $R_{\mathcal{S}}$ are conceptually associated with an existing compression technique. Then, considering the complexity regularizer (26), the optimization for the $\hat{\mathbf{z}}^{(t)}$ in (12) becomes equivalent to an operational rate-distortion optimization

(25) with respect to the implicit architecture of the ready-to-use compression method. This motivates the replacement of (12) with an application of a black-box compression module, followed by its respective decompression process, i.e.,

$$\mathbf{b}^{(t)} = \text{StandardCompression}(\tilde{\mathbf{v}}^{(t)}; \theta) \quad (27)$$

$$\hat{\mathbf{z}}^{(t)} = \text{StandardDecompression}(\mathbf{b}^{(t)}). \quad (28)$$

Note that the application of both compression and decompression is in accordance with the optimization form in (25) that looks for the optimal decompressed signal corresponding to the given signal to compress. Interestingly, the utilized compression modules do not have to rely on rate-distortion optimizations (25), as in the case of transform coding architectures (such as the JPEG2000 method included in the following demonstrations). The Restoration by Compression procedure is summarized in Algorithm 4. See Dar et al. (2018c) for further detail on the parameters given to the compression modules in the iterative process. Moreover, the main concepts of the proposed algorithm are explained by Dar et al. (2018c) using rate-distortion theory for cyclo-stationary Gaussian signals.

Algorithm 4 Restoration by Compression: Basic Complexity Regularization

- 1: Inputs: \mathbf{x} , θ , $\tilde{\beta}$.
 - 2: Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
 - 3: **repeat**
 - 4: $t \leftarrow t + 1$
 - 5: $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
 - 6: $\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v} \in \mathbb{R}^N}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{H}\mathbf{v}\|_2^2 + \tilde{\beta} \|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\|_2^2$
 - 7: $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
 - 8: $\mathbf{b}^{(t)} = \text{StandardCompression}(\tilde{\mathbf{v}}^{(t)}; \theta)$
 - 9: $\hat{\mathbf{z}}^{(t)} = \text{StandardDecompression}(\mathbf{b}^{(t)})$
 - 10: $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)})$
 - 11: **until** stopping criterion is satisfied
 - 12: Output: $\hat{\mathbf{v}}^{(t)}$.
-

Clearly, the artifacts introduced by the compression module participating in restoration process affect the produced estimate. Many compression artifacts originate in the common approach of independently coding nonoverlapping segments of the image. This block-based design also influences the complexity measure defining the regularizer in (26), essentially equivalent to summing the compression bit-costs of all the nonoverlapping blocks. This aspect was identified by Dar et al. (2018c) as introducing shift sensitivity into the regularizer (26). Accordingly, a shift-invariant complexity regularizer was proposed by Dar et al. (2018c), measuring the total bit-cost of *all the shifted versions* of the estimate evaluated. The shift operator $\text{shift}_j\{\cdot\}$

can be defined as a two-dimensional cyclical shift on an image or, alternatively, as returning the rectangular portion of the image that starts at a shifted coordinate from the upper-left corner pixel of the full image (see Dar et al. 2018c for details). For each $j \in \{1, \dots, N_b\}$, the two-dimensional offset applied by $shift_j\{\cdot\}$ is different. This leads to an extended Restoration by Compression process, described in Algorithm 5, including ADMM-based decoupling of the compressions of various shifts of the processed signals. Further details on the shift-invariant regularizer and the algorithm development are provided in Dar et al. (2018c). The applications of Algorithm 5 for deblurring and inpainting of images are presented in Figs. 3 and 4, respectively. The compression modules employed are JPEG2000 and HEVC (in its BPG implementation for image coding (Bellard)). Since HEVC provides significantly better compression performance than JPEG2000, a corresponding gap in their restoration abilities is also evident.

Algorithm 5 Restoration by Compression: Shift-Invariant Complexity Regularization

- 1: Inputs: \mathbf{y} , θ , $\tilde{\beta}$, and the number of shifts N_b .
 - 2: Initialize $\{\hat{\mathbf{z}}^{j,(0)}\}_{j=1}^{N_b}$ (depending on the deterioration type).
 - 3: $t = 1$ and $\mathbf{u}^{j,(1)} = \mathbf{0}$ for $j = 1, \dots, N_b$.
 - 4: **repeat**
 - 5: $\tilde{\mathbf{z}}^{j,(t)} = \hat{\mathbf{z}}^{j,(t-1)} - \mathbf{u}^{j,(t)}$ for $j = 1, \dots, N_b$
 - 6: Solve the ℓ_2 -constrained deconvolution:

$$\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v} \in \mathbb{R}^N}{\operatorname{argmin}} \left\| \mathbf{x} - \mathbf{H}\mathbf{v} \right\|_2^2 + \tilde{\beta} \sum_{j=1}^{N_b} \left\| \mathbf{v} - \tilde{\mathbf{z}}^{j,(t)} \right\|_2^2$$
 - 7: **for** $j = 1, \dots, N_b$ **do**
 - 8: $\tilde{\mathbf{v}}_{shifted}^{j,(t)} = shift_j \left\{ \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{j,(t)} \right\}$
 - 9: $\mathbf{b}^{j,(t)} = \text{StandardCompression} \left(\tilde{\mathbf{v}}_{shifted}^{j,(t)}; \theta \right)$
 - 10: $\hat{\mathbf{z}}_{shifted}^{j,(t)} = \text{StandardDecompression} \left(\mathbf{b}^{j,(t)} \right)$
 - 11: $\hat{\mathbf{z}}^{j,(t)} = shift_j^{-1} \left\{ \hat{\mathbf{z}}_{shifted}^{j,(t)} \right\}$
 - 12: $\mathbf{u}^{j,(t+1)} = \mathbf{u}^{j,(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{j,(t)} \right)$
 - 13: **end for**
 - 14: $t \leftarrow t + 1$
 - 15: **until** stopping criterion is satisfied
 - 16: Output: $\hat{\mathbf{v}}^{(t)}$.
-

Modular Strategies for Intricate Compression Problems

The utilization of available compression methods in modular restoration processes (Algorithms 4 and 5) naturally raises the question whether modular optimization strategies are relevant also to intricate compression problems. This is indeed the case, as established by the *System-Aware Compression* framework



Fig. 3 The deblurring experiment (settings #2 in Dar et al. 2018c) for the Cameraman image (256×256 pixels). (a) The underlying image. (b) Degraded image (20.76 dB). (c) Restored image using Algorithm 5 with JPEG2000 compression (28.10 dB). (d) Restored image using Algorithm 5 with HEVC compression (30.14 dB)

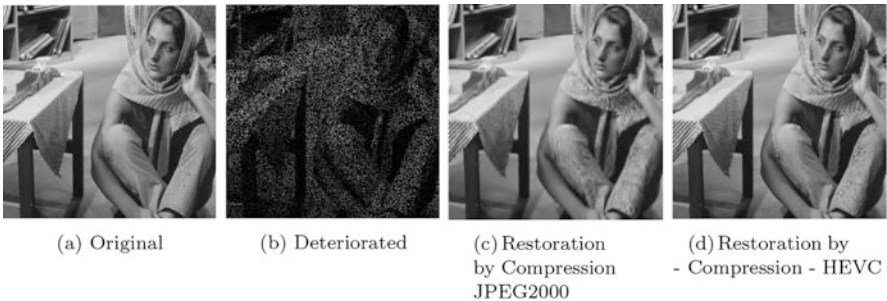


Fig. 4 The inpainting experiment (80% missing pixels) for the Barbara image (512×512 pixels). (a) The original image. (b) Deteriorated image. (c) Restored image using Algorithm 5 with JPEG2000 compression (24.83 dB). (d) Restored image using Algorithm 5 with HEVC compression (28.80 dB)

(Dar et al. 2018a,b,d), where ADMM-based modular strategies are employed for optimizing end-to-end performance of systems involving acquisition, compression, and rendering stages. The main idea is to decouple unusual distortion metrics from the actual compression tasks that, in turn, can be applied using black-box compression modules (which are operated with respect to the elementary squared-error metric). Hence, this methodology opens a new research path for addressing complex compression problems including, for example, optimizations for nonlocal processing/prediction architectures, enhancement filters or degradation processes, and perceptual metrics assessing subjective quality of audio/visual signals. Indeed, a successful implementation of this approach for perceptually oriented image compression (using an alternating minimization procedure) was proposed by Rott Shaham and Michaeli (2018).

In this section, we overview the *System-Aware Compression* concept (Dar et al. 2018a,b,d), demonstrating the main aspects of using modular optimizations for intricate compression problems. The motivation for the *System-Aware Compression*

framework stems from a structure common to many imaging systems (see Fig. 5), where a source signal is first acquired, then compressed for its storage or transmission, and eventually decompressed and rendered back into a signal that can be displayed or further processed. Obviously, in such systems, the quality of the eventual output depends on the entire acquisition-rendering chain and not solely on the lossy compression component. Yet, the employed compression technique is often system independent, hence inducing suboptimal rate-distortion performance for the entire system. The *System-Aware Compression* architecture is a practical and modular way for optimizing the end-to-end performance (in its rate-distortion trade-off sense) of such acquisition-rendering systems.

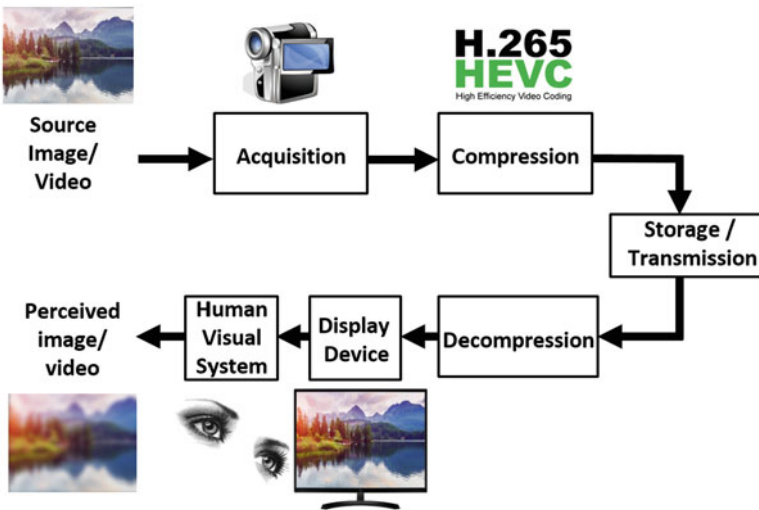


Fig. 5 The general imaging system structure motivating the System-Aware Compression approach

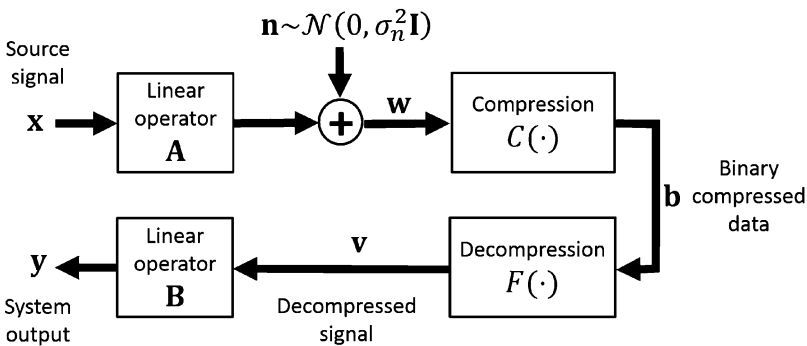


Fig. 6 The system model addressed by the System-Aware Compression framework

Let us describe the system structure considered for the mathematical development of the method (Fig. 6). A source signal, an N -length column vector $\mathbf{x} \in \mathbb{R}^N$, undergoes a linear processing represented by the $M \times N$ matrix \mathbf{A} and, then, deteriorated by an additive white Gaussian noise vector $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I})$, resulting in the signal

$$\mathbf{w} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad (29)$$

where \mathbf{w} and \mathbf{n} are M -length column vectors. We represent the lossy compression procedure via the mapping $C : \mathbb{R}^M \rightarrow \mathcal{B}$ from the M -dimensional signal domain to a discrete set \mathcal{B} of binary compressed representations (which may have different lengths). The signal \mathbf{w} is the input to the compression component of the system, producing the compressed binary data $\mathbf{b} = C(\mathbf{w})$ that can be stored or transmitted in an error-free manner. Then, on a device and settings depending on the specific application, the compressed data $\mathbf{b} \in \mathcal{B}$ is decompressed to provide the signal $\mathbf{v} = F(\mathbf{b})$ where $F : \mathcal{B} \rightarrow \mathcal{S}$ represents the decompression mapping between the binary compressed representations in \mathcal{B} to the corresponding decompressed signals in the discrete set $\mathcal{S} \subset \mathbb{R}^M$. The decompressed signal \mathbf{v} is further processed by the linear operator denoted as the $N \times M$ matrix \mathbf{B} , resulting in the system output signal

$$\mathbf{y} = \mathbf{B}\mathbf{v}, \quad (30)$$

which is an N -length real-valued column vector.

As an example, consider an acquisition-compression-rendering system where the signal \mathbf{w} is a sampled version of the source signal \mathbf{x} and the system output \mathbf{y} is the rendered version of the decompressed signal \mathbf{v} .

We assume here that the operators \mathbf{A} and \mathbf{B} , as well as the noise variance σ_n^2 , are known and fixed (i.e., cannot be optimized). Consequently, we formulate a new compression procedure in order to optimize the end-to-end rate-distortion performance of the entire system. Specifically, we want the system output \mathbf{y} to be the best approximation of the source signal \mathbf{x} under the bit-budget constraint. However, at the compression stage, we do not accurately know \mathbf{x} but rather its degraded form \mathbf{w} formulated in (29). This motivates us to suggest the following distortion metric with respect to the system output \mathbf{y}

$$D_s(\mathbf{w}, \mathbf{y}) = \frac{1}{M} \|\mathbf{w} - \mathbf{A}\mathbf{y}\|_2^2. \quad (31)$$

This metric conforms with the fact that if \mathbf{y} is close to \mathbf{x} , then, by (29), \mathbf{w} will be close to $\mathbf{A}\mathbf{y}$ up to the noise \mathbf{n} . Indeed, for the ideal case of $\mathbf{y} = \mathbf{x}$, the metric (31) becomes

$$D_s(\mathbf{w}, \mathbf{x}) = \frac{1}{M} \|\mathbf{n}\|_2^2 \approx \sigma_n^2 \quad (32)$$

where the last approximate equality is under the assumption of a sufficiently large M (the length of \mathbf{n}). Since $\mathbf{y} = \mathbf{B}\mathbf{v}$, we can rewrite the distortion $D_s(\mathbf{w}, \mathbf{y})$ in (31) as a function of the decompressed signal \mathbf{v} , namely,

$$D_c(\mathbf{w}, \mathbf{v}) = \frac{1}{M} \|\mathbf{w} - \mathbf{A}\mathbf{B}\mathbf{v}\|_2^2. \quad (33)$$

Since the operator \mathbf{B} produces the output signal \mathbf{y} , an ideal result will be $\mathbf{y} = \mathbf{P}_B\mathbf{x}$, where \mathbf{P}_B is the matrix projecting onto \mathbf{B} 's range. The corresponding ideal distortion is

$$d_0 \triangleq D_s(\mathbf{w}, \mathbf{P}_B\mathbf{x}) = \frac{1}{M} \|\mathbf{A}(\mathbf{I} - \mathbf{P}_B)\mathbf{x} + \mathbf{n}\|_2^2. \quad (34)$$

We use the distortion metric (33) to constrain the bit-cost minimization in the following rate-distortion optimization

$$\begin{aligned} \hat{\mathbf{v}} &= \underset{\mathbf{v} \in \mathcal{S}}{\operatorname{argmin}} R(\mathbf{v}) \\ \text{subject to} \quad d_0 &\leq \frac{1}{M} \|\mathbf{w} - \mathbf{A}\mathbf{B}\mathbf{v}\|_2^2 \leq d_0 + d \end{aligned} \quad (35)$$

where $R(\mathbf{v})$ evaluates the length of the binary compressed description of the decompressed signal \mathbf{v} and $d \geq 0$ determines the allowed distortion. By (34), the value d_0 depends on the operator \mathbf{A} , the null space of \mathbf{B} , the source signal \mathbf{x} , and the noise realization \mathbf{n} . Since \mathbf{x} and \mathbf{n} are unknown, d_0 cannot be accurately calculated in the operational case (in Dar et al. (2018d) we formulate the expected value of d_0 for the case of a cyclo-stationary Gaussian source signal). We address the optimization (35) using its unconstrained Lagrangian form

$$\hat{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{S}}{\operatorname{argmin}} R(\mathbf{v}) + \lambda \frac{1}{M} \|\mathbf{w} - \mathbf{A}\mathbf{B}\mathbf{v}\|_2^2 \quad (36)$$

where $\lambda \geq 0$ is a Lagrange multiplier corresponding to some distortion constraint $d_\lambda \geq d_0$ (such optimization strategy with respect to some Lagrange multiplier is common, e.g., in video coding (Sullivan et al. 2012)). In the case of high-dimensional signals, the discrete set \mathcal{S} is extremely large, and, therefore, it is impractical to directly solve the Lagrangian form in (36) for generally structured matrices \mathbf{A} and \mathbf{B} . This difficulty vanishes, for example, when $\mathbf{A} = \mathbf{B} = \mathbf{I}$, reducing the Lagrangian optimization in (36) to the standard (system independent) compression form (see, e.g., Shoham and Gersho 1988 and Ortega and Ramchandran 1998).

The optimization (36) matches the generic template presented in section “[Modular ADMM-Based Optimization: General Construction and Guidelines](#)”, and, therefore, we can formulate an ADMM-based modular procedure to address it. This modular optimization process is a special case of the generic procedure

described in Algorithm 1, taking here the form of Algorithm 6. Note that we use the form of Algorithm 1 where the eventual output is the output of the module applied in the last iteration, which in our case corresponds to the output of the compression module in the last iteration (and this is the desired output because in this section we consider compression application, unlike the Restoration by Compression method in Algorithm 4 that its purpose is restoration by means of compression-based regularization). The interested reader is referred to Dar et al. (2018d) for a rate-distortion theoretic analysis for cyclo-stationary Gaussian signals and linear shift-invariant operators, explaining various aspects of the proposed procedure.

Algorithm 6 System-Aware Compression

- 1: Inputs: $\mathbf{w}, \theta, \tilde{\beta}$.
 - 2: Initialize $t = 0, \hat{\mathbf{z}}^{(0)} = \mathbf{w}, \mathbf{u}^{(1)} = \mathbf{0}$.
 - 3: **repeat**
 - 4: $t \leftarrow t + 1$
 - 5: $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
 - 6: $\mathbf{b}^{(t)} = \text{StandardCompression}(\tilde{\mathbf{z}}^{(t)}, \theta)$
 - 7: $\hat{\mathbf{v}}^{(t)} = \text{StandardDecompression}(\mathbf{b}^{(t)})$
 - 8: $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
 - 9: $\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z} \in \mathbb{R}^N}{\text{argmin}} \|\mathbf{w} - \mathbf{A}\mathbf{B}\mathbf{z}\|_2^2 + \tilde{\beta} \|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\|_2^2$
 - 10: $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + (\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)})$
 - 11: **until** stopping criterion is satisfied
 - 12: Output: $\mathbf{b}^{(t)}$, which is the binary compressed data obtained in the last iteration.
-

To demonstrate the essence of the System-Aware Compression approach, we provide here a representative example taken from Dar et al. (2018b), excluding the acquisition stage (i.e., $\mathbf{A} = \mathbf{I}$ and $\sigma_n^2 = 0$) while considering a post-decompression operator \mathbf{B} implementing a shift-invariant Gaussian blur degradation. One can perceive this setting as optimizing image compression with respect to degradation occurring later (after decompression) at the display device, where no additional processing is done after decompression in order to counterbalance the degradation. To observe the gains achieved by the modular optimization approach, let us first examine the unoptimized (regular) compression process where the input image (Fig. 7a) is compressed using the state-of-the-art HEVC standard at a bit-rate of 3.75 bits per pixel (bpp), yielding the decompressed image in Fig. 7b (this is the image before blur degradation). Then, the decompressed image after degradation (Fig. 7c) is very blurry, as also reflected in the respective PSNR value (measured with respect to the image before compression). In the modular optimization approach, the input image is processed such that the compression in the last iteration is

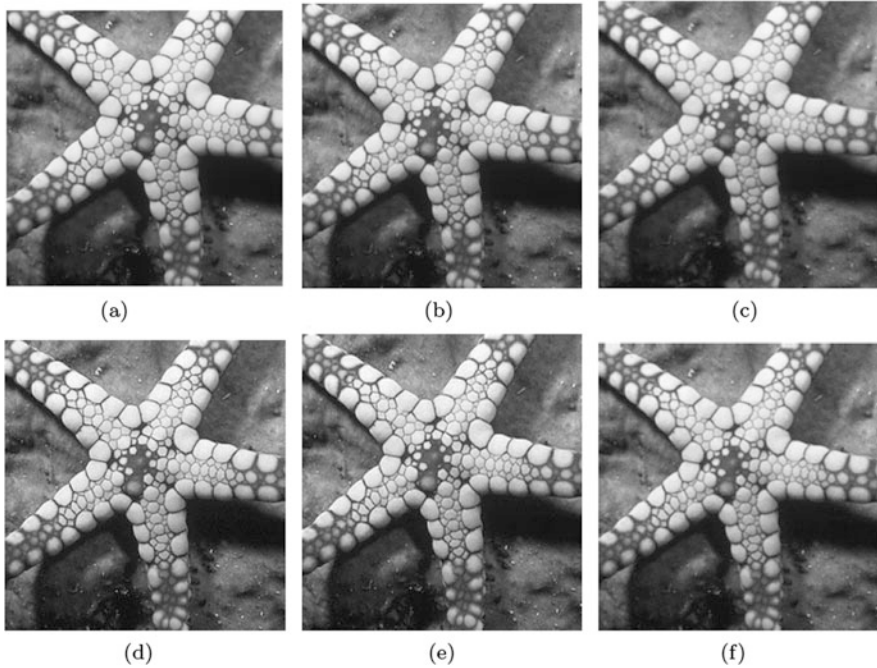


Fig. 7 Comparison of the System-Aware Compression approach and regular compression. The settings consider a Gaussian blur operator degrading the decompressed image. The intermediate and eventual images of the regular and the modular optimization process are presented. **(a)** Input. **(b)** Regular Decompression (3.75 bpp). **(c)** Regular Degraded Decompression (34.32 dB). **(d)** System-Aware Compression: Input to Last Iteration Compression. **(e)** System-Aware Compression: Decompression (2.21 bpp). **(f)** System-Aware Compression: Degraded Decompression (41.84 dB)

applied on a sharpened version (see Fig. 7d) adjusted to the known blur operator; then, the compressed image at bit-rate 2.21 bpp eventually results in a degraded decompression with moderate blur effects (Fig. 7f) and a PSNR gain of 7.52 dB with respect to the regular compression (which used even a higher bit-rate). See Dar et al. (2018b) for extensive experimental evaluation including PSNR-bitrate performance curves and comparison to additional alternatives. Furthermore, LCD display degradations associated with motion blur are also examined by Dar et al. (2018b).

Additional evaluations of the System-Aware Compression approach are provided by Dar et al. (2018d) for video compression settings including acquisition degradation of low-pass filtering and subsampling and post-decompression nearest-neighbor upsampling. In Dar et al. (2018a), the idea is demonstrated for a simplified model of multimedia distribution networks, where a set of possible degradation operators and their probabilities are considered by the optimized compression process.

Distributed Representations Using Black-Box Modules

All the above problems conduct optimizations for finding one signal (or compressed representation) that minimizes a Lagrangian cost of interest. As shown, these tasks are addressed very well by modular optimizations, relying on sequential black-box module applications. In this section, we demonstrate that the modular optimization approach is useful also to problems seeking for a set of signals (or representations) that collaboratively minimize a joint Lagrangian cost.

The General Framework

The following extends the settings and developments given in section “[Unconstrained Lagrangian Optimizations via ADMM](#)”. The general optimization form for distributed representations broadens the single-representation problem in (1) to an unconstrained Lagrangian form optimizing several signals, i.e.,

$$(\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_K) = \underset{\mathbf{v}_1, \dots, \mathbf{v}_K \in \mathcal{M}}{\operatorname{argmin}} \sum_{i=1}^K R(\mathbf{v}_i) + \lambda D(\mathbf{x}; \mathbf{v}_1, \dots, \mathbf{v}_K) \quad (37)$$

to be solved for the K representations $\mathbf{v}_1, \dots, \mathbf{v}_K \in \mathcal{M} \subset \mathbb{R}^N$, where \mathcal{M} is a (continuous or discrete) subset of the N -dimensional real space. While there are several optimization variables, they all intend to (possibly differently) represent the single given signal $\mathbf{x} \in \mathbb{R}^M$. The general scalar-valued function $R : \mathcal{M} \rightarrow \mathbb{R}$ is defined for individual representation as inputs, and D is a scalar-valued function receiving \mathbf{x} and all the representations together as inputs.

The computational challenge of solving (37) is clear, as it is hard even in the case of optimizing one signal (as discussed in section “[Modular Optimizations Based on Standard Compression Techniques](#)”). Nevertheless, we can utilize variable splitting and ADMM techniques to develop a sequential optimization process addressing (37). Essentially, this is an extension of the ADMM constructions presented in section “[Unconstrained Lagrangian Optimizations via ADMM](#)”. Here the developments originate in the variable splitting applied on (37) via

$$\left(\{\hat{\mathbf{v}}_i\}_{i=1}^K, \{\hat{\mathbf{z}}_i\}_{i=1}^K \right) = \underset{\substack{\{\mathbf{v}_i\}_{i=1}^K \in \mathcal{M} \\ \{\mathbf{z}_i\}_{i=1}^K \in \mathbb{R}^N}}{\operatorname{argmin}} \sum_{i=1}^K R(\mathbf{v}_i) + \lambda D(\mathbf{x}; \mathbf{z}_1, \dots, \mathbf{z}_K) \quad (38)$$

subject to $\mathbf{v}_i = \mathbf{z}_i$ for $i = 1, \dots, K$

where $\mathbf{z}_1, \dots, \mathbf{z}_K \in \mathbb{R}^N$ are auxiliary variables that are not directly constrained to the eventual output domain \mathcal{M} (similar to the developments in section “[Unconstrained Lagrangian Optimizations via ADMM](#)”).

Then, the scaled form of the augmented Lagrangian and the method of multipliers (Boyd et al. 2011, Ch. 2) renders (38) into the sequential process

$$\left(\left\{ \hat{\mathbf{v}}_i^{(t)} \right\}_{i=1}^K, \left\{ \hat{\mathbf{z}}_i^{(t)} \right\}_{i=1}^K \right) = \quad (39)$$

$$\begin{aligned} & \underset{\substack{\{\mathbf{v}_i\}_{i=1}^K \in \mathcal{M} \\ \{\mathbf{z}_i\}_{i=1}^K \in \mathbb{R}^N}}{\operatorname{argmin}} \sum_{i=1}^K R(\mathbf{v}_i) + \lambda D(\mathbf{x}; \mathbf{z}_1, \dots, \mathbf{z}_K) + \frac{\beta}{2} \sum_{i=1}^K \left\| \mathbf{v}_i - \mathbf{z}_i + \mathbf{u}_i^{(t)} \right\|_2^2 \\ & \mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)} \right) \quad \text{for } i = 1, \dots, K \end{aligned} \quad (40)$$

where t denotes the iteration index, $\left\{ \mathbf{u}_i^{(t)} \right\}_{i=1}^K \in \mathbb{R}^N$ are the scaled dual variables, and β is an auxiliary parameter originating at the augmented Lagrangian (note that β is an intentionally joined parameter for the purpose of easing the parameter tuning process). The corresponding ADMM process is obtained by applying one iteration of alternating minimization on (39), leading to

$$\hat{\mathbf{v}}_i^{(t)} = \underset{\mathbf{v}_i \in \mathcal{M}}{\operatorname{argmin}} R(\mathbf{v}_i) + \frac{\beta}{2} \left\| \mathbf{v}_i - \tilde{\mathbf{z}}_i^{(t)} \right\|_2^2 \quad \text{for } i = 1, \dots, K \quad (41)$$

$$\begin{aligned} \hat{\mathbf{z}}_i^{(t)} &= \underset{\mathbf{z}_i \in \mathbb{R}^N}{\operatorname{argmin}} \lambda D \left(\mathbf{x}; \left\{ \hat{\mathbf{z}}_j^{(t)} \right\}_{j=1}^{i-1}, \mathbf{z}_i, \left\{ \hat{\mathbf{z}}_j^{(t-1)} \right\}_{j=i+1}^K \right) + \frac{\beta}{2} \left\| \mathbf{z}_i - \tilde{\mathbf{v}}_i^{(t)} \right\|_2^2 \\ & \quad \text{for } i = 1, \dots, K \end{aligned} \quad (42)$$

$$\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)} \right) \quad \text{for } i = 1, \dots, K \quad (43)$$

where $\tilde{\mathbf{z}}_i^{(t)} \triangleq \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$ and $\tilde{\mathbf{v}}_i^{(t)} \triangleq \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$. Nicely, the obtained process does not only decouple the treatment of $\{\mathcal{M}, R\}$ from D as before (see section “Unconstrained Lagrangian Optimizations via ADMM”) but also separates the treatment of the various representations. Thus, (41), (42), and (43) simplify the challenging structure in (37). Moreover, the subproblems in (41) have the same form associated with black-box modules applied on individual signals (see section “Employing Black-Box Modules”). This casting leads us to the process summarized in Algorithm 7. Also note that in each iteration t the treatment of the K representations is sequential (this reordered procedure is equivalent to the form in (41), (42), and (43)).

Modular Optimizations for Holographic Compression of Images

We now turn to exemplify the generic approach in Algorithm 7 for the purpose of optimizing distributed representations in compressed, standard-compatible, forms.

Algorithm 7 General Modular Optimization of Multiple Representations

- 1: Inputs: $\mathbf{x}, \theta, \tilde{\beta}$.
 - 2: Initialize $t = 0, \hat{\mathbf{z}}^{(0)} = \mathbf{x}, \mathbf{u}^{(1)} = \mathbf{0}$.
 - 3: Initialize $t = 0$.
 - 4: Initialize (for $i = 1, \dots, K$) $\mathbf{u}_i^{(1)} = \mathbf{0}$ and $\hat{\mathbf{z}}_i^{(0)}$ according to the specific application.
 - 5: **repeat**
 - 6: $t \leftarrow t + 1$
 - 7: **for** $i = 1, \dots, K$ **do**
 - 8: $\tilde{\mathbf{z}}_i^{(t)} = \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$
 - 9: $\hat{\mathbf{v}}_i^{(t)} = \text{BlackBoxModule}(\tilde{\mathbf{z}}_i^{(t)}; \theta)$
 - 10: $\tilde{\mathbf{v}}_i^{(t)} = \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$
 - 11: $\hat{\mathbf{z}}_i^{(t)} = \underset{\mathbf{z}_i \in \mathbb{R}^N}{\operatorname{argmin}} \lambda D\left(\mathbf{x}; \left\{\hat{\mathbf{z}}_j^{(t)}\right\}_{j=1}^{i-1}, \mathbf{z}_i, \left\{\hat{\mathbf{z}}_j^{(t-1)}\right\}_{j=i+1}^K\right) + \tilde{\beta} \|\mathbf{z}_i - \tilde{\mathbf{v}}_i^{(t)}\|_2^2$
 - 12: $\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)}\right)$
 - 13: **end for**
 - 14: **until** stopping criterion is satisfied
 - 15: Output: $\hat{\mathbf{v}}_1^{(t)}, \dots, \hat{\mathbf{v}}_K^{(t)}$ and/or other application-specific outputs of *BlackBoxModule*.
-

Our recent framework for holographic compression (Dar and Bruckstein 2021) represents a given signal using a set of distinct compressed descriptions, that any subset of them enables reconstruction of the signal at a quality determined solely by the number of compressed representations utilized. This property of holographic representations is useful for designing progressive refinement mechanisms independent of the order the representations are accessible (Bruckstein et al. 1998, 2000, 2018).

In Dar and Bruckstein (2021) we identified the shift sensitivity of standard compression techniques as a property useful for constructing holographic representations in binary compressed forms. Specifically, compressions of shifted versions of an image provide a set of distinct decompressed images of similar individual qualities, but combining subsets of them (by back-shifts and averaging) achieves remarkable quality gains (see details in Dar and Bruckstein 2021). While this architecture is new and interesting, it does not include optimization aspects. This led us to suggest an optimization procedure unleashing the potential benefits of the shift-based holographic compression settings. Here we can consider this optimization framework as a special case of the generic process described in Algorithm 7, described as follows. First, the general $\{\mathcal{M}, R\}$ notions are set to the respective components $\{\mathcal{S}, R_{\mathcal{S}}\}$ of a standard compression method (as defined in section “[Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization](#)”). This makes the first component in (37) the accumulated bit-cost of all the compressed representations. In Dar and Bruckstein (2021) we set the function D to evaluate the average MSE of reconstructions formed using subsets of m out of the K representations, where $m \in \{2, \dots, K\}$ and assuming $K > 1$. This improves the reconstruction quality for subsets of m representations, at the inevitable expense of reducing their individual qualities. Therefore, we also include in D a regularization

term computing the average MSE of the single-representation reconstructions. We denote the sequence of integers from 1 to K as $[[K]] \triangleq \{1, \dots, K\}$. For $m \in [[K]]$, an m -combination of the set $[[K]]$ is a subset of m distinct numbers from $[[K]]$. We denote the set of all m -combinations of $[[K]]$ as $\binom{[[K]]}{m}$, where the latter contains $\binom{K}{m}$ elements. The corresponding formulation of D is

$$D(\mathbf{x}; \mathbf{v}_1, \dots, \mathbf{v}_K) = \frac{1}{\binom{K}{m}} \sum_{(i_1, \dots, i_m) \in \binom{[[K]]}{m}} D^{(m)}(\mathbf{x}; \mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_m}) \quad (44)$$

$$+ \eta \frac{1}{K} \sum_{i=1}^K D^{(1)}(\mathbf{x}; \mathbf{v}_i)$$

The parameter η determines the regularization level of the individual representation quality. Moreover,

$$D^{(1)}(\mathbf{x}; \mathbf{v}_i) \triangleq \frac{1}{N} \left\| \mathbf{x} - \mathbf{S}_i^T \mathbf{v}_i \right\|_2^2 \quad (45)$$

is the reconstruction MSE corresponding to the single representation \mathbf{v}_i , and

$$D^{(m)}(\mathbf{x}; \mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_m}) \triangleq \frac{1}{N} \left\| \mathbf{x} - \frac{1}{m} \sum_{j=1}^m \mathbf{S}_{i_j}^T \mathbf{v}_{i_j} \right\|_2^2 \quad (46)$$

is the MSE of reconstruction using the m representations $\mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_m}$. The matrices \mathbf{S}_i^T and $\mathbf{S}_{i_j}^T$ correspond to back shift operators matching the shift forms used to create the compressed representations (further details are available in Dar and Bruckstein 2021). Then, by the settings of \mathcal{M} , R , and D , Algorithm 7 is specified for optimizing shift-based holographic compressed representations – this process is described in Algorithm 8.

In Figs. 8 and 9, we provide representative results taken from Dar and Bruckstein (2021). First, Fig. 8 presents reconstructions obtained from JPEG2000-compatible holographic compressions optimized for using sets of four representations. Specifically note the similar quality obtained using the individual representations and how they collaboratively achieve progressive refinement. This behavior is also clearly demonstrated in Fig. 9 by the curves of PSNR versus number of representations (packets) used for reconstructions. In particular, Fig. 9 shows the curves obtained for all the subset combinations in each of the examined methods. This exhibits the ability of the proposed method for optimizing reconstructions that rely on a specified number of representations (independent of the actual participating signals). The interested reader is referred to Dar and Bruckstein (2021) for additional details and experimental demonstrations.

Algorithm 8 Modular Holographic Compression: Optimized for Reconstructions Using m Representations

- 1: Inputs: \mathbf{x} , $\tilde{\beta}$, λ , η , θ , m , K .
 - 2: Initialize $t = 0$.
 - 3: Initialize (for $i = 1, \dots, K$) $\hat{\mathbf{z}}_i^{(0)} = \mathbf{S}_i \mathbf{x}$ and $\mathbf{u}_i^{(1)} = \mathbf{0}$.
 - 4: **repeat**
 - 5: $t \leftarrow t + 1$
 - 6: **for** $i = 1, \dots, K$ **do**
 - 7: $\tilde{\mathbf{z}}_i^{(t)} = \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$
 - 8: $\mathbf{b}_i^{(t)} = \text{StandardCompression}(\tilde{\mathbf{z}}_i^{(t)}, \theta)$
 - 9: $\hat{\mathbf{v}}_i^{(t)} = \text{StandardDecompression}(\mathbf{b}_i^{(t)})$
 - 10: $\tilde{\mathbf{v}}_i^{(t)} = \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$
 - 11:
$$\hat{\mathbf{z}}_i^{(t)} = \frac{N\tilde{\beta}\tilde{\mathbf{v}}_i^{(t)} + \frac{\eta}{\lambda K} \mathbf{S}_i \mathbf{x} + \frac{\lambda}{m^2 \binom{K}{m}} \mathbf{S}_i \mathbf{w}_i^{(m)}}{N\tilde{\beta} + \frac{\eta}{\lambda K} + \frac{\lambda}{m^2 \binom{K}{m}} |\mathcal{I}_i^{(m)}|}$$
 where

$$\mathbf{w}_i^{(m)} \triangleq \sum_{(i_1, \dots, i_m) \in \mathcal{I}_i^{(m)}} \left(m\mathbf{x} - \sum_{j \in \{1, \dots, m\}} \mathbf{S}_{i_j}^T \hat{\mathbf{z}}_{i_j}^{(t)} - \sum_{j \in \{1, \dots, m\}} \mathbf{S}_{i_j}^T \hat{\mathbf{z}}_{i_j}^{(t-1)} \right)$$
 and $\mathcal{I}_i^{(m)}$ contains all the m -combinations including the i^{th} representation.
 - 12: $\mathbf{u}_i^{[t+1]} = \mathbf{u}_i^{(t)} + (\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)})$
 - 13: **end for**
 - 14: **until** stopping criterion is satisfied
 - 15: Output: The binary compressed packets $\mathbf{b}_1^{(t)}, \dots, \mathbf{b}_K^{(t)}$.
-

Conclusion

In this chapter, we presented the recent methodology of modular optimizations, employing black-box modules in procedures addressing various imaging problems. The main idea is that fundamental tasks, such as denoising and compression, have excellent ready-to-use techniques that can be utilized for solving more intricate problems. We presented the developments of ADMM-based algorithms for modular optimizations, starting in general settings exhibiting the essence and prominent guidelines of the approach. Then, we overviewed settings where denoising and compression techniques are operated as stages in sequential procedures for restoration and intricate compression problems. We also outlined the extension of modular optimizations to formation of distributed representations and particularly exemplified it for the case of holographic compression of images. The perspectives emphasized in this chapter should motivate new ideas and settings extending the current class of module types used and problems addressed via modular optimization strategies.

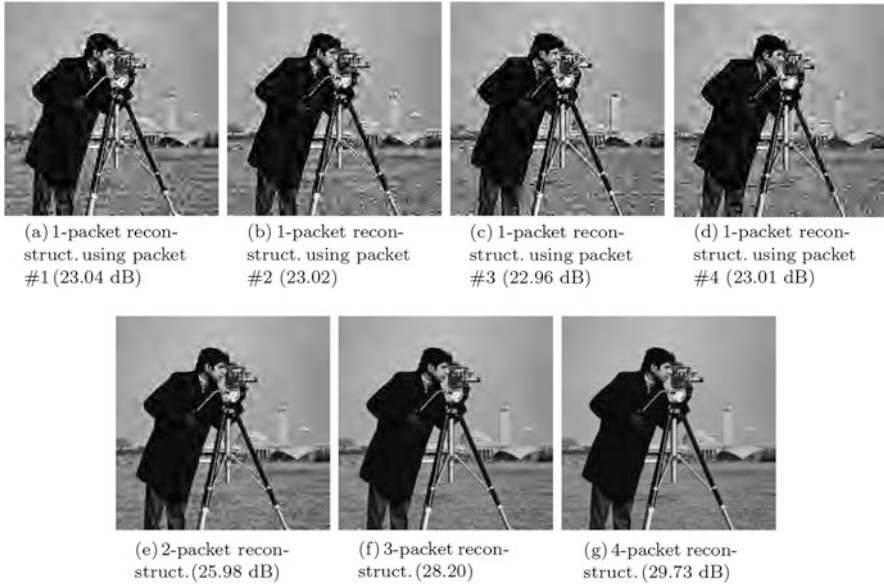


Fig. 8 Examples (taken from Dar and Bruckstein 2021) for reconstructions of the “Cameraman” image using several representations out of a set of four holographic compressed descriptions. The compression employed is JPEG2000 at a compression ratio of 1:50. (a)–(d) the 1-packet reconstructions using each of the individual packets. (e)–(g) examples for the m -packet reconstructions for $m = 2, 3, 4$

Appendix: Operational Rate-Distortion Optimizations in Block-Based Architectures

The computational challenge of operational rate-distortion optimizations (see section “[Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization](#)”) is often addressed via the squared-error metric

$$D(\mathbf{x}, \mathbf{v}) = \|\mathbf{x} - \mathbf{v}\|_2^2, \quad (47)$$

leading to practical forms of the Lagrangian rate-distortion optimization (25). These useful structures also process the signal \mathbf{x} based on its segmentation into a set of nonoverlapping blocks $\{\mathbf{x}_i\}_{i \in \mathcal{I}}$; here, each of them is a column vector of N_b samples, and \mathcal{I} is the set of indices corresponding to the nonoverlapping blocks of the signal. Correspondingly, the decompressed signal \mathbf{v} is decomposed into a set of nonoverlapping blocks $\{\mathbf{v}_i\}_{i \in \mathcal{I}}$. This lets us casting (47) into

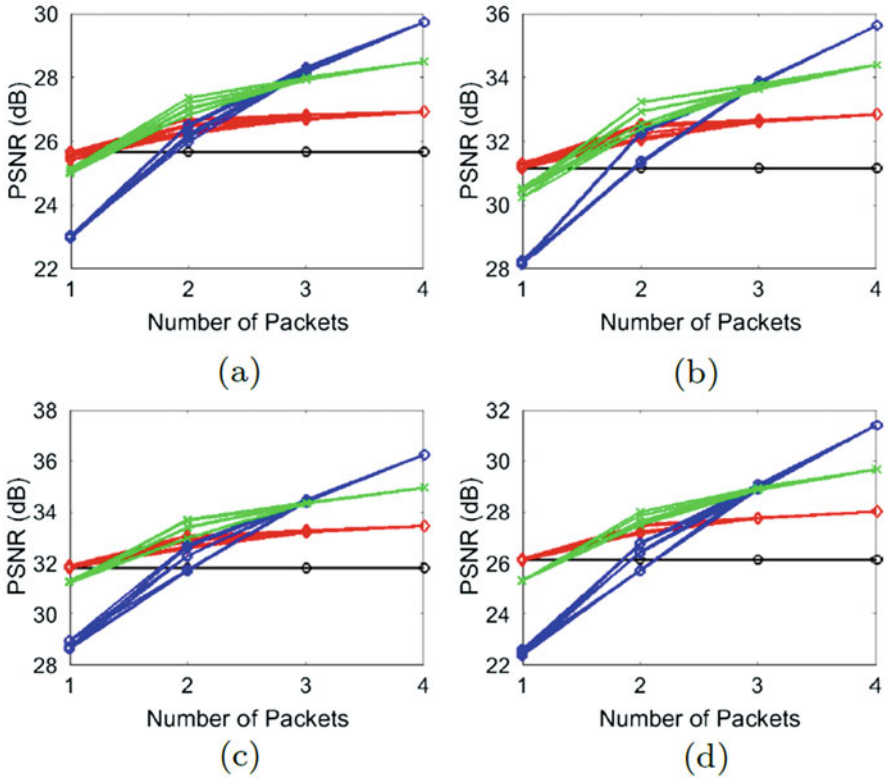


Fig. 9 PSNR versus the number of representations used for the reconstructions. The entire set contains four packets, each formed by JPEG2000 compression at 1:50 compression ratio. The black, red, green, and blue curves, respectively, represent the methods of exact duplications, baseline (unoptimized), optimized for reconstruction from pairs of packets, and optimized for reconstruction from four packets. (a) Cameraman. (b) House. (c) Lena. (d) Barbara

$$D(\mathbf{x}, \mathbf{v}) = \sum_{i \in \mathcal{I}} \|\mathbf{x}_i - \mathbf{v}_i\|_2^2, \quad (48)$$

exhibiting that, for squared-error measures, the total distortion can be computed as the sum of distortions associated with its nonoverlapping blocks. While this property is satisfied for any segmentation of the signal into nonoverlapping blocks, we will exemplify it here for blocks of equal sizes that allow using one block-level compression procedure for all the blocks.

Mirroring the definitions described in section “[Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization](#)” for full-signal compression architectures, the block-level process corresponds to a function $C_b : \mathbb{R}^{N_b} \rightarrow \mathcal{B}_b$, mapping the N_b -dimensional signal-block domain to a discrete set \mathcal{B}_b of binary compressed representations of blocks. The associated block decompression process is denoted

by the function $F_b : \mathcal{B}_b \rightarrow \mathcal{S}_b$, mapping the binary compressed representations in \mathcal{B}_b to their decompressed signal blocks from the discrete set $\mathcal{S}_b \subset \mathbb{R}^{N_b}$. The bit-cost evaluation function $R_b(\mathbf{v}_i)$ is defined to quantify the number of bits needed for the compressed representation matching the decompressed signal block $\mathbf{v}_i \in \mathbb{R}^{N_b}$. Then, the compression of the nonoverlapping signal blocks $\{\mathbf{x}_i\}_{i \in \mathcal{I}}$ producing the decompressed blocks $\{\mathbf{v}_i\}_{i \in \mathcal{I}}$ requires a total bit budget satisfying

$$R(\mathbf{v}) = \sum_{i \in \mathcal{I}} R_b(\mathbf{v}_i). \quad (49)$$

Plugging the block-based compression design into the Lagrangian form (25) gives

$$\{\hat{\mathbf{v}}_i\}_{i \in \mathcal{I}} = \underset{\{\mathbf{v}_i\}_{i \in \mathcal{I}} \in \mathcal{S}_b}{\operatorname{argmin}} \sum_{i \in \mathcal{I}} R_b(\mathbf{v}_i) + \lambda \sum_{i \in \mathcal{I}} \|\mathbf{x}_i - \mathbf{v}_i\|_2^2 \quad (50)$$

that reduces to a set of block-level rate-distortion Lagrangian optimizations, i.e.,

$$\text{For } i \in \mathcal{I}: \hat{\mathbf{v}}_i = \underset{\mathbf{v}_i \in \mathcal{S}_b}{\operatorname{argmin}} R_b(\mathbf{v}_i) + \lambda \|\mathbf{x}_i - \mathbf{v}_i\|_2^2. \quad (51)$$

Note that the block-level optimizations in (51) are independent and refer to the same Lagrangian multiplier λ . Commonly, compression designs are based on processing of low-dimensional blocks, allowing to practically address the block-level optimizations in (51). For example, one can evaluate the Lagrangian cost for all the elements in \mathcal{S}_b (since this set is sufficiently small).

References

- Afonso, M.V., Bioucas-Dias, J.M., Figueiredo, M.A.: Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.* **19**(9), 2345–2356 (2010)
- Ahmad, R., Bouman, C.A., Buzzard, G.T., Chan, S., Reehorst, E.T., Schniter, P.: Plug and play methods for magnetic resonance imaging. *arXiv preprint arXiv:1903.08616* (2019)
- Bahat, Y., Efrat, N., Irani, M.: Non-uniform blind deblurring by reblurring. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3286–3294 (2017)
- Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: *Proceedings of ICLR* (2017)
- F. Bellard, BPG 0.9.6. [Online]. Available: <http://bellard.org/bpg/>
- Beygi, S., Jalali, S., Maleki, A., Mitra, U.: Compressed sensing of compressible signals. In: *IEEE International Symposium on Information Theory (ISIT)*, pp. 2158–2162 (2017a)
- Beygi, S., Jalali, S., Maleki, A., Mitra, U.: An efficient algorithm for compression-based compressed sensing. *arXiv preprint arXiv:1704.01992* (2017b)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011)

- Brifman, A., Romano, Y., Elad, M.: Turning a denoiser into a super-resolver using plug and play priors. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 1404–1408. IEEE (2016)
- Brifman, A., Romano, Y., Elad, M.: Unified single-image and video super-resolution via denoising algorithms. *IEEE Trans. Image Process.* **28**(12), 6063–6076 (2019)
- Bruckstein, A.M., Holt, R.J., Netravali, A.N.: Holographic representations of images. *IEEE Trans. Image Process.* **7**(11), 1583–1597 (1998)
- Bruckstein, A.M., Holt, R.J., Netravali, A.N.: On holographic transform compression of images. *Int. J. Imag. Syst. Technol.* **11**(5), 292–314 (2000)
- Bruckstein, A.M., Ezerman, M.F., Fahreza, A.A., Ling, S.: Holographic sensing. arXiv preprint arXiv:1807.10899 (2018)
- Burger, M., Dirks, H., Schönlieb, C.-B.: A variational model for joint motion estimation and image reconstruction. *SIAM J. Imag. Sci.* **11**(1), 94–128 (2018)
- Buzzard, G.T., Chan, S.H., Sreehari, S., Bouman, C.A.: Plug-and-play unplugged: optimization-free reconstruction using consensus equilibrium. *SIAM J. Imag. Sci.* **11**(3), 2001–2020 (2018)
- Chan, S.H.: Performance analysis of plug-and-play ADMM: a graph signal processing perspective. *IEEE Trans. Comput. Imag.* **5**(2), 274–286 (2019)
- Chan, S.H., Wang, X., Elgandy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. *IEEE Trans. Comput. Imag.* **3**(1), 84–98 (2017)
- Chatterjee, P., Milanfar, P.: Is denoising dead? *IEEE Trans. Image Process.* **19**(4), 895–911 (2009)
- Chou, P.A., Lookabaugh, T., Gray, R.M.: Optimal pruning with applications to tree-structured source coding and modeling. *IEEE Trans. Inf. Theory* **35**(2), 299–315 (1989)
- Corona, V., Aviles-Rivero, A.I., Debroux, N., Graves, M., Le Guyader, C., Schönlieb, C.-B., Williams, G.: Multi-tasking to correct: motion-compensated mri via joint reconstruction and registration. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 263–274. Springer (2019a)
- Corona, V., Benning, M., Ehrhardt, M.J., Gladden, L.F., Mair, R., Recic, A., Sederman, A.J., Reichelt, S., Schönlieb, C.-B.: Enhancing joint reconstruction and segmentation with non-convex bregman iteration. *Inverse Probl.* **35**(5), 055001 (2019b)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Dar, Y., Bruckstein, A.M.: Benefiting from duplicates of compressed data: shift-based holographic compression of images. *J. Math. Imag. Vis.* 1–14 **63**, 380–393 (2021)
- Dar, Y., Bruckstein, A.M., Elad, M.: Image restoration via successive compression. In: Picture Coding Symposium (PCS), pp. 1–5 (2016a)
- Dar, Y., Bruckstein, A.M., Elad, M., Giryes, R.: Postprocessing of compressed images via sequential denoising. *IEEE Trans. Image Process.* **25**(7), 3044–3058 (2016b)
- Dar, Y., Elad, M., Bruckstein, A.M.: Compression for multiple reconstructions. In: IEEE International Conference on Image Processing (ICIP), pp. 440–444 (2018a)
- Dar, Y., Elad, M., Bruckstein, A.M.: Optimized pre-compensating compression. *IEEE Trans. Image Process.* **27**(10), 4798–4809 (2018b)
- Dar, Y., Elad, M., Bruckstein, A.M.: Restoration by compression. *IEEE Trans. Sig. Process.* **66**(22), 5833–5847 (2018c)
- Dar, Y., Elad, M., Bruckstein, A.M.: System-aware compression. In: IEEE International Symposium on Information Theory (ISIT), pp. 2226–2230 (2018d)
- Hong, T., Romano, Y., Elad, M.: Acceleration of red via vector extrapolation. *J. Vis. Commun. Image Represent.* **63**, 102575 (2019)
- Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Sig. Process. Lett.* **24**(12), 1872–1876 (2017)
- Kwan, C., Choi, J., Chan, S., Zhou, J., Budavari, B.: A super-resolution and fusion approach to enhancing hyperspectral images. *Remote Sens.* **10**(9), 1416 (2018)
- Lai, W.-S., Huang, J.-B., Hu, Z., Ahuja, N., Yang, M.-H.: A comparative study for single image blind deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1709 (2016)

- Laparra, V., Berardino, A., Ballé, J., Simoncelli, E.P.: Perceptually optimized image rendering. *J. Opt. Soc. Am. A* **34**, 1511 (2017)
- Liu, J., Moulin, P.: Complexity-regularized image denoising. *IEEE Trans. Image Process.* **10**(6), 841–851 (2001)
- Moulin, P., Liu, J.: Statistical imaging and complexity regularization. *IEEE Trans. Inf. Theory* **46**(5), 1762–1777 (2000)
- Natarajan, B.K.: Filtering random noise from deterministic signals via data compression. *IEEE Trans. Sig. Process.* **43**(11), 2595–2605 (1995)
- Ono, S.: Primal-dual plug-and-play image restoration. *IEEE Sig. Process. Lett.* **24**(8), 1108–1112 (2017)
- Ortega, A., Ramchandran, K.: Rate-distortion methods for image and video compression. *IEEE Sig. Process. Mag.* **15**(6), 23–50 (1998)
- Rissanen, J.: MDL denoising. *IEEE Trans. Inf. Theory* **46**(7), 2537–2543 (2000)
- Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (RED). *SIAM J. Imag. Sci.* **10**(4), 1804–1844 (2017)
- Rond, A., Giryes, R., Elad, M.: Poisson inverse problems by the plug-and-play scheme. *J. Vis. Commun. Image Represent.* **41**, 96–108 (2016)
- Rott Shaham, T., Michaeli, T.: Deformation aware image compression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2453–2462 (2018)
- Shoham, Y., Gersho, A.: Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoust. Speech Sig. Process.* **36**(9), 1445–1453 (1988)
- Shukla, R., Dragotti, P.L., Do, M.N., Vetterli, M.: Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. *IEEE Trans. Image Process.* **14**(3), 343–359 (2005)
- Sreehari, S., Venkatakrishnan, S., Wohlberg, B., Buzzard, G.T., Drummy, L.F., Simmons, J.P., Bouman, C.A.: Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Trans. Comput. Imag.* **2**(4), 408–423 (2016)
- Sullivan, G.J., Wiegand, T.: Rate-distortion optimization for video compression. *IEEE Sig. Process. Mag.* **15**(6), 74–90 (1998)
- Sullivan, G.J., Ohm, J., Han, W.-J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
- Sun, Y., Wohlberg, B., Kamilov, U.S.: An online plug-and-play algorithm for regularized image reconstruction. *IEEE Trans. Comput. Imag.* **5**, 395–408 (2019a)
- Sun, Y., Xu, S., Li, Y., Tian, L., Wohlberg, B., Kamilov, U.S.: Regularized fourier ptychography using an online plug-and-play algorithm. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7665–7669. IEEE (2019b)
- Tirer, T., Giryes, R.: Image restoration by iterative denoising and backward projections. *IEEE Trans. Image Process.* **28**(3), 1220–1234 (2018a)
- Tirer, T., Giryes, R.: An iterative denoising and backwards projections method and its advantages for blind deblurring. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 973–977. IEEE (2018b)
- Tirer, T., Giryes, R.: Back-projection based fidelity term for ill-posed linear inverse problems. *arXiv preprint arXiv:1906.06794* (2019)
- Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: *IEEE GlobalSIP* (2013)
- Yazaki, Y., Tanaka, Y., Chan, S.H.: Interpolation and denoising of graph signals using plug-and-play ADMM. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5431–5435. IEEE (2019)
- Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 479–486 (2011)



Connecting Hamilton-Jacobi Partial Differential Equations with Maximum a Posteriori and Posterior Mean Estimators for Some Non-convex Priors

Jérôme Darbon, Gabriel P. Langlois, and Tingwei Meng

Contents

Introduction	210
First-Order Hamilton-Jacobi PDEs and Optimization Problems	212
Single-Time HJ PDEs and Image Denoising Models	213
Multi-time HJ PDEs and Image Decomposition Models	214
Min-Plus Algebra for HJ PDEs and Certain Non-convex Regularizations	216
Application to Certain Decomposition Problems	220
Viscous Hamilton-Jacobi PDEs and Bayesian Estimation	224
Viscous HJ PDEs and Posterior Mean Estimators for Log-Concave Models	225
On Viscous HJ PDEs with Certain Non-log-Concave Priors	227
Conclusion	230
References	230

Abstract

Many imaging problems can be formulated as inverse problems expressed as finite-dimensional optimization problems. These optimization problems generally consist of minimizing the sum of a data fidelity and regularization terms. In Darbon (SIAM J. Imag. Sci. 8:2268–2293, 2015), Darbon and Meng, (On decomposition models in imaging sciences and multi-time Hamilton-Jacobi partial differential equations, arXiv preprint arXiv:1906.09502, 2019), connections between these optimization problems and (multi-time) Hamilton-Jacobi partial differential equations have been proposed under the convexity assumptions of both the data fidelity and regularization terms. In particular, under these convexity assumptions, some representation formulas for a minimizer can

J. Darbon (✉) · G. P. Langlois · T. Meng
Division of Applied Mathematics, Brown University, Providence, RI, USA
e-mail: jerome_darbon@brown.edu; gabriel_provencher_langlois@brown.edu;
tingwei_meng@brown.edu

be obtained. From a Bayesian perspective, such a minimizer can be seen as a maximum a posteriori estimator. In this chapter, we consider a certain class of non-convex regularizations and show that similar representation formulas for the minimizer can also be obtained. This is achieved by leveraging min-plus algebra techniques that have been originally developed for solving certain Hamilton-Jacobi partial differential equations arising in optimal control. Note that connections between viscous Hamilton-Jacobi partial differential equations and Bayesian posterior mean estimators with Gaussian data fidelity terms and log-concave priors have been highlighted in Darbon and Langlois, (On Bayesian posterior mean estimators in imaging sciences and Hamilton-Jacobi partial differential equations, arXiv preprint arXiv:2003.05572, 2020). We also present similar results for certain Bayesian posterior mean estimators with Gaussian data fidelity and certain non-log-concave priors using an analogue of min-plus algebra techniques.

Keywords

Hamilton–Jacobi partial differential equation · Maximum a posteriori estimation · Bayesian posterior mean estimation · Min-plus algebra · Imaging inverse problems

Introduction

Many low-level signal, image processing, and computer vision problems are formulated as inverse problems that can be solved using variational (Aubert and Kornprobst 2002; Scherzer et al. 2009; Vese et al. 2016) or Bayesian approaches (Winkler 2003). Both approaches have been very effective, for example, at solving image restoration (Bouman and Sauer 1993; Likas and Galatsanos 2004; Rudin et al. 1992), segmentation (Boykov et al. 2001; Chan et al. 2006; Chan and Vese 2001), and image decomposition problems (Aujol et al. 2005; Osher et al. 2003).

As an illustration, let us consider the following image denoising problem in finite dimension that formally reads as follows:

$$\mathbf{x} = \bar{\mathbf{u}} + \boldsymbol{\eta},$$

where $\mathbf{x} \in \mathbb{R}^n$ is the observed image that is the sum of an unknown ideal image $\bar{\mathbf{u}} \in \mathbb{R}^n$ and an additive perturbation or noise realization $\boldsymbol{\eta} \in \mathbb{R}^n$. We aim to estimate $\bar{\mathbf{u}}$.

A standard variational approach for solving this problem consists of estimating $\bar{\mathbf{u}}$ as a minimizer of the following optimization problem:

$$\min_{\mathbf{u} \in \mathbb{R}^n} \{ \lambda D(\mathbf{x} - \mathbf{u}) + J(\mathbf{u}) \}, \quad (1)$$

where $D : \mathbb{R}^n \rightarrow \mathbb{R}$ is generally called the data fidelity term and contains the knowledge we have on the perturbation $\boldsymbol{\eta}$ while $J : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is

called the regularization term and encodes the knowledge on the image we wish to reconstruct. The nonnegative parameter λ relatively weights the data fidelity and the regularization terms. Note that minimizers of (1) are called maximum a posteriori (MAP) estimators in a Bayesian setting. Also note that variational-based approaches for estimating $\bar{\mathbf{u}}$ are particularly appealing when both the data fidelity and regularization terms are convex because (1) becomes a convex optimization problem that can be efficiently solved using convex optimization algorithms (see, e.g., Chambolle and Pock 2016). Many regularization terms have been proposed in the literature (Aubert and Kornprobst 2002; Winkler 2003). Popular choices for these regularization terms involve robust edge-preserving priors (Bouman and Sauer 1993; Charbonnier et al. 1997; Geman and Yang 1995; Geman and Reynolds 1992; Nikolova and Chan 2007; ?; Rudin et al. 1992) because they allow the reconstructed image to have sharp edges. For the sake of simplicity, we only describe in this introduction regularizations that are expressed using pairwise interactions which take the following form:

$$J(\mathbf{u}) = \sum_{i,j=1}^n w_{ij} f(u_i - u_j), \quad (2)$$

where $f : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$ and $w_{i,j} \geq 0$. Note that our results that will be presented later do not rely on pairwise interaction-based models and work for more general regularization terms. A popular choice is the celebrated Total Variation (Bouman and Sauer 1993; Rudin et al. 1992), which corresponds to consider $f(z) = |z|$ in (2). The use of Total Variation as a regularization term has been very popular since the seminal works of Bouman and Sauer (1993); Rudin et al. (1992) because it is convex and allows the reconstructed image to preserve edges well. When the data fidelity D is quadratic, this model is known as the celebrated Rudin-Osher-Fatemi model (Rudin et al. 1992). Following the seminal works of Charbonnier et al. (1997), Geman and Yang (1995) and Geman and Reynolds (1992), another class of edge-preserving priors corresponds to half-quadratic-based regularizations that read as follows:

$$f(z) = \begin{cases} |z|^2 & \text{if } |z| \leq 1, \\ 1 & \text{otherwise.} \end{cases} \quad (3)$$

Note that the quadratic term above can be replaced by $|\cdot|$, i.e., we consider:

$$f(z) = \begin{cases} |z| & \text{if } |z| \leq 1, \\ 1 & \text{otherwise,} \end{cases} \quad (4)$$

which corresponds to the truncated Total Variation regularization (see Darbon et al. 2009; Dou et al. 2017 for instance).

There is a large body of literature on variational methods (e.g., Aubert and Kornprobst 2002; Chambolle et al. 2010; Chan and Shen 2005; Scherzer et al. 2009; Vese et al. 2016). In particular, in Darbon (2015) and Darbon and Meng (2020), connections between convex optimization problems of the form of (1) and Hamilton-Jacobi partial differential equations (HJ PDEs) were highlighted. Specifically, it is shown that the dependence of the minimal value of these problems with respect to the observed data \mathbf{x} and the smoothing parameter λ is governed by HJ PDEs, where the initial data corresponds to the regularization term J and the Hamiltonian is related to the data fidelity (see section “[First-Order Hamilton-Jacobi PDEs and Optimization Problems](#)” for details). However, the connections between HJ PDEs and certain variational imaging problems described in Darbon (2015) and Darbon and Meng (2020) require the convexity of both the data fidelity and regularization terms. Note that these connections between HJ PDEs and imaging problems also hold for image decomposition models (see section “[Multi-time HJ PDEs and Image Decomposition Models](#)”) using multi-time HJ PDEs (Darbon and Meng 2020).

Our goal is to extend the results of Darbon (2015) and Darbon and Meng (2020) to certain non-convex regularization terms using min-plus algebra techniques (Akian et al. 2006, 2008; Dower et al. 2015; Fleming and McEneaney 2000; Gaubert et al. 2011; Kolokoltsov and Maslov 1997; McEneaney 2006, 2007; McEneaney et al. 2008; McEneaney and Kluberg 2009) that were originally designed for solving certain HJ PDEs arising in optimal control problems. We also propose an analogue of this approach for certain Bayesian posterior mean estimators when the data fidelity is Gaussian.

The rest of this chapter is as follows. Section “[First-Order Hamilton-Jacobi PDEs and Optimization Problems](#)” reviews connections of image denoising and decomposition models with HJ PDEs under convexity assumptions. We then present a min-plus algebra approach for single-time and multi-time HJ PDEs that allows us to consider certain non-convex regularizations in these image denoising and decomposition models. In particular, this min-plus algebra approach yields practical numerical optimization algorithms for solving certain image denoising and decomposition models. Section “[Viscous Hamilton-Jacobi PDEs and Bayesian Estimation](#)” reviews connections between viscous HJ PDEs and posterior mean estimators with Gaussian data fidelity term and log-concave priors. We also present an analogue of the min-plus algebra technique for these viscous HJ PDEs with certain priors that are not log-concave. Finally, we draw some conclusions in section “[Conclusion](#)”.

First-Order Hamilton-Jacobi PDEs and Optimization Problems

In this section, we discuss the connections between some variational optimization models in imaging sciences and HJ PDEs. In section “[Single-Time HJ PDEs and Image Denoising Models](#)”, we consider the convex image denoising model (1) and

the single-time HJ PDE. In section “[Multi-time HJ PDEs and Image Decomposition Models](#)”, we review the connections between convex image decomposition models and the multi-time HJ PDE system. In section “[Min-Plus Algebra for HJ PDEs and Certain Non-convex Regularizations](#)”, we use the min-plus algebra technique to solve certain optimization problems in which one regularization term is non-convex. In section “[Application to Certain Decomposition Problems](#)”, we provide an application of the min-plus algebra technique to certain image decomposition problems, which yields practical numerical optimization algorithms.

Single-Time HJ PDEs and Image Denoising Models

As described in the introduction, an important class of optimization models in imaging sciences for denoising takes the form of (1), where $\lambda > 0$ is a positive parameter, $\mathbf{x} \in \mathbb{R}^n$ is the observed image with n pixels, and $\mathbf{u} \in \mathbb{R}^n$ is the reconstructed image. The objective function is the weighted sum of the convex regularization term J and the convex data fidelity term D .

The connection between the class of optimization models (1) and first-order HJ PDEs has been discussed in Darbon (2015). Specifically, if the data fidelity term λD can be written in the form of $tH^*\left(\frac{\cdot}{t}\right)$ (where H^* denotes the Legendre transform of a convex function H and $t > 0$ is a new parameter that depends on λ), then the minimization problem (1) defines a function $S: \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}$ as follows:

$$S(\mathbf{x}, t) = \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J(\mathbf{u}) + tH^*\left(\frac{\mathbf{x} - \mathbf{u}}{t}\right) \right\}. \quad (5)$$

For instance, if the noise is assumed to be Gaussian, independent, identically distributed, and additive, we impose the quadratic data fidelity $D(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ for each $\mathbf{x} \in \mathbb{R}^n$. Then D satisfies $\lambda D(\mathbf{x}) = tH^*\left(\frac{\mathbf{x}}{t}\right)$ where $H^*(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and $t = \frac{1}{\lambda}$.

Formula (5) is called the Lax-Oleinik formula (Bardi and Evans 1984; Evans 2010; Hopf 1965) in the PDE literature, and it solves the following first-order HJ PDE:

$$\begin{cases} \frac{\partial S}{\partial t}(\mathbf{x}, t) + H(\nabla_{\mathbf{x}} S(\mathbf{x}, t)) = 0 & \mathbf{x} \in \mathbb{R}^n, t > 0, \\ S(\mathbf{x}, 0) = J(\mathbf{x}) & \mathbf{x} \in \mathbb{R}^n, \end{cases} \quad (6)$$

where the function $H: \mathbb{R}^n \rightarrow \mathbb{R}$ is called the Hamiltonian and $J: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is the initial data. In Darbon (2015), a representation formula for the minimizer of (5) is given, and we state it in the following proposition. Here and in the remainder of this chapter, we use $\Gamma_0(\mathbb{R}^n)$ to denote the set of convex, proper and lower semicontinuous functions from \mathbb{R}^n to $\mathbb{R} \cup \{+\infty\}$.

Proposition 1. *Assume $J \in \Gamma_0(\mathbb{R}^n)$, and assume $H: \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable, strictly convex, and 1-coercive function. Then the Lax-Oleinik formula (5) gives the differentiable and convex solution $S: \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}$ to the HJ PDE (6). Moreover, for each $\mathbf{x} \in \mathbb{R}^n$ and $t > 0$, the minimizer in (5) exists and is unique, which we denote by $\mathbf{u}(\mathbf{x}, t)$, and satisfies*

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{x} - t \nabla H(\nabla_{\mathbf{x}} S(\mathbf{x}, t)). \tag{7}$$

Equation (7) in this proposition gives the relation between the minimizer \mathbf{u} in the Lax-Oleinik formula (5) and the spatial gradient of the solution to the HJ PDE (6). In other words, one can compute the minimizer in the corresponding denoising model (1) using the spatial gradient $\nabla_{\mathbf{x}} S(\mathbf{x}, t)$ of the solution, and vice versa.

There is another set of assumptions for the conclusion of the proposition above to hold. For the details, we refer the reader to Darbon (2015).

Multi-time HJ PDEs and Image Decomposition Models

In this subsection, we consider the following image decomposition models:

$$\min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ J \left(\mathbf{x} - \sum_{i=1}^N \mathbf{u}_i \right) + \sum_{i=1}^N \lambda_i f_i(\mathbf{u}_i) \right\}, \tag{8}$$

where $\lambda_1, \dots, \lambda_N$ are positive parameters, $\mathbf{x} \in \mathbb{R}^n$ is the observed image with n pixels, and $\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n$ correspond to the decomposition of the original image \mathbf{x} . In Darbon and Meng (2020), the relation between the decomposition model (8) and the multi-time HJ PDE system has been proposed under the convexity assumptions of J and the functions f_1, \dots, f_N .

In the decomposition model, an image is assumed to be the summation of $N + 1$ components, denoted as $\mathbf{u}_1, \dots, \mathbf{u}_N$ and the residual $\mathbf{x} - \sum_{i=1}^N \mathbf{u}_i$. The feature of each part \mathbf{u}_i is characterized by a convex function f_i , and the residual $\mathbf{x} - \sum_{i=1}^N \mathbf{u}_i$ is characterized by a convex regularization term J . If the function $\lambda_i f_i$ can be written in the form of $t_i H_i^* \left(\frac{\cdot}{t_i} \right)$ (where H_i^* denotes the Legendre transform of a convex function H_i and $t_i > 0$ is a new parameter which depends on λ_i) for each $i \in \{1, \dots, N\}$, then the image decomposition model (8) defines a function $S: \mathbb{R}^n \times (0, +\infty)^N \rightarrow \mathbb{R}$ as follows:

$$S(\mathbf{x}, t_1, \dots, t_N) = \min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ J \left(\mathbf{x} - \sum_{i=1}^N \mathbf{u}_i \right) + \sum_{i=1}^N t_i H_i^* \left(\frac{\mathbf{u}_i}{t_i} \right) \right\}. \tag{9}$$

This formula is called the generalized Lax-Oleinik formula (Lions and Rochet 1986; Tho 2005) which solves the following multi-time HJ PDE system:

$$\left\{ \begin{array}{l} \frac{\partial S(\mathbf{x}, t_1, \dots, t_N)}{\partial t_1} + H_1(\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)) = 0 \quad \mathbf{x} \in \mathbb{R}^n, t_1, \dots, t_N > 0, \\ \vdots \\ \frac{\partial S(\mathbf{x}, t_1, \dots, t_N)}{\partial t_j} + H_j(\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)) = 0 \quad \mathbf{x} \in \mathbb{R}^n, t_1, \dots, t_N > 0, \\ \vdots \\ \frac{\partial S(\mathbf{x}, t_1, \dots, t_N)}{\partial t_N} + H_N(\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)) = 0 \quad \mathbf{x} \in \mathbb{R}^n, t_1, \dots, t_N > 0, \\ S(\mathbf{x}, 0, \dots, 0) = J(\mathbf{x}) \quad \mathbf{x} \in \mathbb{R}^n, \end{array} \right. \quad (10)$$

where $H_1, \dots, H_N: \mathbb{R}^n \rightarrow \mathbb{R}$ are called Hamiltonians and $J: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is the initial data. Under certain assumptions (see Prop. 2), the generalized Lax-Oleinik formula (9) gives the solution $S(\mathbf{x}, t_1, \dots, t_N)$ to the multi-time HJ PDE system (10). In Darbon and Meng (2020), the relation between the minimizer in (9) and the spatial gradient $\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)$ of the solution to the multi-time HJ PDE system (10) is studied. This relation is described in the following proposition.

Proposition 2. *Assume $J \in \Gamma_0(\mathbb{R}^n)$, and assume $H_j: \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex and l -coercive function for each $j \in \{1, \dots, N\}$. Suppose there exists $j \in \{1, \dots, N\}$ such that H_j is strictly convex. Then the generalized Lax-Oleinik formula (9) gives the differentiable and convex solution $S: \mathbb{R}^n \times (0, +\infty)^N \rightarrow \mathbb{R}$ to the multi-time HJ PDE system (10). Moreover, for each $\mathbf{x} \in \mathbb{R}^n$ and $t_1, \dots, t_N > 0$, the minimizer in (9) exists. We denote by $(\mathbf{u}_1(\mathbf{x}, t_1, \dots, t_N), \dots, \mathbf{u}_N(\mathbf{x}, t_1, \dots, t_N))$ any minimizer of the minimization problem in (9) with parameters $\mathbf{x} \in \mathbb{R}^n$ and $t_1, \dots, t_N \in (0, +\infty)$. Then, for each $j \in \{1, \dots, N\}$, there holds*

$$\mathbf{u}_j(\mathbf{x}, t_1, \dots, t_N) \in t_j \partial H_j(\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)),$$

where ∂H_j denotes the subdifferential of H_j .

Furthermore, if all the Hamiltonians H_1, \dots, H_N are differentiable, then the minimizer is unique and satisfies

$$\mathbf{u}_j(\mathbf{x}, t_1, \dots, t_N) = t_j \nabla H_j(\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)), \quad (11)$$

for each $j \in \{1, \dots, N\}$.

As a result, when the assumptions in the proposition above are satisfied, one can compute the minimizer to the corresponding decomposition model (8) using equation (11) and the spatial gradient $\nabla_{\mathbf{x}} S(\mathbf{x}, t_1, \dots, t_N)$ of the solution to the multi-time HJ PDE (10).

Min-Plus Algebra for HJ PDEs and Certain Non-convex Regularizations

In the previous two subsections, we considered the optimization models (1) and (8) where each term was assumed to be convex. When J is non-convex, solutions to (6) may not be classical (in the sense that it is not differentiable). It is well-known that the concept of viscosity solutions (Bardi and Capuzzo-Dolcetta 1997; Barles 1994; Barron et al. 1984; Crandall et al. 1992; Evans 2010; Fleming and Soner 2006) is generally the appropriate notion of solutions for these HJ PDEs. Note that Lax-Oleinik formulas (1) and (8) yield viscosity solutions to their respective HJ PDEs (6) and (10). However, these Lax-Oleinik formulas result in non-convex optimization problems.

In this subsection, we use the min-plus algebra technique (Akian et al. 2006, 2008; Dower et al. 2015; Fleming and McEneaney 2000; Gaubert et al. 2011; Kolokoltsov and Maslov 1997; McEneaney 2006, 2007; McEneaney et al. 2008; McEneaney and Kluberg 2009) to handle the cases when the term J in (1) and (8) is assumed to be a non-convex function in the following form:

$$J(\mathbf{x}) = \min_{i \in \{1, \dots, m\}} J_i(\mathbf{x}) \text{ for every } \mathbf{x} \in \mathbb{R}^n, \tag{12}$$

where $J_i \in \Gamma_0(\mathbb{R}^n)$ for each $i \in \{1, \dots, m\}$.

First, we consider the single-time HJ PDE (6). By min-plus algebra theory, the semigroup of this HJ PDE is linear with respect to the min-plus algebra. In other words, under certain assumptions the solution S to the HJ PDE $\frac{\partial S}{\partial t}(\mathbf{x}, t) + H(\nabla_{\mathbf{x}} S(\mathbf{x}, t)) = 0$ with initial data J is the minimum of the solution S_i to the HJ PDE $\frac{\partial S_i}{\partial t}(\mathbf{x}, t) + H(\nabla_{\mathbf{x}} S_i(\mathbf{x}, t)) = 0$ with initial data J_i . Specifically, if the Lax-Oleinik formula (5) solves the HJ PDE (6) for each $i \in \{1, \dots, m\}$ and the minimizer \mathbf{u} exists (for instance, when $J_i \in \Gamma_0(\mathbb{R}^n)$ for each $i \in \{1, \dots, m\}$, and $H: \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable, strictly convex, and 1-coercive function), then we have:

$$\begin{aligned} S(\mathbf{x}, t) &= \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \\ &= \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \min_{i \in \{1, \dots, m\}} J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \\ &= \min_{\mathbf{u} \in \mathbb{R}^n} \min_{i \in \{1, \dots, m\}} \left\{ J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \tag{13} \\ &= \min_{i \in \{1, \dots, m\}} \left\{ \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \right\} \\ &= \min_{i \in \{1, \dots, m\}} S_i(\mathbf{x}, t). \end{aligned}$$

Therefore, the solution $S(\mathbf{x}, t)$ is given by the pointwise minimum of $S_i(\mathbf{x}, t)$ for $i \in \{1, \dots, m\}$. Note that the Lax-Oleinik formula (5) yields a convex problem for each $S_i(\mathbf{x}, t)$ with $i \in \{1, \dots, m\}$. Therefore this approach seems particularly appealing to solve these non-convex optimization problems and associated HJ PDEs. Note that such an approach is embarrassingly parallel since we can solve the initial data J_i for each $i \in \{1, \dots, m\}$ independently and compute in linear time the pointwise minimum. However, this approach is only feasible if m is not too big. We will see later in this subsection that robust edge-preserving priors (e.g., truncated Total Variation or truncated quadratic) can be written in the form of (12), but m is exponential in n .

We can also compute the set of minimizers $\mathbf{u}(\mathbf{x}, t)$ as follows. Here, we abuse notation and use $\mathbf{u}(\mathbf{x}, t)$ to denote the set of minimizers, which may be not a singleton set when the minimizer is not unique. We can write

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \min_{i \in \{1, \dots, m\}} J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \\ &= \arg \min_{\mathbf{u} \in \mathbb{R}^n} \min_{i \in \{1, \dots, m\}} \left\{ J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \\ &= \bigcup_{i \in I(\mathbf{x}, t)} \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J_i(\mathbf{u}) + tH^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\}, \end{aligned} \quad (14)$$

where the index set $I(\mathbf{x}, t)$ is defined by

$$I(\mathbf{x}, t) = \arg \min_{i \in \{1, \dots, m\}} S_i(\mathbf{x}, t). \quad (15)$$

A specific example is when the regularization term J is the truncated regularization term with pairwise interactions in the following form:

$$J(\mathbf{x}) = \sum_{(i,j) \in E} w_{ij} f(x_i - x_j), \text{ for each } \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n, \quad (16)$$

where $w_{ij} \geq 0$, $f(x) = \min\{g(x), 1\}$ for some convex function $g: \mathbb{R} \rightarrow \mathbb{R}$ and $E = \{1, \dots, n\} \times \{1, \dots, n\}$. This function can be written as the minimum of a collection of convex functions $J_\Omega: \mathbb{R}^n \rightarrow \mathbb{R}$ as the following:

$$J(\mathbf{x}) = \min_{\Omega \subseteq E} J_\Omega,$$

with each J_Ω defined by

$$J_\Omega := \left\{ \sum_{(i,j) \in \Omega} w_{ij} + \sum_{(i,j) \notin \Omega} w_{ij} g(x_i - x_j) \right\},$$

where Ω is any subset of E . The truncated regularization term (16) can therefore be written in the form of (12), and hence the minimizer to the corresponding optimization problem (1) with the non-convex regularization term J in (16) can be computed using (14).

We give here two examples of truncated regularization term with pairwise interactions in the form of (16). First, let g be the ℓ^1 norm. Then J is the truncated discrete Total Variation regularization term defined by

$$J(\mathbf{x}) = \sum_{(i,j) \in E} w_{ij} \min\{|x_i - x_j|, 1\}, \text{ for each } \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n. \quad (17)$$

This function J can be written as the formula (16) with $f: \mathbb{R} \rightarrow \mathbb{R}$ given by Eq. (4). Second, let g be the quadratic function. Then J is the half-quadratic regularization term defined by

$$J(\mathbf{x}) = \sum_{(i,j) \in E} w_{ij} \min\{(x_i - x_j)^2, 1\}, \text{ for each } \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n. \quad (18)$$

This function J can be written as the formula (16) with $f: \mathbb{R} \rightarrow \mathbb{R}$ given by Eq. (3). This specific form of edge-preserving prior was investigated in the seminal works of Charbonnier et al. (1997), Geman and Yang (1995) and Geman and Reynolds (1992). Several algorithms have been proposed to solve the resultant non-convex optimization problem (13), i.e., the solution to the corresponding HJ PDE, for some specific choice of data fidelity terms (e.g., Allain et al. 2006; Idier 2001; Geman and Yang 1995; Geman and Reynolds 1992; Nikolova and Ng 2005; Champagnat and Idier 2004; Nikolova and Ng 2001).

Suppose now, for general regularization terms J in the form of (16), that we have Gaussian noise. Then the data fidelity term is quadratic and $H(\mathbf{p}) = \frac{1}{2} \|\mathbf{p}\|_2^2$ and $t = \frac{1}{\lambda}$. Hence, for this example, using (14), we obtain the set of minimizers:

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \bigcup_{\Omega \in I(\mathbf{x}, t)} \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J_\Omega(\mathbf{u}) + t H^* \left(\frac{\mathbf{x} - \mathbf{u}}{t} \right) \right\} \\ &= \bigcup_{\Omega \in I(\mathbf{x}, t)} \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \sum_{(i,j) \notin \Omega} w_{ij} g(u_i - u_j) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\} \\ &= \bigcup_{\Omega \in I(\mathbf{x}, t)} \{\mathbf{x} - t \nabla_{\mathbf{x}} S_\Omega(\mathbf{x}, t)\} \end{aligned}$$

where

$$S_{\Omega}(\mathbf{x}, t) = \sum_{(i,j) \in \Omega} w_{ij} + \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \sum_{(i,j) \notin \Omega} w_{ij} g(u_i - u_j) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\}$$

and

$$I(\mathbf{x}, t) = \arg \min_{\Omega \subseteq E} S_{\Omega}(\mathbf{x}, t).$$

The same result also holds for the multi-time HJ PDE system (10). Indeed, if J is a non-convex regularization term given by (12), and $S, S_j: \mathbb{R}^n \times (0, +\infty)^N \rightarrow \mathbb{R}$ are the solutions to the multi-time HJ PDE system (10) with initial data J and J_j , respectively, then similarly we have the min-plus linearity of the semigroup under certain assumptions. Specifically, if the Lax-Oleinik formula (9) solves the multi-time HJ PDE system (10) for each $i \in \{1, \dots, m\}$ (for instance, when H and J_i satisfy the assumptions in Prop. 2 for each $i \in \{1, \dots, m\}$), then there holds

$$\begin{aligned} S(\mathbf{x}, t_1, \dots, t_N) &= \min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ \min_{i \in \{1, \dots, m\}} J_i \left(\mathbf{x} - \sum_{j=1}^N \mathbf{u}_j \right) + \sum_{j=1}^N t_j H_j^* \left(\frac{\mathbf{u}_j}{t_j} \right) \right\} \\ &= \min_{i \in \{1, \dots, m\}} \left\{ \min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ J_i \left(\mathbf{x} - \sum_{j=1}^N \mathbf{u}_j \right) + \sum_{j=1}^N t_j H_j^* \left(\frac{\mathbf{u}_j}{t_j} \right) \right\} \right\} \\ &= \min_{i \in \{1, \dots, m\}} S_i(\mathbf{x}, t_1, \dots, t_N). \end{aligned} \tag{19}$$

Let $M \subset \mathbb{R}^n \times \mathbb{R}^N$ be the set of minimizers of (9) with J given by (12). Then M satisfies

$$\begin{aligned} M &= \arg \min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ \min_{i \in \{1, \dots, m\}} J_i \left(\mathbf{x} - \sum_{j=1}^N \mathbf{u}_j \right) + \sum_{j=1}^N t_j H_j^* \left(\frac{\mathbf{u}_j}{t_j} \right) \right\} \\ &= \bigcup_{i \in I(\mathbf{x}, t_1, \dots, t_N)} \arg \min_{\mathbf{u}_1, \dots, \mathbf{u}_N \in \mathbb{R}^n} \left\{ J_i \left(\mathbf{x} - \sum_{j=1}^N \mathbf{u}_j \right) + \sum_{j=1}^N t_j H_j^* \left(\frac{\mathbf{u}_j}{t_j} \right) \right\}, \end{aligned} \tag{20}$$

where the index set $I(\mathbf{x}, t_1, \dots, t_N)$ is defined by

$$I(\mathbf{x}, t_1, \dots, t_N) = \arg \min_{i \in \{1, \dots, m\}} S_i(\mathbf{x}, t_1, \dots, t_N). \tag{21}$$

As a result, we can use (20) to obtain the minimizers of the decomposition model (8) with the non-convex regularization term J in the form of (12), such as the function in (16) and the truncated Total Variation function (17).

In summary, one can compute the minimizers of the optimization problems (1) and (8) with a non-convex function J in the form of (12) using the aforementioned min-plus algebra technique. Furthermore, this technique can be extended to handle other cases. For instance, in the denoising model (1), if the data fidelity term D is in the form of (12) and the prior term $\frac{J(u)}{\lambda}$ can be written as $tH^*\left(\frac{u}{t}\right)$, then one can still compute the minimizer of this problem using the min-plus algebra technique on the HJ PDE with initial data D . Similarly, because of the symmetry in the decomposition model (8), if there is only one non-convex term f_j and if it can be written in the form of (12), then one can apply the min-plus algebra technique to the multi-time HJ PDE with initial data f_j .

In general, however, there is a drawback to the min-plus algebra technique. To compute the minimizers using (14) and (20), we need to compute the index set $I(x, t)$ and $I(x, t_1, \dots, t_N)$ defined in (15) and (21), which involves solving m HJ PDEs to obtain the solutions S_1, \dots, S_m . When m is too large, this approach is impractical since it involves solving too many HJ PDEs. For instance, if J is the truncated Total Variation in (17), the number m equals the number of subsets of the set E , i.e., $m = 2^{|E|}$, which is computationally intractable. Hence, in general, it is impractical to use (14) and (20) to solve the problems (1) and (8) where the regularization term J is given by the truncated Total Variation. The same issue arises when the truncated Total Variation is replaced by half-quadratic regularization. Several authors attempted to address this intractability for half-quadratic regularizations by proposing heuristic optimization methods that aim to compute a global minimizer (Allain et al. 2006; Idier 2001; Geman and Yang 1995; Geman and Reynolds 1992; Nikolova and Ng 2005; Champagnat and Idier 2004; Nikolova and Ng 2001).

Application to Certain Decomposition Problems

In this section, we demonstrate how to use our formulation described in the previous sections to solve certain image decomposition problems. The variational formulation for image decomposition problems is in the form of (8), where the input image $\mathbf{x} \in \mathbb{R}^n$ is decomposed into three components, which includes the geometrical part $\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2$, the texture part \mathbf{u}_1 , and the noise \mathbf{u}_2 . The regularization function J for the geometrical part $\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2$ is chosen to be the widely used Total Variation regularization function in order to preserve edges in the image. Here, we use the anisotropic Total Variation semi-norm (see, e.g., Darbon and Sigelle 2006; Darbon 2015) denoted by $|\cdot|_{TV}$. The noise is assumed to be Gaussian, and hence the data fidelity term f_2 is set to be the quadratic function. Many texture models have been proposed (see Aujol et al. 2003, 2005; Le Guen 2014; Winkler 2003 and the references in these papers). For instance, the indicator function of the unit ball

with respect to Meyer's norm is used in Aujol et al. (2003, 2005), and the ℓ^1 norm is used in Le Guen (2014). Note that each texture model has some pros and cons and, to our knowledge, it remains an open problem whether one specific texture model is better than the others. In this example, we combine different texture regularizations proposed in the literature by taking the minimum of the indicator function of the unit ball with respect to Meyer's norm and the ℓ^1 norm. In other words, we consider the following variational problem:

$$\min_{\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n} \left\{ J(\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2) + t_1 g\left(\frac{\mathbf{u}_1}{t_1}\right) + \frac{1}{2t_2} \|\mathbf{u}_2\|_2^2 \right\}, \quad (22)$$

where $J: \mathbb{R}^n \rightarrow \mathbb{R}$ and $g: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ are defined by

$$J(\mathbf{y}) := |\mathbf{y}|_{TV}, \quad g(\mathbf{y}) := \min\{J^*(\mathbf{y}), \|\mathbf{y}\|_1\},$$

for each $\mathbf{y} \in \mathbb{R}^n$. Problem (22) is equivalent to the following mixed discrete-continuous optimization problem

$$\min_{\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n} \min_{k \in \{1, 2\}} \left\{ J(\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2) + t_1 g_k\left(\frac{\mathbf{u}_1}{t_1}\right) + \frac{1}{2t_2} \|\mathbf{u}_2\|_2^2 \right\}, \quad (23)$$

where $g_1(\mathbf{y}) := J^*(\mathbf{y})$ and $g_2(\mathbf{y}) := \|\mathbf{y}\|_1$ for each $\mathbf{y} \in \mathbb{R}^n$. Note that solving mixed discrete-continuous optimization is hard in general (see Floudas and Pardalos 2009 for instance). However, we shall see that our proposed approach yields efficient optimization algorithms. Since the function g is the minimum of two convex functions, the problem (22) fits into our formulation, and can be solved using a similar idea as in (19) and (20). To be specific, define the two functions S_1 and S_2 by

$$S_1(\mathbf{x}, t_1, t_2) := \min_{\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n} \left\{ J(\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2) + t_1 J^*\left(\frac{\mathbf{u}_1}{t_1}\right) + \frac{1}{2t_2} \|\mathbf{u}_2\|_2^2 \right\}, \quad (24)$$

$$S_2(\mathbf{x}, t_1, t_2) := \min_{\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n} \left\{ J(\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2) + \|\mathbf{u}_1\|_1 + \frac{1}{2t_2} \|\mathbf{u}_2\|_2^2 \right\},$$

where the sets of the minimizers in the two minimization problems above are denoted by $M_1(\mathbf{x}, t_1, t_2)$ and $M_2(\mathbf{x}, t_1, t_2)$, respectively. Using a similar argument as in (19) and (20), we conclude that the minimal value in (22) equals $\min\{S_1(\mathbf{x}, t_1, t_2), S_2(\mathbf{x}, t_1, t_2)\}$, and the set of minimizers in (22), denoted by $M(\mathbf{x}, t_1, t_2)$, satisfies

$$M(\mathbf{x}, t_1, t_2) = \begin{cases} M_1(\mathbf{x}, t_1, t_2) & S_1(\mathbf{x}, t_1, t_2) < S_2(\mathbf{x}, t_1, t_2), \\ M_2(\mathbf{x}, t_1, t_2) & S_1(\mathbf{x}, t_1, t_2) > S_2(\mathbf{x}, t_1, t_2), \\ M_1(\mathbf{x}, t_1, t_2) \cup M_2(\mathbf{x}, t_1, t_2) & S_1(\mathbf{x}, t_1, t_2) = S_2(\mathbf{x}, t_1, t_2). \end{cases} \quad (25)$$

As a result, we solve the two minimization problems in (24) first, and then obtain the minimizers using (25) by comparing the minimal values $S_1(\mathbf{x}, t_1, t_2)$ and $S_2(\mathbf{x}, t_1, t_2)$.

Here, we present a numerical result. We solve the first optimization problem in (24) by a splitting method, where each subproblem can be solved using the proximal operator of the anisotropic Total Variation (for more details, see Darbon and Meng 2020). Similarly, a splitting method is used to split the second optimization problem in (24) to two subproblems, which are solved using the proximal operators of the anisotropic Total Variation and the ℓ^1 -norm, respectively. To compute the proximal point of the anisotropic Total Variation, the algorithm in Chambolle and Darbon (2009), Darbon and Sigelle (2006), and Hochbaum (2001) is adopted, and it computes the proximal point without numerical errors. The input image \mathbf{x} is the image “Barbara” shown in Fig. 1. The parameters are set to be $t_1 = 0.07$ and $t_2 = 0.01$. Let $(\mathbf{u}_1, \mathbf{u}_2) \in M_1(\mathbf{x}, t_1, t_2)$ and $(\mathbf{v}_1, \mathbf{v}_2) \in M_2(\mathbf{x}, t_1, t_2)$ be respectively the minimizers of the two minimization problems in (24) solved by the aforementioned splitting methods. We show these minimizers and the related images in Figs. 2 and 3. To be specific, the decomposition components $\mathbf{x} - \mathbf{u}_1 - \mathbf{u}_2$, $\mathbf{u}_1 + 0.5$, and $\mathbf{u}_2 + 0.5$ given by the first optimization problem in (24) are shown in Fig. 2a, b, and c, respectively. The decomposition components $\mathbf{x} - \mathbf{v}_1 - \mathbf{v}_2$, $\mathbf{v}_1 + 0.5$, and $\mathbf{v}_2 + 0.5$

Fig. 1 The input image \mathbf{x} (“Barbara”) in the example in section “[Application to Certain Decomposition Problems](#)”





Fig. 2 The minimizer of the first problem in (24). The output images $x - u_1 - u_2$, $u_1 + 0.5$, and $u_2 + 0.5$ are shown in (a), (b), and (c), respectively

given by the second optimization problem in (24) are shown in Fig. 3a, b, and c, respectively. We also compute the optimal values $S_1(x, t_1, t_2)$ and $S_2(x, t_1, t_2)$, and obtain

$$S_1(x, t_1, t_2) = 1832.81, \quad S_2(x, t_1, t_2) = 4171.33.$$

Since $S_1(x, t_1, t_2) < S_2(x, t_1, t_2)$, we conclude that (u_1, u_2) is a minimizer in the decomposition problem (22), and the minimal value equals 1832.81. In other words, the optimal decomposition given by (22) is shown in Fig. 2.

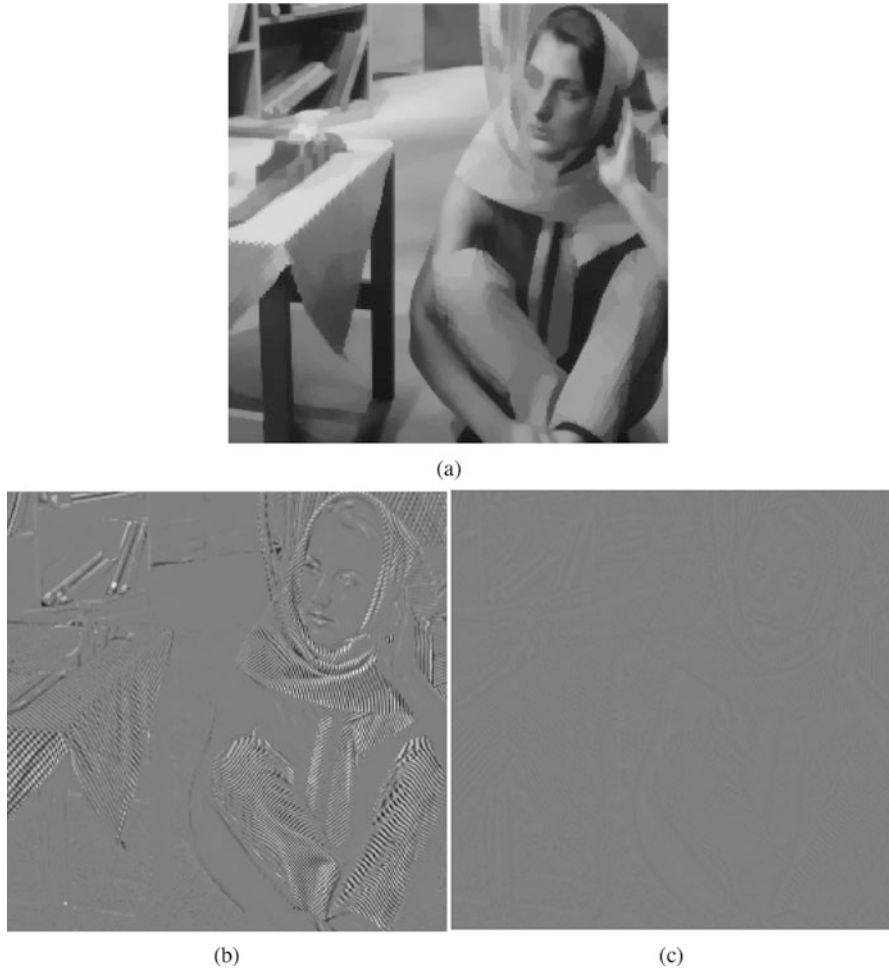


Fig. 3 The minimizer of the second problem in (24). The output images $x - v_1 - v_2$, $v_1 + 0.5$ and $v_2 + 0.5$ are shown in (a), (b) and (c), respectively

Viscous Hamilton-Jacobi PDEs and Bayesian Estimation

In contrast to variational approaches that frame imaging problems as optimization problems, Bayesian approaches frame them in a probabilistic framework. This framework combines observed data through a likelihood function (which models the noise corrupting the unknown image) and prior knowledge through a prior distribution (which models known properties of the image to reconstruct) to generate a posterior distribution from which an appropriate decision rule can select a meaningful image estimate. In this section, we present an analogue of the min-plus

algebra technique discussed in section “[Min-Plus Algebra for HJ PDEs and Certain Non-convex Regularizations](#)” for certain Bayesian posterior mean estimators.

Viscous HJ PDEs and Posterior Mean Estimators for Log-Concave Models

Consider the following class of Bayesian posterior distributions:

$$q(\mathbf{u}|\mathbf{x}, t, \epsilon) := \frac{e^{-\left(J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon}}{\int_{\mathbb{R}^n} e^{-\left(J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u}}, \quad (26)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the observed image with n pixels, and t and ϵ are positive parameters. The posterior distribution (26) is proportional to the product of a log-concave prior $\mathbf{u} \mapsto e^{-J(\mathbf{u})/\epsilon}$ (possibly improper) and a Gaussian likelihood function $\mathbf{u} \mapsto e^{-\frac{1}{2t\epsilon} \|\mathbf{x} - \mathbf{u}\|_2^2}$. This class of posterior distributions generates the family of Bayesian posterior mean estimators $\mathbf{u}_{PM} : \mathbb{R}^n \times (0, +\infty) \times (0, +\infty) \rightarrow \mathbb{R}^n$ defined by

$$\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) := \int_{\mathbb{R}^n} \mathbf{u} q(\mathbf{u}|\mathbf{x}, t, \epsilon) d\mathbf{u}. \quad (27)$$

These are Bayesian estimators because they minimize the mean squared error (Kay 1993, pages 344–345):

$$\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) = \arg \min_{\mathbf{u} \in \mathbb{R}^n} \int_{\mathbb{R}^n} \|\bar{\mathbf{u}} - \mathbf{u}\|_2^2 q(\bar{\mathbf{u}}|\mathbf{x}, t, \epsilon) d\bar{\mathbf{u}}. \quad (28)$$

They are frequently called minimum mean squared error estimators for this reason.

The class of posterior distributions (26) also generates the family of maximum a posteriori estimators $\mathbf{u}_{MAP} : \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}^n$ defined by

$$\mathbf{u}_{MAP}(\mathbf{x}, t) = \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\}, \quad (29)$$

where $\mathbf{u}_{MAP}(\mathbf{x}, t)$ is the mode of the posterior distribution (26). Note that the MAP estimator is also the minimizer of the solution (5) to the first-order HJ PDE (6) with Hamiltonian $H = \frac{1}{2} \|\cdot\|_2^2$ and initial data J .

There is a large body of literature on posterior mean estimators for image restoration problems (see e.g., Demoment 1989; Kay 1993; Winkler 2003). In particular, original connections between variational problems and Bayesian methods have been investigated in Louchet (2008), Louchet and Moisan (2013), Burger and Lucka (2014), Burger and Sciacchitano (2016), Gribonval (2011), Gribonval and Machart (2013), Gribonval and Nikolova (2018), and Darbon and Langlois

(2020). In particular, in Darbon and Langlois (2020), the authors described original connections between Bayesian posterior mean estimators and viscous HJ PDEs when $J \in \Gamma_0(\mathbb{R}^n)$ and the data fidelity term is Gaussian. We now briefly describe these connections here.

Consider the function $S_\epsilon : \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}$ defined by

$$S_\epsilon(\mathbf{x}, t) = -\epsilon \ln \left(\frac{1}{(2\pi t\epsilon)^{n/2}} \int_{\mathbb{R}^n} e^{-\left(J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u} \right), \tag{30}$$

which is proportional to the negative logarithm of the partition function of the posterior distribution (26). Under appropriate assumptions on the regularization term J (see Proposition 3), formula (30) corresponds to a Cole-Hopf transform (Evans 2010) and is the solution to the following viscous HJ PDE:

$$\begin{cases} \frac{\partial S_\epsilon}{\partial t}(\mathbf{x}, t) + \frac{1}{2} \|\nabla_{\mathbf{x}} S_\epsilon(\mathbf{x}, t)\|_2^2 = \frac{\epsilon}{2} \Delta_{\mathbf{x}} S_\epsilon(\mathbf{x}, t) & \mathbf{x} \in \mathbb{R}^n, t > 0, \\ S_\epsilon(\mathbf{x}, 0) = J(\mathbf{x}) & \mathbf{x} \in \mathbb{R}^n, \end{cases} \tag{31}$$

where J is the initial data. The solution to this PDE is also related to the first-order HJ PDE (6) when the Hamiltonian is $H = \frac{1}{2} \|\cdot\|_2^2$. The following proposition, which is given in Darbon and Langlois (2020), describes these connections.

Proposition 3. *Assume $J \in \Gamma_0(\mathbb{R}^n)$, $\text{int}(\text{dom } J) \neq \emptyset$, and $\inf_{\mathbf{u} \in \mathbb{R}^n} J(\mathbf{u}) = 0$. Then for every $\epsilon > 0$, the unique smooth solution $S_\epsilon : \mathbb{R}^n \times (0, +\infty) \rightarrow (0, +\infty)$ to the HJ PDE (31) is given by formula (30), where $(\mathbf{x}, t) \mapsto S_\epsilon(\mathbf{x}, t) - \frac{n\epsilon}{2} \ln t$ is jointly convex. Moreover, for each $\mathbf{x} \in \mathbb{R}^n$, $t > 0$, and $\epsilon > 0$, the posterior mean estimator (27) and minimum mean squared error in (28) (with $\mathbf{u} = \mathbf{u}_{PM}(\mathbf{x}, t, \epsilon)$) satisfy, respectively, the formulas:*

$$\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) = \mathbf{x} - t \nabla_{\mathbf{x}} S_\epsilon(\mathbf{x}, t) \tag{32}$$

and

$$\int_{\mathbb{R}^n} \|\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) - \mathbf{u}\|_2^2 q(\mathbf{u} | (\mathbf{x}, t, \epsilon)) d\mathbf{u} = nt\epsilon - t^2 \epsilon \Delta_{\mathbf{x}} S_\epsilon(\mathbf{x}, t). \tag{33}$$

In addition, for every $\mathbf{x} \in \mathbb{R}^n$ and $t > 0$, the limits of $\lim_{\epsilon \rightarrow 0} S_\epsilon(\mathbf{x}, t)$ and $\lim_{\substack{\epsilon \rightarrow 0 \\ \epsilon > 0}} \mathbf{u}_{PM}(\mathbf{x}, t, \epsilon)$ exist and converge uniformly over every compact set of $\mathbb{R}^n \times (0, +\infty)$ in (\mathbf{x}, t) . Specifically, we have

$$\lim_{\substack{\epsilon \rightarrow 0 \\ \epsilon > 0}} S_\epsilon(\mathbf{x}, t) = \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\}, \tag{34}$$

where the right-hand side solves uniquely the first-order HJ PDE (6) with Hamiltonian $H = \frac{1}{2} \|\cdot\|_2^2$ and initial data J , and

$$\lim_{\substack{\epsilon \rightarrow 0 \\ \epsilon > 0}} \mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) = \arg \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ J(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\}. \quad (35)$$

Under convexity assumptions on J , the representation formulas (32) and (33) relate the posterior mean estimate and the minimum mean squared error to the spatial gradient and Laplacian of the solution to the viscous HJ PDE (31), respectively. Hence one can compute the posterior mean estimator and minimum mean squared error using the spatial gradient $\nabla_{\mathbf{x}} \mathcal{S}_{\epsilon}(\mathbf{x}, t)$ and the Laplacian $\Delta_{\mathbf{x}} \mathcal{S}_{\epsilon}(\mathbf{x}, t)$ of the solution to the HJ PDE (31), respectively, or vice versa by computing the posterior mean and minimum mean squared error using, for instance, Markov chain Monte Carlo sampling strategies.

The limit (35) shows that the posterior mean $\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon)$ converges to the maximum a posteriori $\mathbf{u}_{MAP}(\mathbf{x}, t)$ as the parameter $\epsilon \rightarrow 0$. A rough estimate of the squared Euclidean distance between the posterior mean estimator (27) and the maximum a posteriori (29) in terms of the parameters t and ϵ is given by

$$\|\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) - \mathbf{u}_{MAP}(\mathbf{x}, t)\|_2^2 \leq nt\epsilon. \quad (36)$$

On Viscous HJ PDEs with Certain Non-log-Concave Priors

So far, we have assumed that the regularization term J in the posterior distribution (26) and Proposition 3 is convex. Here, we consider an analogue of the min-plus algebra technique designed for certain first-order HJ PDEs tailed to viscous HJ PDEs, which will enable us to derive representation formulas for posterior mean estimators of the form of (27) whose priors are sums of log-concave priors, i.e., to certain mixture distributions.

Remember that the min-plus algebra technique for first-order HJ PDEs described in section “[Min-Plus Algebra for HJ PDEs and Certain Non-convex Regularizations](#)” involves initial data of the form $\min_{i \in \{1, \dots, m\}} J_i(\mathbf{x})$ where each $J_i: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is convex. Consider now initial data of the form

$$J(\mathbf{x}) = -\epsilon \ln \left(\sum_{i=1}^m e^{-J_i(\mathbf{x})/\epsilon} \right). \quad (37)$$

Note that formula (37) approximates the non-convex term (12) in that

$$\lim_{\substack{\epsilon \rightarrow 0 \\ \epsilon > 0}} -\epsilon \ln \left(\sum_{i=1}^m e^{-J_i(\mathbf{x})/\epsilon} \right) = \min_{i \in \{1, \dots, m\}} J_i(\mathbf{x}) \text{ for each } \mathbf{x} \in \mathbb{R}^n.$$

Now, assume $\text{int}(\text{dom } J_i) \neq \emptyset$ for each $i \in \{1, \dots, m\}$, and let

$$S_{i,\epsilon}(\mathbf{x}, t) = -\epsilon \ln \left(\frac{1}{(2\pi t\epsilon)^{n/2}} \int_{\mathbb{R}^n} e^{-\left(J_i(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u} \right),$$

and

$$\mathbf{u}_{i,PM}(\mathbf{x}, t, \epsilon) = \frac{\int_{\mathbb{R}^n} \mathbf{u} e^{-\left(J_i(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u}}{\int_{\mathbb{R}^n} e^{-\left(J_i(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u}}$$

denote, respectively, the solution to the viscous HJ PDE (31) with initial data J_i and its associated posterior mean. Then, a short calculation shows that for every $\epsilon > 0$, the function $S_\epsilon(\mathbf{x}, t): \mathbb{R}^n \times (0, +\infty) \rightarrow \mathbb{R}$ defined by

$$\begin{aligned} S_\epsilon(\mathbf{x}, t) &= -\epsilon \ln \left(\sum_{i=1}^m \frac{1}{(2\pi t\epsilon)^{n/2}} \int_{\mathbb{R}^n} e^{-\left(J_i(\mathbf{u}) + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2\right)/\epsilon} d\mathbf{u} \right) \\ &= -\epsilon \ln \left(\sum_{i=1}^m e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon} \right) \end{aligned} \tag{38}$$

is the unique smooth solution to the viscous HJ PDE (31) with initial data (37). As stated in section “Viscous HJ PDEs and Posterior Mean Estimators for Log-Concave Models”, the posterior mean estimate $\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon)$ is given by the representation formula:

$$\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) = \mathbf{x} - t \nabla_{\mathbf{x}} S_\epsilon(\mathbf{x}, t), \tag{39}$$

which can be expressed in terms of the solutions $S_{i,\epsilon}(\mathbf{x}, t)$, their spatial gradients $\nabla_{\mathbf{x}} S_{i,\epsilon}(\mathbf{x}, t)$, and posterior mean estimates $\mathbf{u}_{i,PM}(\mathbf{x}, t, \epsilon)$ as the weighted sums

$$\begin{aligned} \mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) &= \mathbf{x} - t \left(\frac{\sum_{i=1}^m \nabla_{\mathbf{x}} S_{i,\epsilon}(\mathbf{x}, t) e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon}}{\sum_{i=1}^m e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon}} \right) \\ &= \frac{\sum_{i=1}^m \mathbf{u}_{i,PM}(\mathbf{x}, t, \epsilon) e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon}}{\sum_{i=1}^m e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon}}. \end{aligned} \tag{40}$$

As an application of this result, we consider the problem of classifying a noisy image $\mathbf{x} \in \mathbb{R}^n$ using a Gaussian mixture model (Duda et al. 2012): Suppose $J_i(\mathbf{u}) = \frac{1}{2\sigma_i^2} \|\mathbf{u} - \boldsymbol{\mu}_i\|_2^2$, where $\boldsymbol{\mu}_i \in \mathbb{R}^n$ and $\sigma_i > 0$. The regularized minimization problem (13) with quadratic data fidelity term $H = \frac{1}{2} \|\cdot\|_2^2$ is given by

$$\begin{aligned}
S_0(\mathbf{x}, t) &= \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \min_{i \in \{1, \dots, m\}} \left\{ \frac{1}{2\sigma_i^2} \|\mathbf{u} - \boldsymbol{\mu}_i\|_2^2 + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\} \right\} \\
&= \min_{i \in \{1, \dots, m\}} \left\{ \min_{\mathbf{u} \in \mathbb{R}^n} \left\{ \frac{1}{2\sigma_i^2} \|\mathbf{u} - \boldsymbol{\mu}_i\|_2^2 + \frac{1}{2t} \|\mathbf{x} - \mathbf{u}\|_2^2 \right\} \right\} \\
&= \min_{i \in \{1, \dots, m\}} \left\{ \frac{1}{2(\sigma_i^2 + t)} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2 \right\}.
\end{aligned} \tag{41}$$

Letting $I(\mathbf{x}, t) = \arg \min_{i \in \{1, \dots, m\}} \left\{ \frac{1}{2(\sigma_i^2 + t)} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2 \right\}$, the MAP estimator is then the collection:

$$\mathbf{u}_{MAP}(\mathbf{x}, t) = \bigcup_{i \in I(\mathbf{x}, t)} \left\{ \frac{\sigma_i^2 \mathbf{x} + t \boldsymbol{\mu}_i}{\sigma_i^2 + t} \right\}.$$

Consider now the initial data (37):

$$J(\mathbf{u}) = -\epsilon \ln \left(\sum_{i=1}^m e^{-\frac{1}{2\sigma_i^2 \epsilon} \|\mathbf{u} - \boldsymbol{\mu}_i\|_2^2} \right).$$

The solution $S_\epsilon(\mathbf{x}, t)$ to the viscous HJ PDE (31) with initial data $J(\mathbf{x})$ is given by formula (38), which in this case can be computed analytically:

$$S_\epsilon(\mathbf{x}, t) = -\epsilon \ln \left(\sum_{i=1}^m \left(\frac{\sigma_i^2}{\sigma_i^2 + t} \right)^{n/2} e^{-\frac{1}{2(\sigma_i^2 + t)\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2} \right). \tag{42}$$

Since $e^{-S_{i,\epsilon}(\mathbf{x}, t)/\epsilon} = \left(\frac{\sigma_i^2}{\sigma_i^2 + t} \right)^{n/2} e^{-\frac{1}{2(\sigma_i^2 + t)\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2}$, we can write the corresponding posterior mean estimator (40) using the representation formulas (39) and (40):

$$\begin{aligned}
\mathbf{u}_{PM}(\mathbf{x}, t, \epsilon) &= \mathbf{x} - t \nabla_{\mathbf{x}} S_\epsilon(\mathbf{x}, t) \\
&= \frac{\sum_{i=1}^m \left(\frac{\sigma_i^2 \mathbf{x} + t \boldsymbol{\mu}_i}{\sigma_i^2 + t} \right) \left(\frac{\sigma_i^2}{\sigma_i^2 + t} \right)^{n/2} e^{-\frac{1}{2(\sigma_i^2 + t)\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2}}{\sum_{i=1}^m \left(\frac{\sigma_i^2}{\sigma_i^2 + t} \right)^{n/2} e^{-\frac{1}{2(\sigma_i^2 + t)\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2}}.
\end{aligned} \tag{43}$$

Conclusion

In this chapter, we reviewed the connections of single-time HJ PDEs with image denoising models and the connections of multi-time HJ PDEs with image decomposition models under convexity assumptions. Specifically, under some assumptions, the minimizers of these optimization problems can be computed using the spatial gradient of the solution to the corresponding HJ PDEs. We also proposed a min-plus algebra technique to cope with certain non-convex regularization terms in imaging sciences problems. This suggests that certain non-convex optimization problem can be solved by computing several convex subproblems. For instance, if the denoising model (1) or the image decomposition model (8) involves a non-convex regularization term J that can be expressed as the minimum of m convex subproblems in the form of (12), then the minimizer of these non-convex problems can be solved using formulas (14) and (20). However, when m in (12) is too large, it is generally impractical to solve (14) and (20) using this min-plus technique because it involves solving too many HJ PDEs. However, our formulation yields practical numerical optimization algorithms for certain image denoising and decomposition problems.

We also reviewed connections between viscous HJ PDEs and a class of Bayesian methods and posterior mean estimators when the data fidelity term is Gaussian and the prior distribution is log-concave. Under some assumptions, the posterior mean estimator (27) and minimum mean squared error in (28) associated to the posterior distribution (26) can be computed using the spatial gradient and Laplacian of the solution to the viscous HJ PDE (31) via the representation formulas (32) and (33), respectively. We also proposed an analogue of the min-plus algebra technique designed for certain first-order HJ PDEs tailored to viscous HJ PDEs that enable us to compute posterior mean estimators with Gaussian fidelity term and prior that involves the sum of m log-concave priors, i.e., to certain mixture models. The corresponding posterior mean estimator with non-convex regularization J of the form of (37) can then be computed using the representation formulas (40) and posterior mean estimators (27) with convex regularization terms J_i .

Let us emphasize again that the proposed min-plus algebra technique for computations directly applies only for moderate m in (12). It would be of great interest to identify classes of non-convex regularizations for which novel numerical algorithms based on the min-plus algebra technique would not require to compute solutions to all m convex subproblems. To our knowledge, there is no available result in the literature on this matter.

Acknowledgments This work was funded by NSF 1820821.

References

- Akian, M., Bapat, R., Gaubert, S.: Max-plus algebra. In: Handbook of Linear Algebra, 39 (2006)
- Akian, M., Gaubert, S., Lakhoua, A.: The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM J. Control. Optim.* **47**, 817–848 (2008)

- Allain, M., Idier, J., Goussard, Y.: On global and local convergence of half-quadratic algorithms. *IEEE Trans. Image Process.* **15**, 1130–1142 (2006)
- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing*. Springer (2002)
- Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition application to SAR images. In: L.D. Griffin, Lillholm, M. (eds.) *Scale Space Methods in Computer Vision*. Springer, Berlin/Heidelberg, pp. 297–312 (2003)
- Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vision* **22**, 71–88 (2005)
- Bardi, M., Capuzzo-Dolcetta, I.: *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Systems & Control: Foundations & Applications, Birkhäuser Boston, Inc., Boston (1997). With appendices by Maurizio Falcone and Pierpaolo Soravia
- Bardi, M., Evans, L.: On Hopf's formulas for solutions of Hamilton-Jacobi equations. *Nonlinear Anal. Theory Methods Appl.* **8**, 1373–1381 (1984)
- Barles, G.: *Solutions de viscosité des équations de Hamilton-Jacobi*. Mathématiques et Applications. Springer, Berlin/Heidelberg (1994)
- Barron, E., Evans, L., Jensen, R.: Viscosity solutions of Isaacs' equations and differential games with Lipschitz controls. *J. Differ. Equ.* **53**, 213–233 (1984)
- Bouman, C., Sauer, K.: A generalized gaussian image model for edge-preserving map estimation. *IEEE Trans. Signal Process.* **2**, 296–310 (1993)
- Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 1222–1239 (2001)
- Burger, M., Lucka, F.: Maximum a posteriori estimates in linear inverse problems with log-concave priors are proper bayes estimators. *Inverse Probl.* **30**, 114004 (2014)
- Burger, Y.D.M., Sciacchitano, F.: Bregman cost for non-gaussian noise. arXiv preprint arXiv:1608.07483 (2016)
- Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximum flows. *Int. J. Comput. Vis.* **84**, 288–307 (2009)
- Chambolle, A., Novaga, M., Cremers, D., Pock, T.: An introduction to total variation for image analysis. In: *Theoretical Foundations and Numerical Methods for Sparse Recovery*, De Gruyter (2010)
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
- Champagnat, F., Idier, J.: A connection between half-quadratic criteria and em algorithms. *IEEE Signal Processing Lett.* **11**, 709–712 (2004)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**, 1632–1648 (2006)
- Chan, T.F., Shen, J.: *Image processing and analysis*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2005). Variational, PDE, wavelet, and stochastic methods
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**, 266–277 (2001)
- Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.* **6**, 298–311 (1997)
- Crandall, M.G., Ishii, H., Lions, P.-L.: User's guide to viscosity solutions of second order partial differential equations. *Bull. Am. Math. Soc.* **27**, 1–67 (1992)
- Darbon, J.: On convex finite-dimensional variational methods in imaging sciences and Hamilton-Jacobi equations. *SIAM J. Imag. Sci.* **8**, 2268–2293 (2015)
- Darbon, J., Ciril, I., Marquina, A., Chan, T.F., Osher, S.: A note on the bregmanized total variation and dual forms. In: 2009 16th IEEE International Conference on Image Processing (ICIP), Nov 2009, pp. 2965–2968
- Darbon, J., Langlois, G.P.: On Bayesian posterior mean estimators in imaging sciences and Hamilton-Jacobi partial differential equations. arXiv preprint arXiv: 2003.05572 (2020)
- Darbon, J., Meng, T.: On decomposition models in imaging sciences and multi-time Hamilton-Jacobi partial differential equations. *SIAM Journal on Imaging Sciences.* **13**(2), 971–1014 (2020). <https://doi.org/10.1137/19M1266332>

- Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part I: Fast and exact optimization. *J. Math. Imaging Vision* **26**, 261–276 (2006)
- Demoment, G.: Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Trans. Acoust. Speech Signal Process.* **37**, 2024–2036 (1989)
- Dou, Z., Song, M., Gao, K., Jiang, Z.: Image smoothing via truncated total variation. *IEEE Access* **5**, 27337–27344 (2017)
- Dower, P.M., McEneaney, W.M., Zhang, H.: Max-plus fundamental solution semigroups for optimal control problems. In: 2015 Proceedings of the Conference on Control and its Applications. SIAM, 2015, pp. 368–375
- Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley (2012)
- Evans, L.C.: *Partial differential equations*, vol. 19 of Graduate Studies in Mathematics, 2nd edn. American Mathematical Society, Providence (2010)
- Fleming, W., McEneaney, W.: A max-plus-based algorithm for a Hamilton–Jacobi–Bellman equation of nonlinear filtering. *SIAM J. Control. Optim.* **38**, 683–710 (2000)
- Fleming, W.H., Soner, H.M.: *Controlled Markov Processes and Viscosity Solutions*, vol. 25. Springer Science & Business Media (2006)
- Floudas, C.A., Pardalos, P.M. (eds.): *Encyclopedia of Optimization*, 2nd edn. (2009)
- Gaubert, S., McEneaney, W., Qu, Z.: Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference. IEEE, 2011, pp. 1054–1061
- Geman, D., Yang, C.: Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Process.* **4**, 932–946 (1995)
- Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 367–383 (1992)
- Gribonval, R.: Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Trans. Signal Process.* **59**, 2405–2410 (2011)
- Gribonval, R., Machart, P.: Reconciling “priors” & “priors” without prejudice? In: *Advances in Neural Information Processing Systems*, 2013, pp. 2193–2201
- Gribonval, R., Nikolova, M.: On bayesian estimation and proximity operators, arXiv preprint arXiv:1807.04021 (2018)
- Hochbaum, D.S.: An efficient algorithm for image segmentation, Markov random fields and related problems. *J. ACM* **48**, 686–701 (2001)
- Hopf, E.: Generalized solutions of non-linear equations of first order. *J. Math. Mech.* **14**, 951–973 (1965)
- Idier, J.: Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *IEEE Trans. Image Process.* **10**, 1001–1009 (2001)
- Kay, S.M.: *Fundamentals of Statistical Signal Processing*. Prentice Hall PTR (1993)
- Kolokoltsov, V.N., Maslov, V.P.: *Idempotent analysis and its applications*, vol. 401 of Mathematics and its Applications. Kluwer Academic Publishers Group, Dordrecht (1997) Translation of *Idempotent analysis and its application in optimal control* (Russian), “Nauka” Moscow, 1994 [MR1375021 (97d:49031)], Translated by V. E. Nazaikinskii, With an appendix by Pierre Del Moral
- Le Guen, V.: Cartoon + Texture Image Decomposition by the TV-L1yModel. *Image Process. Line* **4**, 204–219 (2014)
- Likas, A.C., Galatsanos, N.P.: A variational approach for bayesian blind image deconvolution. *IEEE Trans. Signal Process.* **52**, 2222–2233 (2004)
- Lions, P.L., Rochet, J.-C.: Hopf formula and multitime Hamilton–Jacobi equations. *Proc. Am. Math. Soc.* **96**, 79–84 (1986)
- Louchet, C.: Modèles variationnels et bayésiens pour le débruitage d’images: de la variation totale vers les moyennes non-locales. Ph.D. thesis, Université René Descartes-Paris V (2008)
- Louchet, C., Moisan, L.: Posterior expectation of the total variation model: properties and experiments. *SIAM J. Imaging Sci.* **6**, 2640–2684 (2013)

- McEneaney, W.: Max-plus methods for nonlinear control and estimation. Springer Science & Business Media (2006)
- McEneaney, W.: A curse-of-dimensionality-free numerical method for solution of certain HJB PDEs. *SIAM J. Control. Optim.* **46**, 1239–1276 (2007)
- McEneaney, W.M., Deshpande, A., Gaubert, S.: Curse-of-complexity attenuation in the curse-of-dimensionality-free method for HJB PDEs. In: 2008 American Control Conference. IEEE, 2008, pp. 4684–4690
- McEneaney, W.M., Kluberg, L.J.: Convergence rate for a curse-of-dimensionality-free method for a class of HJB PDEs. *SIAM J. Control. Optim.* **48**, 3052–3079 (2009)
- Nikolova, M., Chan, R.H.: The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Trans. Image Process.* **16**, 1623–1627 (2007)
- Nikolova, M., Ng, M.: Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning. In: Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205), vol. 1. IEEE, 2001, pp. 277–280
- Nikolova, M., Ng, M.K.: Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM J. Sci. Comput.* **27**, 937–966 (2005)
- Osher, S., A. Solé, and Vese, L.: Image decomposition and restoration using total variation minimization and the H^{-1} norm, *Multiscale Modeling & Simulation*, 1 (2003), pp. 349–370.
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational methods in imaging, vol. 167 of Applied Mathematical Sciences. Springer, New York (2009)
- Tho, N.: Hopf-Lax-Oleinik type formula for multi-time Hamilton-Jacobi equations. *Acta Math. Vietnamica* **30**, 275–287 (2005)
- Vese, L.A., Le Guyader, C.: Variational methods in image processing, Chapman & Hall/CRC Mathematical and Computational Imaging Sciences. CRC Press, Boca Raton (2016)
- Winkler, G.: Image Analysis, Random Fields and Dynamic Monte Carlo Methods. Applications of Mathematics. Springer, 2nd edn. (2003)



Multi-modality Imaging with Structure-Promoting Regularizers

7

Matthias J. Ehrhardt

Contents

Introduction	236
Application Examples	236
Variational Regularization	240
Contributions	241
Related Work	241
Mathematical Models for Structural Similarity	243
Measuring Structural Similarity	244
Structure-Promoting Regularizers	245
Isotropic Models	245
Anisotropic Models	247
Algorithmic Solution	250
Algorithm	250
Prewhitening	251
Numerical Comparison	252
Software, Data, and Parameters	252
Numerical Results	253
Discussion on Computational Cost	256
Conclusions	261
Open Problems	262
References	266

Abstract

Imaging with multiple modalities or multiple channels is becoming increasingly important for our modern society. A key tool for understanding and early diagnosis of cancer and dementia is PET-MR, a combined positron emission

M. J. Ehrhardt (✉)
Institute for Mathematical Innovation, University of Bath, Bath, UK
e-mail: m.ehrhardt@bath.ac.uk

tomography and magnetic resonance imaging scanner which can simultaneously acquire functional and anatomical data. Similarly, in remote sensing, while hyperspectral sensors may allow to characterize and distinguish materials, digital cameras offer high spatial resolution to delineate objects. In both of these examples, the imaging modalities can be considered individually or jointly. In this chapter we discuss mathematical approaches which allow combining information from several imaging modalities so that multi-modality imaging can be more than just the sum of its components.

Introduction

Many tasks in almost all scientific fields can be posed as an inverse problem of the form

$$Ku = f \tag{1}$$

where K is a mathematical model that connects an unknown quantity of interest u to measured data f . The task is to recover u from data f under the model K . In practice this task is difficult because of measurement errors in the data f and inaccuracies in the model K . Moreover, in many cases the model (1) lacks information we have at hand about the unknown quantity u such as its regularity. In this chapter we are interested in the situation when we have a priori knowledge about the “structure” of u from a second measurement v which we want to exploit in the inversion. Throughout this chapter we will refer to v as the *side information*. Intuitively, this is the case when u and v describe different properties of the same geometry (in medicine: anatomy). We will be more precise in section “[Mathematical Models for Structural Similarity](#)” where we discuss mathematical models for structural similarity. The two notions we will discuss in detail are that the edges of the two images u and v have similar (1) locations (Arridge et al. 2008; Bresson and Chan 2008; Haber and Holtzman-Gazit 2013; Knoll et al. 2014; Ehrhardt et al. 2015) and (2) directions (Gallardo and Meju 2003, 2004; Haber and Holtzman-Gazit 2013; Ehrhardt and Arridge 2014; Ehrhardt et al. 2015; Rigie and La Riviere 2015; Ehrhardt and Betcke 2016; Ehrhardt et al. 2016; Knoll et al. 2016; Schramm et al. 2017; Bathke et al. 2017; Bungert et al. 2018; Kolehmainen et al. 2019). Real-world examples for these mathematical models are numerous as we will see in the next section.

Application Examples

Historically the first application where information from several modalities was combined was positron emission tomography (PET) and magnetic resonance imaging (MRI) in the early 1990s (Leahy and Yan 1991). Sharing information between two different imaging modalities is motivated by the fact that all images



Fig. 1 PET-MR and PET-CT. A low resolution functional PET image (left) is to be reconstructed with the help of an anatomical MRI (middle) or CT image (right). As is evident from the images, all three images share many edges due to the same underlying anatomy. Note that the high soft tissue contrast in MRI makes it favorable over CT for this application. (Images courtesy of P. Markiewicz and J. Schott)

will be highly influenced by the same underlying anatomy; see Fig. 1. Since single-photon emission computed tomography (SPECT) imaging is both mathematically and physically similar to PET imaging, most of the proposed models can be directly translated and often models are proposed for both modalities simultaneously; see, e.g., Bowsher et al. (1996), Rangarajan et al. (2000), Chan et al. (2007) and Nuyts (4154). Over the years there always has been research in this direction (see, e.g., Bowsher et al. (1996), Rangarajan et al. (2000), Comtat et al. (2002), Bowsher et al. (2004), Baete et al. (2004), Chan et al. (2007), Chan et al. (2009), Tang and Rahmim (2009), Bousse et al. (2010), Pedemonte et al. (2011), Somayajula et al. (2011), Cheng-Liao and Qi (2011), Vunckx et al. (2012), Kazantsev et al. (2012), Bousse et al. (2012) and Bai et al. (2013)), which was intensified with the advent of the first simultaneous PET-MR scanner in 2011 (Delso et al. 2011); see, e.g., (Knoll et al. 2014; Ehrhardt et al. 2014, 2015; Tang and Rahmim 2015; Ehrhardt et al. 2016; Knoll et al. 2016; Schramm et al. 2017; Mehranian et al. 2018, 2017; Tsai et al. 2018; Zhang and Zhang 2018; Ehrhardt et al. 2019; Deidda et al. 2019).

The same motivation applies to other medical imaging techniques, for example, multi-contrast MRI; see, e.g., Bilgic et al. (2011), Ehrhardt and Betcke (2016), Huang et al. (2014), Sodickson et al. (2015), Song et al. (2018) and Xiang et al. (2019). In multi-contrast MRI multiple acquisition sequences are used to acquire data of the same patient; see Fig. 2 for a T_1 - and a T_2 -weighted image with shared anatomy. Other special cases are the combination of anatomical MRI (e.g., T_1 -weighted) and magnetic particle imaging (Bathke et al. 2017), functional MRI (fMRI) and anatomical MRI (Rasch et al. 2018b), as well as anatomical (^1H) and fluorinated gas (^{19}F) MRI (Obert et al. 2020). A related imaging task is quantitative MRI (such as Magnetic Resonance Fingerprinting Ma et al. 7440) (Davies et al. 2013; Tang et al. 2018; Dong et al. 2019; Golbabaee et al. 2020) where one aims to reconstruct quantitative maps of tissue parameters (e.g., T_1 , T_2 , proton density, off-resonance frequency), but regularizers coupling these maps have not been used to date. The idea to couple channels has also been used for parallel MRI (Chen et al. 2013).

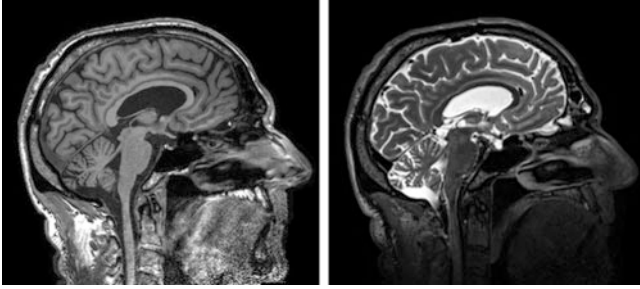


Fig. 2 Multi-contrast MRI. The same MRI scanner can produce different images depending on the acquisition sequence such as T_1 -weighted (left) and T_2 -weighted images (right). (Images courtesy of N. Burgos)



Fig. 3 Color imaging. The color image (left) is composed of three color channels (right) all of which show similar edges due to the same scenery. (Images courtesy of M. Ehrhardt)

Starting from the 1990s, mathematical models were developed that make use of the expected correlations between color channels of RGB images (Sapiro and Ringach 1996; Blomgren and Chan 1998; Sochen et al. 1998); see Fig. 3. Research in this field is still very active today; see, e.g., Tschumperlé and Deriche (2005), Bresson and Chan (2008), Goldluecke et al. (2012), Holt (2014), Ehrhardt and Arridge (2014), and Möller et al. (2014).

In remote sensing observations are often available from multiple sensors either mounted on a plane or on a satellite. For example, a hyperspectral camera with low spatial resolution and a digital camera with higher spatial resolution may be used simultaneously; see Fig. 4. This situation naturally invites for the fusion of information; see Ballester et al. (2006), Möller et al. (2012), Fang et al. (2013), Loncan et al. (2015), Yokoya et al. (2017), Duran et al. (2017), Bungert et al. (2018), Bungert et al. (2018) and references therein. In some situations the response of the cameras to certain wavelengths is (assumed to be) known such that the data can be fused making use of this knowledge. This is commonly referred to as *pansharpening* (Loncan et al. 2015; Yokoya et al. 2017; Duran et al. 2017). It is important to note that this assumption is sometimes not fulfilled, and many of the aforementioned algorithms are flexible enough to fuse data in this more general situation.

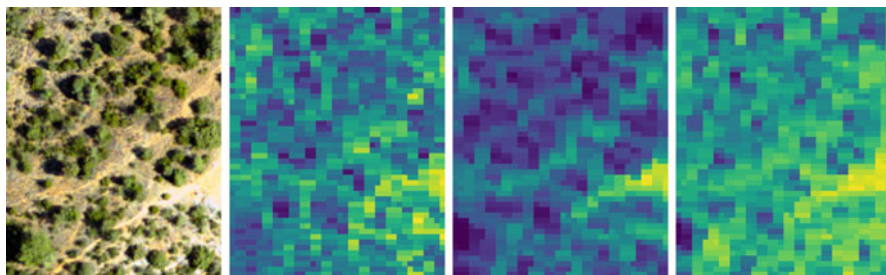


Fig. 4 Hyperspectral imaging + photography. A nowadays common scenario is that multiple cameras are mounted on a plane or satellite for remote sensing. While one camera carries spectral information (right), the other has high spatial resolution (left). (Images courtesy of D. Coomes)

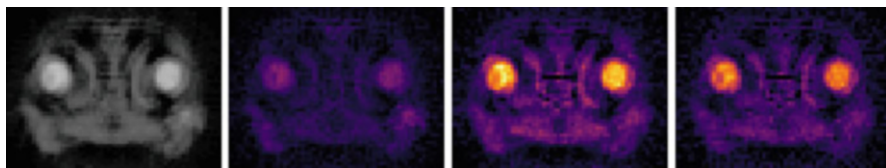


Fig. 5 Spectral CT. Standard (white-beam) CT on the left and three channels (28, 34, and 39 keV) of spectral CT on the right of an iodine-stained lizard head reconstructed by CIL (Ametova et al. 2019). The spectral channels clearly show a large increase in intensity from 28 to 34 keV, thereby revealing the presence, location, and concentration of iodine. (Images courtesy of J. Jorgensen and R. Warr)

Dual and spectral computed tomography (CT) is becoming increasingly popular in (bio-) medical imaging and material sciences due to its ability to distinguish different materials which would not be possible using a single energy; see Fig. 5. Since the energy channels have a very different signal-to-noise ratio, coupling them within the reconstruction allows to transfer information from high signal to low signal channels (Rigie and La Riviere 2015; Foygel Barber et al. 2016; Rigie et al. 2017; Kazantsev et al. 2018).

In geophysics, the coupling between modalities has been used to model similarity between electrical resistivity and seismic velocity (Gallardo and Meju 2003, 2004), estimating conductivity from multi-frequency data (Haber and Oldenburg 1997), inverting gravity and seismic tomography (Haber and Oldenburg 1997), and controlled-source electromagnetic resistivity inversion (Meju et al. 2019). For an overview and more details on examples in geophysics, see in Gallardo and Meju (2011) and Haber and Holtzman-Gazit (2013) and references therein.

Ideas from multi-modality imaging have recently also been used for art restoration. When a canvas is painted on both sides, an x-ray image shows the superposition of both paintings. The x-ray information can then be separated using photos of both sides of the canvas (Deligiannis et al. 2017).

Other examples that were considered in the literature are combining anatomical information and electrical impedance tomography (Kaipio et al. 1999; Kolehmainen

et al. 2019), CT and MRI (Xi et al. 2015), photoacoustic and optical coherence tomography (Elbau et al. 2018), x-ray fluorescence and transmission tomography (Di et al. 2016), and various channels in multi-modal electron tomography (Huber et al. 2019). The combination of various imaging modalities into one system may eventually lead to what is sometimes referred to as *omni-tomography* (Wang et al. 2012).

Image reconstruction with side information is mathematically similar to multi-modal image registration, and thus it is not surprising that both fields share a lot of mathematical models; see, e.g., Wells III et al. (1996), Maes et al. (1997), Pluim et al. (2000), and Haber and Modersitzki (2006).

Variational Regularization

Inverse problems of the form (1) can be solved using variational regularization, i.e., framed as the optimization problem

$$u_\alpha \in \arg \min_u \mathcal{D}(Ku, f) + \alpha \mathcal{R}(u). \quad (2)$$

Here the *data fidelity* $\mathcal{D} : Y \times Y \rightarrow \mathbb{R}_\infty := \mathbb{R} \cup \{\infty\}$ measures how close the estimated data Ku fits the acquired data f . The *regularizer* (also referred to as the *prior*) $\mathcal{R} : X \rightarrow \mathbb{R}_\infty$ defines which properties of the image u we favor and which we do not. The trade-off between data fitting and regularization can be chosen using the *regularization parameter* $\alpha > 0$. Problems of this form have been extensively studied; see, for instance, (Engl et al. 1996; Scherzer et al. 2008; Ito and Jin 2014; Bredies and Lorenz 2018; Benning and Burger 2018) and references therein.

Three popular regularizers for imaging are the *squared H^1 -semi norm* (H^1), the *total variation* (TV) (Rudin et al. 1992; Burger and Osher 2013), and the *total generalized variation* (TGV) (Bredies et al. 2010; Bredies and Holler 2014, 2015). It is common to model images as functions $u : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$. If u is smooth enough, then these regularizers are defined as

$$H^1(u) = \int_{\Omega} |\nabla u(x)|^2 dx \quad (3)$$

$$\text{TV}(u) = \int_{\Omega} |\nabla u(x)| dx \quad (4)$$

$$\text{TGV}(u) = \inf_{\zeta} \int_{\Omega} |\nabla u(x) - \zeta(x)| + \beta |E\zeta(x)| dx. \quad (5)$$

Here $\nabla u : \Omega \rightarrow \mathbb{R}^d$, $[\nabla u]_i = \partial_i u$ denotes the gradient of u , $E\zeta : \Omega \rightarrow \mathbb{R}^{d \times d}$, $[E\zeta]_{i,j} = (\partial_i \zeta_j + \partial_j \zeta_i)/2$ denotes the symmetrized gradient of a vector-field $\zeta : \Omega \rightarrow \mathbb{R}^d$ (see Bredies and Holler (2015) for more details), and $|\cdot|$ denotes the Euclidean/Frobenius norm. For TV and TGV it is of interest to develop other

formulations which are well-defined even when u is not smooth. For simplicity, we do not go into more detail in this direction but refer the interested reader to the literature, e.g., Bredies et al. (2010) and Burger and Osher (2013).

All three regularizers promote solutions with different smoothness properties. H^1 promotes smooth solutions with small gradients everywhere, whereas TV promotes solutions which have sparse gradients, i.e., the images are piecewise constant and appear cartoon-like. The latter also leads to the staircase artifact which can be overcome by TGV which promotes piecewise linear solutions. None of these regularizers are able to encode additional information on the location or direction of edges.

Contributions

The contributions in this chapter are threefold.

Overview over existing methods We provide an overview on existing mathematical models for structural similarity which are related to the shared location or direction of edges. We then discuss various regularizers which promote similarity in this sense.

Higher order models Existing methods focus on incorporating additional information into regularizers modeling first-order smoothness. We extend existing methodology to second-order smoothness using the total generalized variation framework.

Extensive numerical comparison We highlight the properties of the discussed regularizers and the dependence on various parameters using two inverse problems: tomography and super-resolution.

Related Work

Joint Reconstruction

One can think of the setting (1) with extra information v as a special case when multiple measurements

$$K_i u_i = f_i \quad i = 1, \dots, m \quad (6)$$

are taken. If $m = 2$ and one inverse problem is considerably less ill-posed, then this can be solved first to guide the inversion of the other. Some of the described models can be extended to the more general case (e.g., an arbitrary number of modalities) or the joint recovery of both/all unknowns (see, e.g., (Sapiro and Ringach 1996; Haber and Oldenburg 1997; Arridge and Simmons 1997; Gallardo and Meju 2003, 2004, 2011; Chen et al. 2013; Haber and Holtzman-Gazit 2013; Knoll et al. 2014;

Ehrhardt and Arridge 2014; Holt 2014; Ehrhardt et al. 2015; Rigie and La Riviere 2015; Knoll et al. 2016; Di et al. 2016; Mehranian et al. 2018; Zhang and Zhang 2018; Meju et al. 2019; Huber et al. 2019)), but it is out of the scope of this chapter to provide an overview on those. For an overview up to 2015, see Ehrhardt (2015). A few recent contributions are summarized in Arridge et al. (2020).

Model (6) may include several special cases: (i) multiple measurements of the same unknown, i.e., $u_i = u$, and (ii) measurements correspond to different states of the same unknown, e.g., in dynamic imaging $u_i = u(\cdot, t_i)$. The former case is covered by the standard literature when concatenating the measurements and the systems models, i.e., $(Ku)_i := K_i u$ and $f = (f_1, \dots, f_m)$. The latter has been widely studied in the literature, too; see, e.g., (Schmitt and Louis 2002; Schmitt et al. 2002; Schuster et al. 2018) and references therein. Both of these are in general unrelated to multi-modality imaging.

Other Models for Similarity

The earliest contributions to structure-promoting regularizers for multi-modality imaging were made in the early 1990s by Leahy and Yan (1991) who used a segmentation of an anatomical MRI image to enhance PET reconstruction. This is achieved by carefully handcrafting a regularizer which can encode this information. In this chapter we will use the same strategy but in a continuous setting which is independent of the discretization and will not rely on a segmentation of the side information v . These ideas were subsequently refined in various directions (Bowsher et al. 1996; Rangarajan et al. 2000; Comtat et al. 2002; Bowsher et al. 2004; Baete et al. 2004; Chan et al. 2007, 2009; Bousse et al. 2010; Pedemonte et al. 2011, Bilgic et al. 2011; Bousse et al. 2012; Bai et al. 2013) of which Bowsher's prior (Bowsher et al. 2004) remains most popular today.

Other models that can combine information of multiple modalities are based on coupled diffusion (Arridge and Simmons 1997; Tschumperlé and Deriche 2005; Arridge et al. 2008), level sets (Cheng-Liao and Qi 2011), information theoretic priors (joint entropy, mutual information) (Nuyts 2004; Tang and Rahmim 2009; Somayajula et al. 2011; Tang and Rahmim 2015), Bregman distances (Ballester et al. 2006; Möller et al. 2012; Estellers et al. 2013; Kazantsev et al. 2014; Rasch et al. 2018b), Bregman iterations (Möller et al. 2014; Rasch et al. 2018a), the structure tensor (Estellers et al. 2015), joint dictionary learning (Deligiannis et al. 2017; Song et al. 2018, 2019), common edge weighting (Zhang and Zhang 2018), and deep learning (Xiang et al. 2019). Most of these methods are very different to what will be described in this chapter. There are some similarities between the methods of this chapter and methods which are based on the Bregman distance of the total variation (Ballester et al. 2006; Möller et al. 2012, 2014; Estellers et al. 2013; Kazantsev et al. 2014; Rasch et al. 2018a,b), but a detailed treatment is outside the scope of this section.

Mathematical Models for Structural Similarity

In this section we define mathematical models where we aim to capture the similarities as shown in Figs. 1, 2, 3, 4, and 5. We start by explicitly stating two definitions which capture structural similarity which have been used implicitly in the literature. The first is based on the location of edges or the edge set (Arridge et al. 2008; Bresson and Chan 2008; Haber and Holtzman-Gazit 2013; Chen et al. 2013; Knoll et al. 2014; Möller et al. 2014; Ehrhardt et al. 2015; Zhang and Zhang 2018), and the second is based on direction of edges or the shape of an object (Gallardo and Meju 2003, 2004; Haber and Holtzman-Gazit 2013; Ehrhardt and Arridge 2014; Ehrhardt et al. 2015; Rigie and La Riviere 2015; Knoll et al. 2016). The latter is essentially the same as Definition 5.1.6 in Ehrhardt (2015) except for the degenerate case when either $\nabla u(x) = 0$ or $\nabla v(x) = 0$.

Definition 1 (Structural similarity with edge sets). Two differentiable images $u, v : \Omega \rightarrow \mathbb{R}$ are said to be *structurally similar in the sense of edge sets* if

$$\mathcal{E}u = \mathcal{E}v \tag{7}$$

where $\mathcal{E}u = \{x \in \Omega \mid \nabla u(x) \neq 0\}$. We also write $u \stackrel{\mathcal{E}}{\sim} v$ to denote that u and v are structurally similar in the sense of edge sets.

Definition 2 (Structural similarity with parallel level sets). Two differentiable images $u, v : \Omega \rightarrow \mathbb{R}$ are said to be *structurally similar in the sense of parallel level sets* if $u \stackrel{\mathcal{E}}{\sim} v$ and for all $x \in \mathcal{E}u$ the gradients $\nabla u(x)$ and $\nabla v(x)$ are co-linear which we denote by $\nabla u(x) \parallel \nabla v(x)$, i.e., there exists $\alpha(x) \in \mathbb{R}$ such that

$$\nabla u(x) = \alpha(x) \nabla v(x). \tag{8}$$

We also write $u \stackrel{d}{\sim} v$ to denote that u and v are structurally similar in the sense of parallel level sets.

Remark 1. For smooth images u and v , their gradients are perpendicular to their level sets, i.e., $u^{-1}(s) = \{x \in \Omega \mid u(x) = s\}$. Thus parallel gradients are equivalent to parallel level sets which explains the naming. The notion that the structure of an image is contained in its level sets dates back to Caselles et al. (2002).

Remark 2. By definition, similarity with parallel level sets (Definition 2) is stronger than the definition that only involves edge sets (Definition 1). An example of two images u and v which have the same edge set but do not have parallel level sets is the following. $u, v : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, $u(x) = x_1$, $v(x) = x_2$. Clearly they have the

same edge set since $\mathcal{E}u = \mathcal{E}v = \Omega$, but they do not have parallel level sets since $\nabla u(x) = [1, 0]$ but $\nabla v(x) = [0, 1]$.

Remark 3. Examples of images which have parallel level sets include:

1. *Function value transformations.* Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be smooth and strictly monotonic, i.e., $f' > 0$ or $f' < 0$. Then $v := f \circ u \stackrel{d}{\sim} u$. This is readily to be seen from the fact that $\nabla v(x) = f'(u(x))\nabla u(x) \neq 0$ if and only if $\nabla u(x) \neq 0$.
2. *Local function value transformations.* Let $f_i : \mathbb{R} \rightarrow \mathbb{R}$ be smooth and strictly monotonic and $u = \sum_i u_i$ where u_i are smooth functions whose gradients have mutually disjoint support. Then $v := \sum_i f_i \circ u_i \stackrel{d}{\sim} u$.

Remark 4. It has been argued in the literature that many multi-modality images $z : \Omega \rightarrow \mathbb{R}^m$ essentially decompose as

$$z_i(x) = \tau_i(x)\rho(x) \tag{9}$$

where $\rho(x)$ describes its structure and τ is a material property; see, e.g., Kimmel et al. (2000) and Holt (2014). Since the material does not change arbitrarily, it is natural to assume that τ_i is slowly varying or even piecewise constant. In the latter case, if x is such that $\nabla \tau_i(x) = 0$, then we have

$$\nabla z_i(x) = \tau_i(x)\nabla \rho(x), \tag{10}$$

in particular if $\tau_i, \tau_j \neq 0$, then $z_i \stackrel{d}{\sim} z_j$. This property is also related to the material decomposition in spectral CT; see, e.g., Fessler et al. (2002), Heismann et al. (2012) and Long and Fessler (2014).

Measuring Structural Similarity

Measuring the degree of similarity with respect to the previous two definitions of structural similarity is not easy, and we will now discuss a couple of ideas from the literature. Here and for the rest of this chapter, we will make frequent use of the vector-valued representation of a set of images $z : \Omega \rightarrow \mathbb{R}^2$, $z(x) := [u(x), v(x)]$. We denote by J its Jacobian, i.e., $J : \Omega \rightarrow \mathbb{R}^{d \times 2}$, $J_{i,j} = \partial_i z_j$.

With the definition of the Jacobian, we see that $u \stackrel{e}{\sim} v$ if and only if

$$\int_{\Omega} |J(x)|_0 \, dx = \int_{\Omega} |\nabla u(x)|_0 \, dx = \int_{\Omega} |\nabla v(x)|_0 \, dx \tag{11}$$

where $|x|_0 := 1$ if $x \neq 0$ and 0 else.

Similarly, by definition $u \stackrel{d}{\sim} v$ if and only if $u \stackrel{\mathcal{L}}{\sim} v$ and (a) $\text{rank } J(x) = 1$ for all $x \in \mathcal{E}u$. (a) is equivalent to (b) a vanishing determinant, i.e., $\det J^\top(x)J(x) = 0$. Simple calculations (see, e.g., Ehrhardt (2015)) show that

$$\det J^\top(x)J(x) = |\nabla u(x)|^2 |\nabla v(x)|^2 - \langle \nabla u(x), \nabla v(x) \rangle^2, \quad (12)$$

where we use the notation $\langle x, y \rangle = x^\top y$ for the inner product of two column vectors x and y . In order to get further equivalent statements, we turn to the singular values of the Jacobian which are given by

$$\sigma_1^2(x) = \frac{1}{2} \left[|J(x)|^2 + \sqrt{|J(x)|^4 - \det J^\top(x)J(x)} \right] \quad (13)$$

$$\sigma_2^2(x) = \frac{1}{2} \left[|J(x)|^2 - \sqrt{|J(x)|^4 - \det J^\top(x)J(x)} \right] \quad (14)$$

with $|J(x)|^2 = |\nabla u(x)|^2 + |\nabla v(x)|^2$; see, e.g., Ehrhardt (2015). Since $\sigma_1(x) \geq \sigma_2(x) \geq 0$ we have that (a) holds if and only if (c) the second singular value vanishes, i.e., $\sigma_2(x) = 0$ or (d) the vector of singular vectors $\sigma(x) = [\sigma_1(x), \sigma_2(x)]$ is 1-sparse.

Structure-Promoting Regularizers

Many of the abstract models from the previous section to measure the degree of similarity with respect to the previous two definitions of structural similarity are computationally challenging as they relate to non-convex constraints. In this section we will define convex structure-promoting regularizers which make them computationally tractable.

Isotropic Models

We first look at isotropic models which only depend on gradient magnitudes rather than directions, thus promoting structural similarity in the sense of edge sets, Definition 1.

First, based on (11) if we approximate $|J(x)|_0$ by $|J(x)|$, then

$$\text{JTV}(u) = \int_{\Omega} |J(x)| \, dx = \int_{\Omega} \sqrt{|\nabla u(x)|^2 + |\nabla v(x)|^2} \, dx \quad (15)$$

$$\leq \int_{\Omega} (|\nabla u(x)| + |\nabla v(x)|) \, dx = \text{TV}(u) + \text{TV}(v) \quad (16)$$

with equality if and only if $\mathcal{E}u \cap \mathcal{E}v = \emptyset$. This regularizer is called *joint total variation* in some communities (see, e.g., Chen et al. 2013; Haber and Holtzman-Gazit 2013; Ehrhardt et al. 2015, 2016) and *vectorial total variation* in others (see, e.g., Bresson and Chan 2008).

Remark 5. Note that JTV has the favorable property that if $\nabla v = 0$, then $\text{JTV}(u) = \text{TV}(u)$, so that it reduces to a well-defined regularization in u in this degenerate case. Note that this property also holds locally.

Remark 6. We would also like to note that there is a connection between JTV and the singular values of J . Let $\sigma_1, \sigma_2 : \Omega \rightarrow [0, \infty)$ be the two singular values of J , and then we have

$$\text{JTV}(u) = \int_{\Omega} \sqrt{\sigma_1^2(x) + \sigma_2^2(x)} \, dx . \tag{17}$$

Another strategy to favor edges at similar locations while reducing to a well-defined regularizer in the degenerate case is to introduce local weighting. Let $w : \Omega \rightarrow [0, 1]$ be an edge indicator function for v such that $w(x) = 1$ when $\nabla v(x) = 0$ and a small value whenever $|\nabla v(x)|$ is large. For example, choose

$$w(x) = \frac{\eta}{\sqrt{\eta^2 + |\nabla v(x)|^2}} \tag{18}$$

which is illustrated in Fig. 6. The figure shows that with a medium η the weight w in (18) shows the main structures of the images so that these can be promoted in the other image. If η is too small, then also unwanted structures are captured in w such as a smooth background variation. If η is too large, then the structures start to disappear.

For regularizers which are based on the image gradient ∇u , the weighting w can be used to favor edges at certain locations by replacing ∇ by $w\nabla$. For instance, for H^1 (3), TV (4), and TGV (5), this strategy results in

$$wH^1(u) = \int_{\Omega} |w(x)\nabla u(x)|^2 \, dx = \int_{\Omega} w^2(x)|\nabla u(x)|^2 \, dx \tag{19}$$

$$w\text{TV}(u) = \int_{\Omega} |w(x)\nabla u(x)| \, dx = \int_{\Omega} w(x)|\nabla u(x)| \, dx \tag{20}$$

$$w\text{TGV}(u) = \inf_{\zeta} \int_{\Omega} |w(x)\nabla u(x) - \zeta(x)| + \beta|E\zeta(x)| \, dx \tag{21}$$

which we will refer to as *weighted squared H^1 -semi norm*, *weighted total variation*, and *weighted total generalized variation*. $w\text{TV}$ was used in Arridge et al. (2008), Lenzen and Berger (2015) and Ehrhardt and Betcke (2016). A variant of $w\text{TV}$ has been considered for single modality imaging in Hintermüller and Rincon-Camacho

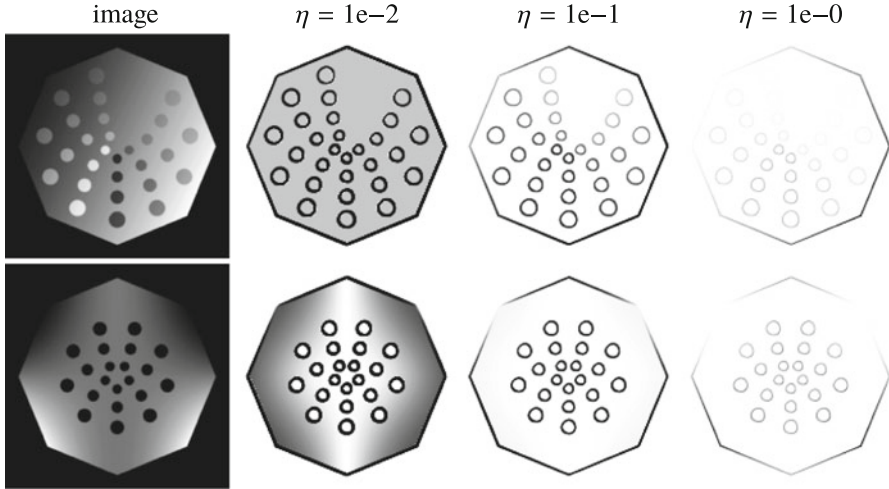


Fig. 6 Influence of the parameter η on estimation of **edge location**. The images on the right show the scalar field $w : \Omega \rightarrow [0, 1]$ which locally weights the influence of the regularizer; see (18). Here “black” denotes 0 and “white” denotes 1

(2010) and Dong et al. (2011) and extended to a variant of wTGV (Bredies et al. 2012).

Remark 7. The parameter η in w (see (18)) should be chosen in relation to $|\nabla v(x)|$. A common strategy is to normalize the side information first such that $\sup_{x \in \Omega} |\nabla v(x)| = 1$. Then desirable values of η are usually within the range $[0.01, 1]$.

Anisotropic Models

The same idea which resulted in isotropically “weighted” variants of common regularizers can be used anisotropically, i.e., by making the local weights vary with direction. Let us denote the anisotropic weighting by $D : \Omega \rightarrow \mathbb{R}^{d \times d}$. Similar to the isotropic variant, one would like the weight to become the identity matrix, i.e., $D(x) = I$, when $\nabla v(x) = 0$. In order to promote parallel level sets, it is desirable that $D(x)\nabla u(x)$ should be small if $\nabla u(x) \parallel \nabla v(x)$ and $D(x)\nabla u(x) = \nabla u(x)$ if $\nabla u(x) \perp \nabla v(x)$, i.e., $\langle \nabla u(x), \nabla v(x) \rangle = 0$. For example,

$$D(x) = I - \gamma \xi(x) \xi^\top(x), \quad \xi(x) = \frac{\nabla v(x)}{\sqrt{\eta^2 + |\nabla v(x)|^2}} \quad (22)$$

for $\gamma \in (0, 1]$ (usually close to 1) and $\eta > 0$ satisfies all of these properties. Clearly if $\nabla v(x) = 0$, then $\xi = 0$ such that $D(x) = I$. Moreover, if $\nabla u(x) \parallel \nabla v(x)$, then there exists an α such that $\nabla u(x) = \alpha \nabla v(x)$ and

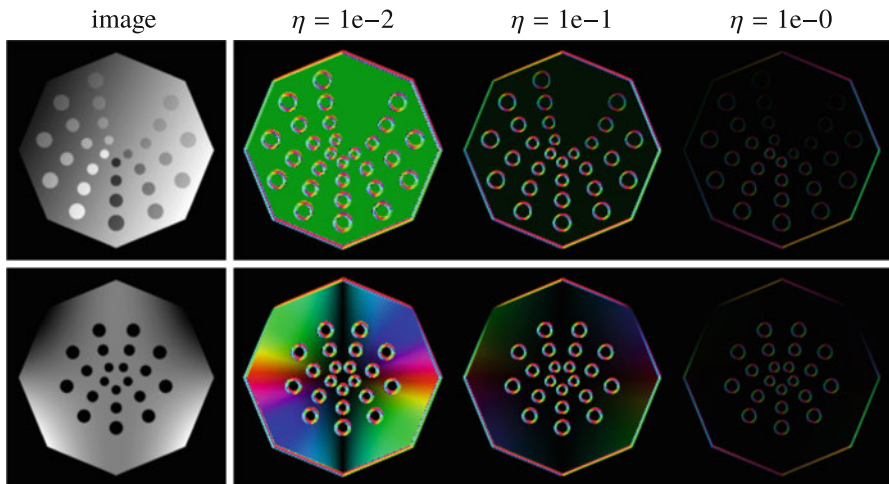


Fig. 7 Influence of the parameter η on estimation of **edge location and direction**. The images on the right show the vector field $\xi : \Omega \rightarrow \mathbb{R}^d$ which locally defines the influence of the regularizer; see, e.g., (22). Here “black” denotes that the magnitude of ξ , i.e., $|\xi(x)|$, is 0, and a bright color denotes that $|\xi(x)|$ is 1. The colors show the direction of the vector field ξ modulo its sign

$$D(x)\nabla u(x) = \left[I - \frac{\gamma}{\eta^2 + |\nabla v(x)|^2} \nabla v(x)\nabla v^\top(x) \right] \nabla u(x) \tag{23}$$

$$= \left[1 - \frac{\gamma|\nabla v(x)|^2}{\eta^2 + |\nabla v(x)|^2} \right] \nabla u(x). \tag{24}$$

The scalar weighting factor converges to $1 - \gamma$ for $|\nabla v(x)| \rightarrow \infty$. Finally, if $\nabla u(x) \perp \nabla v(x) = 0$, then clearly $D(x)\nabla u(x) = \nabla u(x)$.

The example of the matrix-field $D : \Omega \rightarrow \mathbb{R}^{d \times d}$ in (22) is determined by the vector-field $\xi : \Omega \rightarrow \mathbb{R}^d$ which we visualize in Fig. 7. The colors show the direction of the vector-field modulo its sign (since $\xi(x)\xi^\top(x)$ is invariant to a change of sign), and the brightness indicates its magnitude $|\xi(x)|$. Note that images appear as color versions of Fig. 6 which shows the isotropic weighting w .

Using a matrix-field in common regularizers leads to their “directional” variant

$$dH^1(u) = \int_{\Omega} |D(x)\nabla u(x)|^2 dx \tag{25}$$

$$dTV(u) = \int_{\Omega} |D(x)\nabla u(x)| dx \tag{26}$$

$$dTGV(u) = \inf_{\zeta} \int_{\Omega} |D(x)\nabla u(x) - \zeta(x)| + \beta|E\zeta(x)| dx. \tag{27}$$

Remark 8. There is a connection between the particular choice of the matrix-field D in (22) and the Jacobian J .

$$\begin{aligned} |D(x)\nabla u(x)|^2 &= |\nabla u(x) - \frac{\gamma}{\eta^2 + |\nabla v(x)|^2} \langle \nabla u(x), \nabla v(x) \rangle \nabla v(x)|^2 \\ &= |\nabla u(x)|^2 - \frac{2\gamma\eta^2 + \gamma(2-\gamma)|\nabla v(x)|^2}{(\eta^2 + |\nabla v(x)|^2)^2} \langle \nabla u(x), \nabla v(x) \rangle^2. \end{aligned} \quad (28)$$

$$(29)$$

For $\eta = 0$, $\gamma = 1$, and $|\nabla v(x)| = 1$, then with (12) we have

$$|D(x)\nabla u(x)|^2 = |\nabla u(x)|^2 |\nabla v(x)|^2 - \langle \nabla u(x), \nabla v(x) \rangle^2 = \det J^\top(x) J(x). \quad (30)$$

Thus, dH^1 corresponds to penalizing the determinant. This regularizer is widely used for joint reconstruction in geophysics under the name *cross-gradient* function since it is also the cross product of $\nabla u(x)$ and $\nabla v(x)$; see, e.g., (Gallardo and Meju 2003, 2004, 2011; Meju et al. 2019). Similarly the dTV used, for instance, in medical imaging (Ehrhardt and Betcke 2016; Ehrhardt et al. 2016; Bathke et al. 2017; Schramm et al. 2017; Kolehmainen et al. 2019; Obert et al. 2020) and remote sensing (Bungert et al. 2018) can be seen as penalizing the square root of the determinant.

Another strategy to promote parallel level sets is via nuclear norm of the Jacobian which is defined as $|J(x)|_* = \sum_{i=1}^{\min(d,2)} \sigma_i(x)$ where $\sigma_i(x)$ denotes the i th singular value of $J(x)$. Using the nuclear norm promotes sparse vectors of singular values $\sigma(x) = [\sigma_1(x), \sigma_2(x)]$ and thereby parallel level sets. As a regularizer

$$\text{TNV}(u) = \int_{\Omega} |J(x)|_* \, dx \quad (31)$$

this strategy became known as *total nuclear variation*; see Holt (2014), Rigie and La Riviere (2015), Knoll et al. (2016), and Rigie et al. (2017).

All first-order regularizers of this section can be readily summarized in the following standard form

$$\mathcal{J}(u) = \int_{\Omega} \phi[B(x)\nabla u(x)] \, dx \quad (32)$$

where $B(x) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is an affine transformation and $\phi : \mathbb{R}^m \rightarrow \mathbb{R}$. For details how B and ϕ can be chosen for specific regularizers to fit this framework, see Table 1. It is useful for Jacobian-based regularizers to use the reweighted Jacobian $[\nabla u(x), \xi(x)]$ with $\xi(x) = \eta \nabla v(x)$ instead.

Table 1 Examples of first-order structure-promoting regularizers; see (32)

Regularizer	Definition	$B(x)y$	m	$\phi(x)$
H^1	(3)	y	d	$ x ^2$
wH^1	(19)	$w(x)y$	d	$ x ^2$
dH^1	(25)	$D(x)y$	d	$ x ^2$
TV	(4)	y	d	$ x $
wTV	(20)	$w(x)y$	d	$ x $
dTV	(26)	$D(x)y$	d	$ x $
JTV	(16)	$[y, \xi(x)]$	$d \times 2$	$ x $
TNV	(31)	$[y, \xi(x)]$	$d \times 2$	$ x _*$

Algorithmic Solution

Note that the solution to variational regularization (2) with either first- (32) or second-order structural regularization (5), (21), (27) can be cast into the general non-smooth composite optimization form

$$\min_x \mathcal{F}(Ax) + \mathcal{G}(x) \tag{33}$$

with $\mathcal{F}(y) = \sum_{i=1}^n \mathcal{F}_i(y_i)$ and $Ax = [A_1x, \dots, A_nx]$; see Table 2. We denote by $\|\cdot\|_{2,1}$, $\|\cdot\|_2^2$ and $\|\cdot\|_{*,1}$ discretizations of

$$z \mapsto \int_{\Omega} |z(x)| \, dx, \quad z \mapsto \int_{\Omega} |z(x)|^2 \, dx \quad \text{and} \quad z \mapsto \int_{\Omega} |z(x)|_* \, dx. \tag{34}$$

Note that all functionals \mathcal{F}_i and \mathcal{G} in Table 2 are proper, convex, and lower-semi continuous.

Algorithm

A popular algorithm to solve (33) and therefore (2) is the primal-dual hybrid gradient (PDHG) (Esser et al. 2010; Chambolle and Pock 2011); see Algorithm 1. It consists of two simple steps only involving basic linear algebra and the evaluation of the operator A and its adjoint A^* . Moreover, it involves the computation of the proximal operator of $\tau\mathcal{G}$ and the convex conjugate of $\sigma\mathcal{F}^*$ where τ and σ are scalar step sizes. The proximal operator of a functional \mathcal{H} is defined as

$$\text{prox}_{\mathcal{H}}(z) := \arg \min_x \left\{ \frac{1}{2} \|x - z\|_2^2 + \mathcal{H}(x) \right\}. \tag{35}$$

The proximal operator can be computed in closed-form for $\|\cdot\|_{2,1}$ and $\|\cdot\|_2^2$. It also can be computed in closed-form for $\|\cdot\|_{*,1}$ if either the number channels

Table 2 Mapping the variational regularization models into the composite optimization framework (33). In all cases we choose $A_1x = Ku$, $\mathcal{F}_1(y_1) = \mathcal{D}(y_1, b)$, and $\mathcal{G}(x) = \iota_{\geq 0}(u)$

Regularizer	Definition	x	A_2x	A_3x	$\mathcal{F}_2(y_2)$	$\mathcal{F}_3(y_3)$
H^1	(3)	u	∇u	–	$\alpha \ y_2\ _2^2$	–
w H^1	(19)	u	$w\nabla u$	–	$\alpha \ y_2\ _2^2$	–
d H^1	(19)	u	$D\nabla u$	–	$\alpha \ y_2\ _2^2$	–
TV	(4)	u	∇u	–	$\alpha \ y_2\ _{2,1}$	–
wTV	(20)	u	$w\nabla u$	–	$\alpha \ y_2\ _{2,1}$	–
dTV	(26)	u	$D\nabla u$	–	$\alpha \ y_2\ _{2,1}$	–
JTV	(16)	u	$[\nabla u, 0]$	–	$\alpha \ y_2 - [0, \xi]\ _{2,1}$	–
TNV	(31)	u	$[\nabla u, 0]$	–	$\alpha \ y_2 - [0, \xi]\ _{*,1}$	–
TGV	(5)	(u, ζ)	$\nabla u - \zeta$	$E\zeta$	$\alpha \ y_2\ _{2,1}$	$\alpha\beta \ y_3\ _{2,1}$
wTGV	(21)	(u, ζ)	$w\nabla u - \zeta$	$E\zeta$	$\alpha \ y_2\ _{2,1}$	$\alpha\beta \ y_3\ _{2,1}$
dTGV	(27)	(u, ζ)	$D\nabla u - \zeta$	$E\zeta$	$\alpha \ y_2\ _{2,1}$	$\alpha\beta \ y_3\ _{2,1}$

Algorithm 1 Primal-dual hybrid gradient (PDHG) to solve (33). Default values given in brackets

Input: iterates $x(=0)$, $y(=0)$, step size parameter $\rho(=1)$

Initialize: extrapolation $\bar{x} = x$, step sizes $\sigma = \rho/\|A\|$, $\tau = 0.999/(\rho\|A\|)$

1: **for** $k = 1, \dots$ **do**

2: $x^+ = \text{prox}_{\tau\mathcal{G}}(x - \tau A^*y)$

3: $y^+ = \text{prox}_{\sigma\mathcal{F}^*}(y + \sigma A(2x^+ - x))$

4: **end for**

or the dimension of the domain are strictly less than 5, i.e., $m, d < 5$; see Holt (2014) for more details. Note also that the proximal operator of $\alpha\mathcal{F}(\cdot - \xi)$ can be readily computed based on the proximal operator of \mathcal{F} . More details on proximal operators, convex conjugates, and examples can be found, for example, in Bauschke and Combettes (2011), Combettes and Pesquet (2011), Parikh and Boyd (2014), and Chambolle and Pock (2016).

For some applications (e.g., x-ray tomography), a preconditioned (Pock and Chambolle 2011; Ehrhardt et al. 2019) or randomized (Chambolle et al. 2018; Ehrhardt et al. 2019) variant can be useful, but we will not consider these here for simplicity.

Prewhitening

Since the operator norms $\|A_i\|, i = 1, \dots, n$ can vary significantly, it is often advisable to “prewhiten” the problem by recasting it as

$$\min_x \tilde{\mathcal{F}}(\tilde{A}x) + \mathcal{G}(x). \quad (36)$$

with $\tilde{\mathcal{F}}(y) := \sum_{i=1}^n \mathcal{F}_i(\|A_i\| \cdot y_i)$ and $\tilde{A}_i x := A_i x / \|A_i\|$. Then trivially $\|\tilde{A}_i\| = 1$, $i = 1, \dots, n$ so that all operator norms are equal. Note that the proximal operator of $\sigma \tilde{\mathcal{F}}$ is simple to compute if the proximal operators of $\sigma \mathcal{F}_i$, $i = 1, \dots, n$ are simple to compute, since

$$[\text{prox}_{\sigma \tilde{\mathcal{F}}}(y)]_i = \lambda_i^{-1} [\text{prox}_{\sigma \lambda_i^2 \mathcal{F}_i}(\lambda_i y_i)], \quad (37)$$

for any $\lambda_i > 0$; see, for instance, Bredies and Lorenz (2018, Lemma 6.136).

Numerical Comparison

This section describes numerical experiments to compare first- and second-order structure-promoting regularizers.

Software, Data, and Parameters

Software The numerical computations are carried out in Python using ODL (version 1.0.0.dev0) (Adler et al. 2017) and ASTRA (van Aarle et al. 2015, 2016) for computing line integrals in the tomography example. The source code which reproduces all experiments in this chapter can be found at <https://github.com/mehrhardt/Multi-Modality-Imaging-with-Structural-Priors>.

Data We consider two test cases with different characteristics, both of which are visualized in Fig. 8. The first test case, later referred to as `x-ray`, is parallel beam x-ray reconstruction from only 15 views where additionally some detectors are broken. The latter is modeled by salt-and-pepper noise where 5% of all detectors are corrupted. We aim to recover an image with domain $[-1, 1]^2$ discretized with 200^2 pixels. The simulated x-ray camera has 100 detectors and a width of 3 in the same dimensions as the image domain. Therefore, the challenges are (1) sparse views, (2) small number of detectors, and (3) broken detectors.

The second test case, which we refer to as `super-resolution`, considers the task of super-resolution. Also here we aim to recover an image with domain $[-1, 1]^2$ discretized with 200^2 pixels. The forward operator is integrating over 5^2 pixels, thus mapping images of size 200^2 to images of size 40^2 . In addition, Gaussian noise of mean zero and standard deviation of 0.01 is added.

Algorithmic parameters We chose the default value $\rho = 1$ for balancing the step sizes in PDHG and ran the algorithm for 3,000 iterations without choosing a specific stopping criterion.

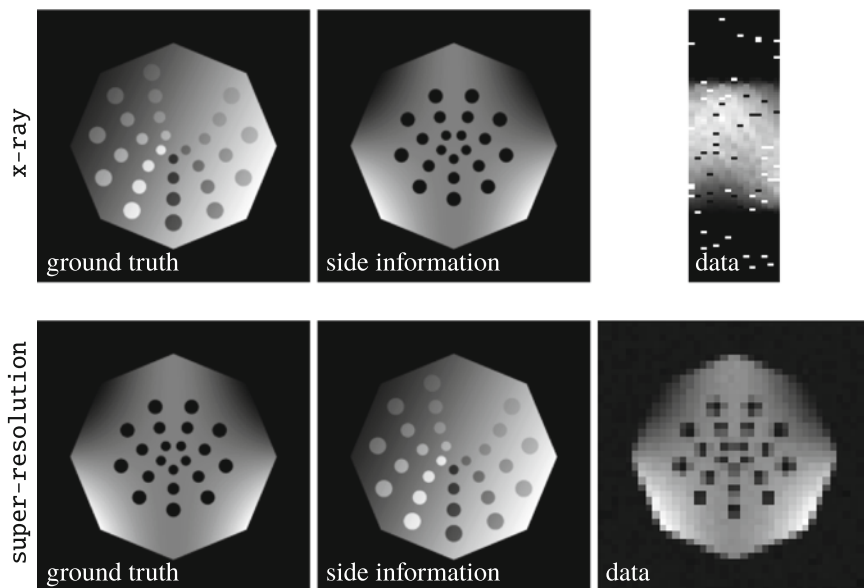


Fig. 8 Test cases for numerical experiments. Top: x-ray reconstruction from sparse views and failed detectors. Bottom: super-resolution by a factor of 5 and Gaussian noise

Numerical Results

The multiplicative scaling of an unconstrained optimization problem is arbitrary; nevertheless we report the absolute values here for completeness. For simplicity, all regularization parameters are shown as multiples of $1e-4$. The figures at the bottom right of each image are PSNR and SSIM.

Test Case x-ray

Effect of edge weighting All structure-promoting regularizers described in section “[Structure-Promoting Regularizers](#)” have in common that they rely to some extent on the size of edges in the side information, i.e., $|\nabla v(x)|$. For JTV and TNV the actual values of $|\nabla v(x)|$ matter so that a parameter η is needed to correct for this. For all other regularizers a parameter η is needed to decide which edges to trust and which not. The effect of this edge weighting parameter η on all described regularizers is illustrated in Figs. 9, 10, and 11. The locally weighted regularizers (i.e., wH^1 , wTV , and $wTGV$) and the directional regularizers (i.e., dH^1 , dTV , and $dTGV$) have in common that if η is too small, then small artifacts around the edges appear. This effect is more pronounced in locally weighted regularizers. If η is too large, then the structure-promoting effect becomes too small. For joint total variation and total nuclear variation, similar effects exist with reverse relationship to η .

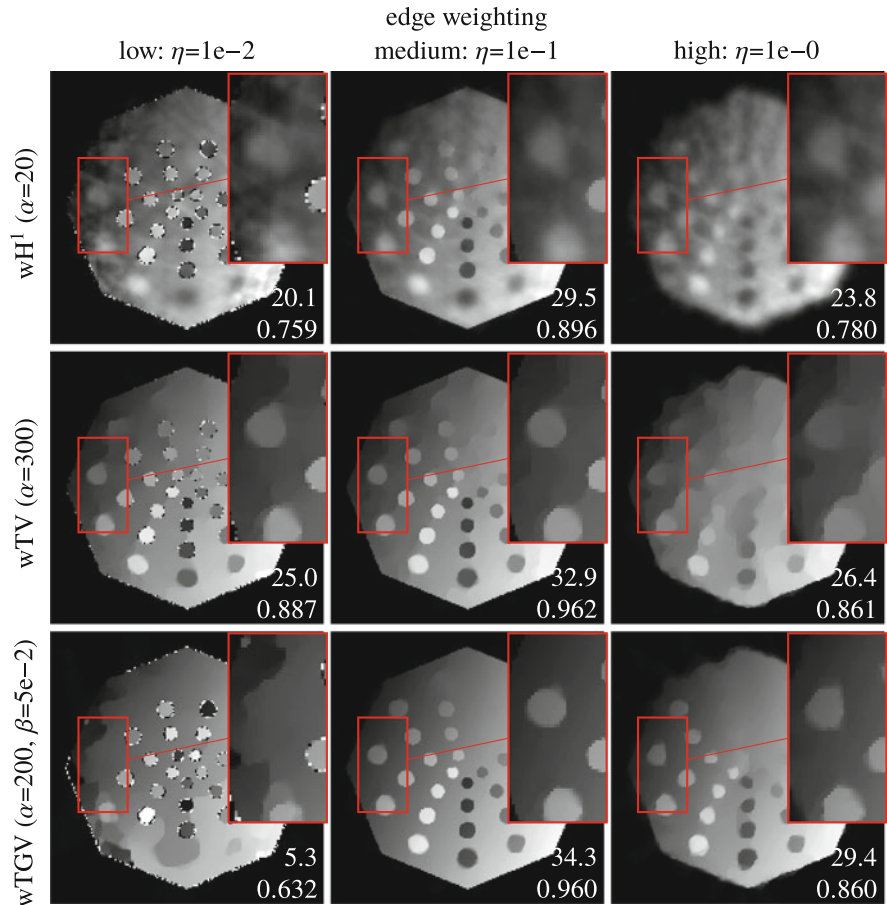


Fig. 9 Effect of edge weighting on locally weighted models for test case x -ray: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality

Effect of regularization The effect of the regularization parameter α on the solution is illustrated in Figs. 12, 13, and 14. All regularizers show the same behavior if α is too small or too large. If the regularization parameter is chosen too small, then artifacts from inverting an ill-posed operator are introduced, and if it is chosen too large, then all regularizers oversmooth the solution. Note that all structure-promoting regularizers have an increased robustness in areas of shared structures.

Comparison of regularizers All eleven regularizers are compared in Fig. 15. It can be seen that the structure-promoting regularizers perform much better in terms of PSNR and SSIM as their non-structure-promoting counterparts. Moreover, one can

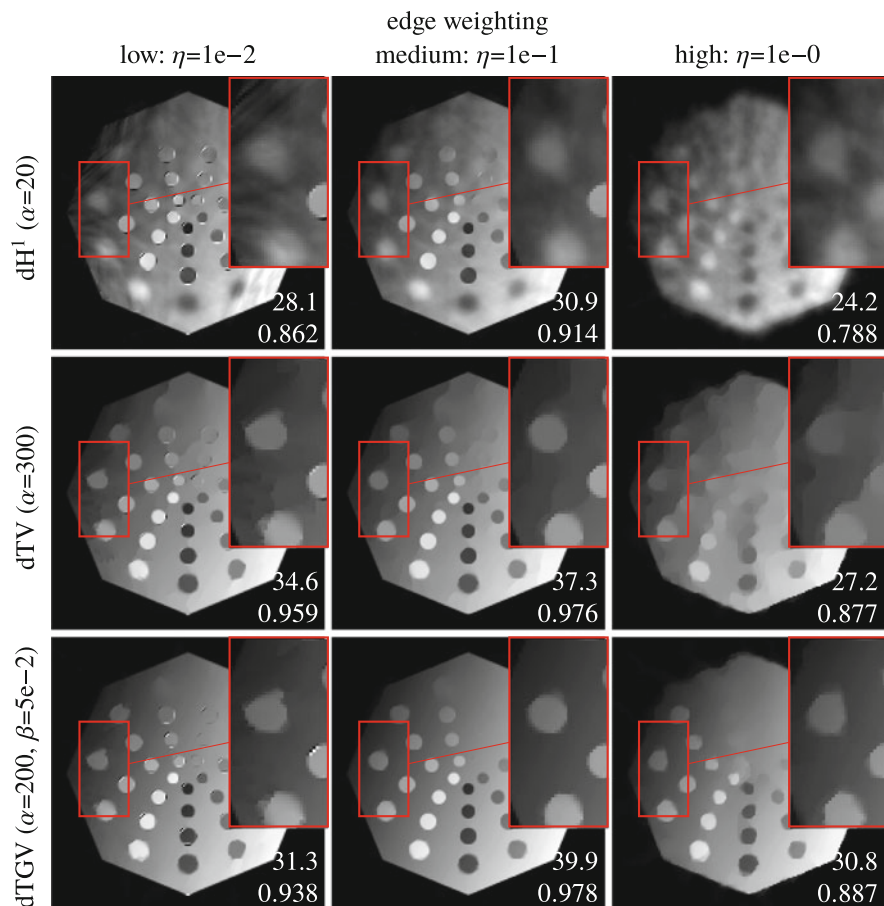


Fig. 10 Effect of edge weighting on directional models for test case x-ray: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality ($\gamma = 1$)

observe an interesting effect that the structure-promoting regularizers also perform visually better in regions where the structure is not shared, e.g., the outer ring of circles. This effect is most dominant for dTGV where the circle at the top left is clearly visible, while it is difficult to spot for many of the other regularizers.

Test Case Super-Resolution

Effect of edge weighting Figs. 16, 17, and 18 show the effect of the edge weighting parameter η . One can make similar observations as in Figs. 9, 10, and 11 for the test case x-ray. In addition, one can observe from the close-ups that if η is too small (or too large for JTV and TNV), then ghosting artifacts may appear. Note that these are present for TNV even for a moderate choice of η .

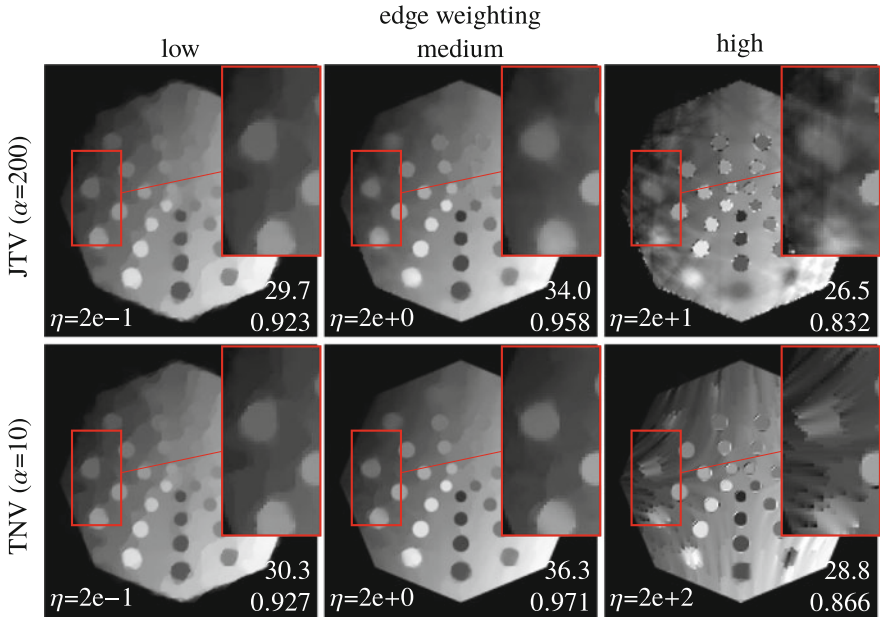


Fig. 11 Effect of edge weighting on joint total variation and total nuclear variation for test case x-ray: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality

Comparison of regularizers All regularizers are compared in Fig. 19 for the test case super-resolution. It can be noted from all images that introducing structural information allows to resolve some of the inner circles which have been merged for regularizers which are not structure-promoting. Moreover, all total generalized variation-based regularizers do not perform much better than the total variation-based regularizers. The directional regularizers as well as JTV and TNV perform best in terms of PSNR for this example.

Discussion on Computational Cost

The median computing times for the numerical experiments are reported in Table 3. The computing time of PDHG is mainly influenced by the dimensions of the models, the proximal operator, and the forward model. As can be seen from the table, H^1 and TV are roughly the same fast. TGV which uses a second primal variable in the space of the image gradient is significantly slower with about twice the computational cost. In all three cases, introducing isotropic weights (i.e., wH^1 , wTV , and $wTGV$) increases the cost by about 6 seconds, and anisotropic weights (i.e., dH^1 , dTV , and

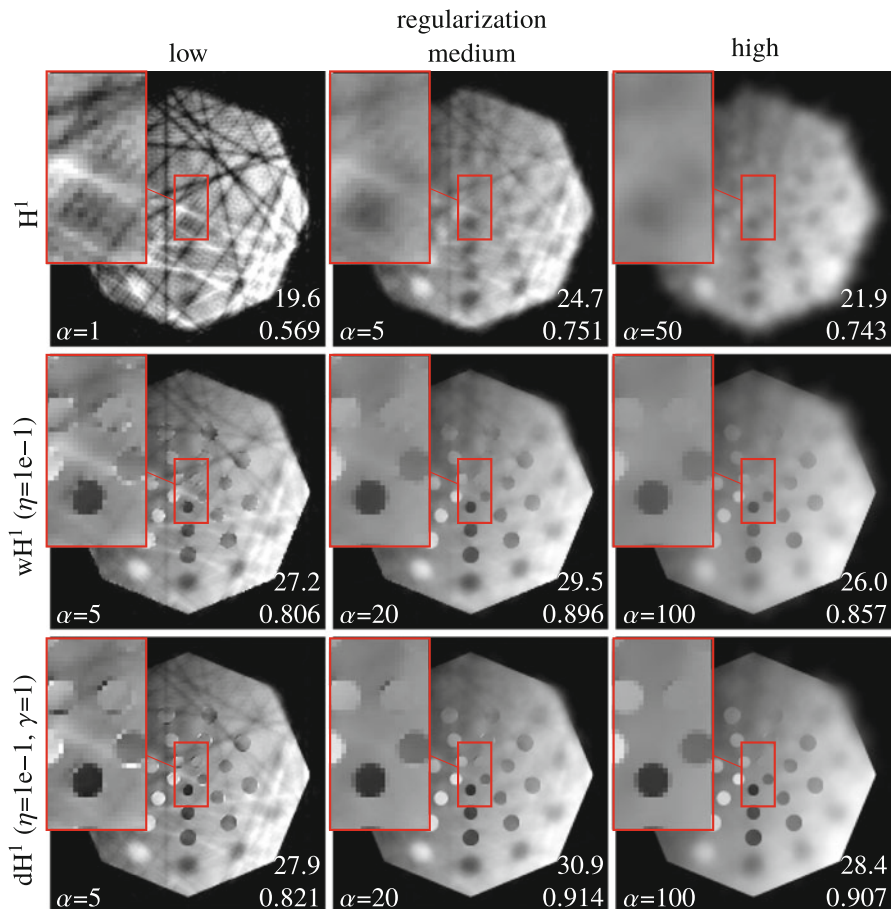


Fig. 12 H^1 -semi norm-based structure-promoting regularizers for test case x-ray: increasing the regularization parameter α from left to right. All other parameters were tuned to maximize the PSNR and visual image quality. All regularizers in this figure reduce to the H^1 -semi norm in areas where the side information is flat

dTGV) by about 12 s. JTV is more costly than dTV but not as costly as TGV. TNV is by far the most costly of all algorithms due to the need to compute singular value decompositions of 2×2 -matrices for every pixel.

Since we run PDHG always for 3,000 iterations, we do not report computational time “till convergence” but computational cost for the full 3,000 iterations. It was observed at several occasions (see, e.g., Ehrhardt et al. 2019) that including side information into the regularizer not only improves the reconstruction but also

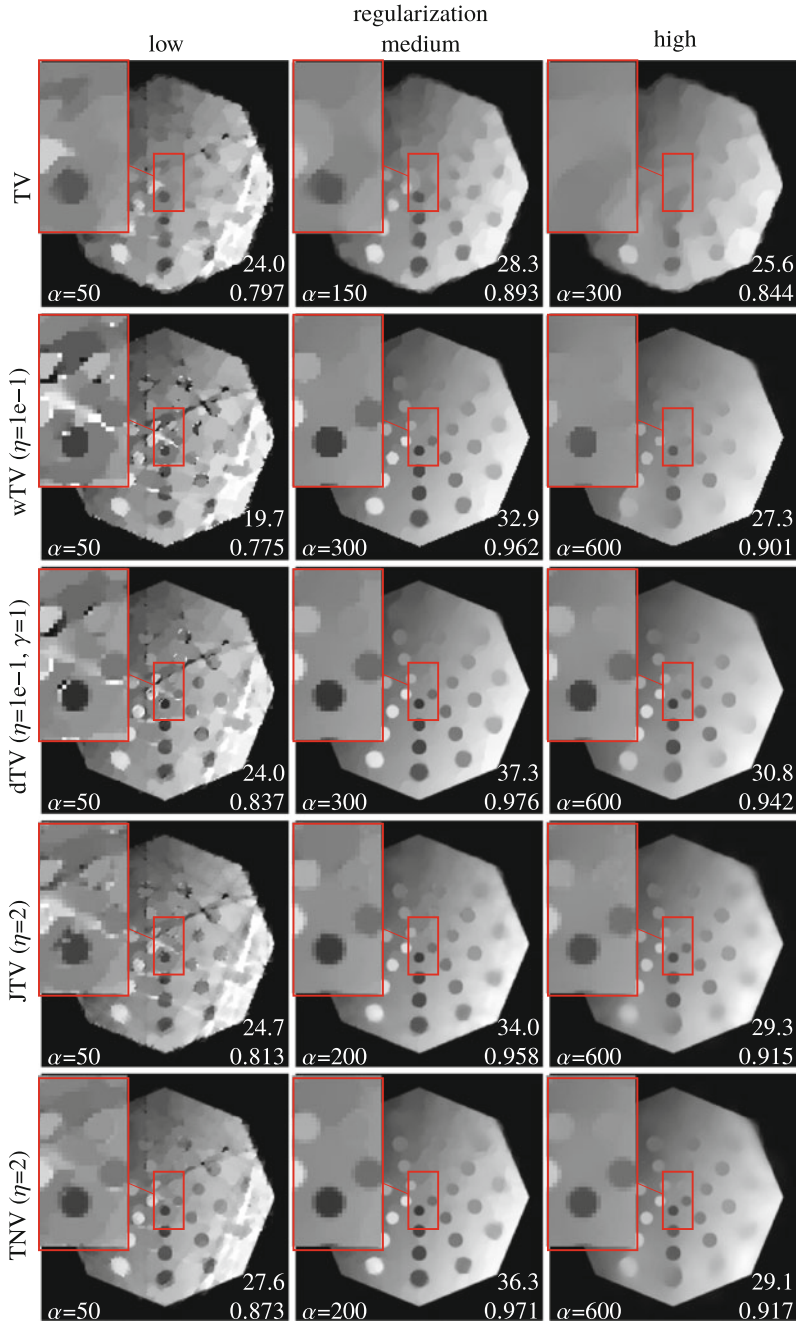


Fig. 13 Total variation based structure-promoting regularizers for test case x-ray: increasing the regularization parameter α from left to right. All other parameters were tuned to maximize the PSNR and visual image quality. All regularizers in this figure reduce to the total variation in areas where the side information is flat

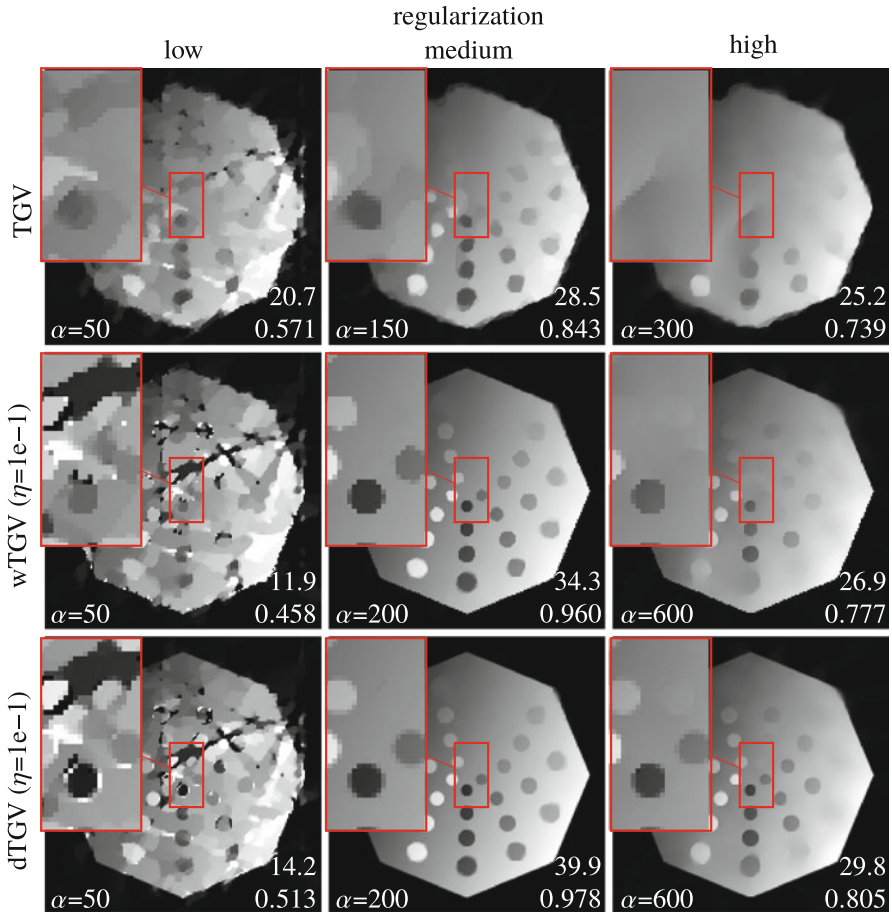


Fig. 14 Total generalized variation-based structure-promoting regularizers for test case x-ray: increasing the regularization parameter α from left to right. All other parameters were tuned to maximize the PSNR and visual image quality ($\beta = 5e-2$). All regularizers in this figure reduce to the total generalized variation in areas where the side information is flat

speeds up the algorithmic convergence. Intuitively this can be understood as more information is included into the optimization problem.

Comparing the regularizers regarding their computational time versus image quality trade-off, it can be noted that TNV should not be chosen since it is not better than dTV at 7-10x the computational cost. Whether H^1 , TV, or TGV based regularizer is desirable depends on each individual application. For each of them, there is a clear trend that one achieves better image quality by introducing more information, i.e., first isotropic information and then anisotropic information, each of which increases their computational cost. However, the increase in computational cost is so small that for most applications the directional variant is likely to be favored.

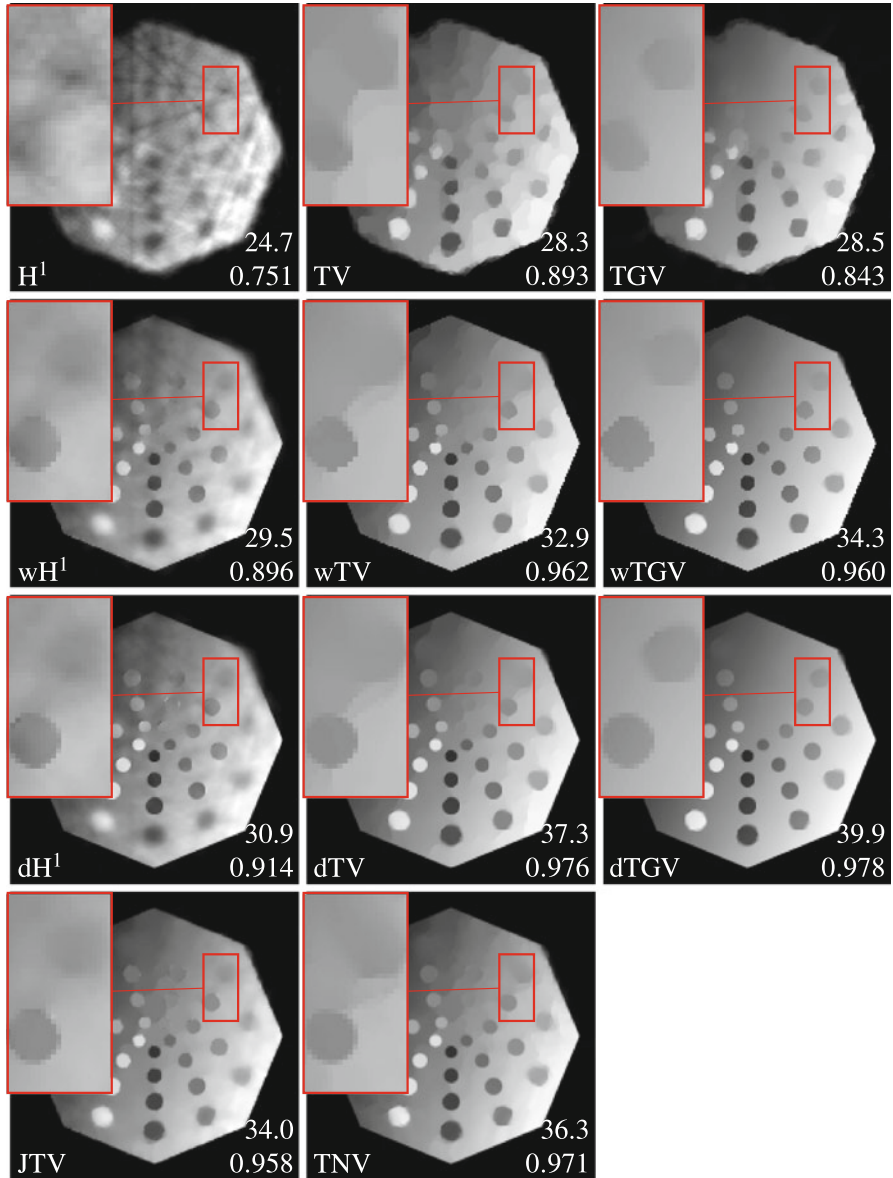


Fig. 15 Comparison of structure-promoting regularizers for test case `x-ray`. All parameters were tuned to maximize the PSNR and visual image quality

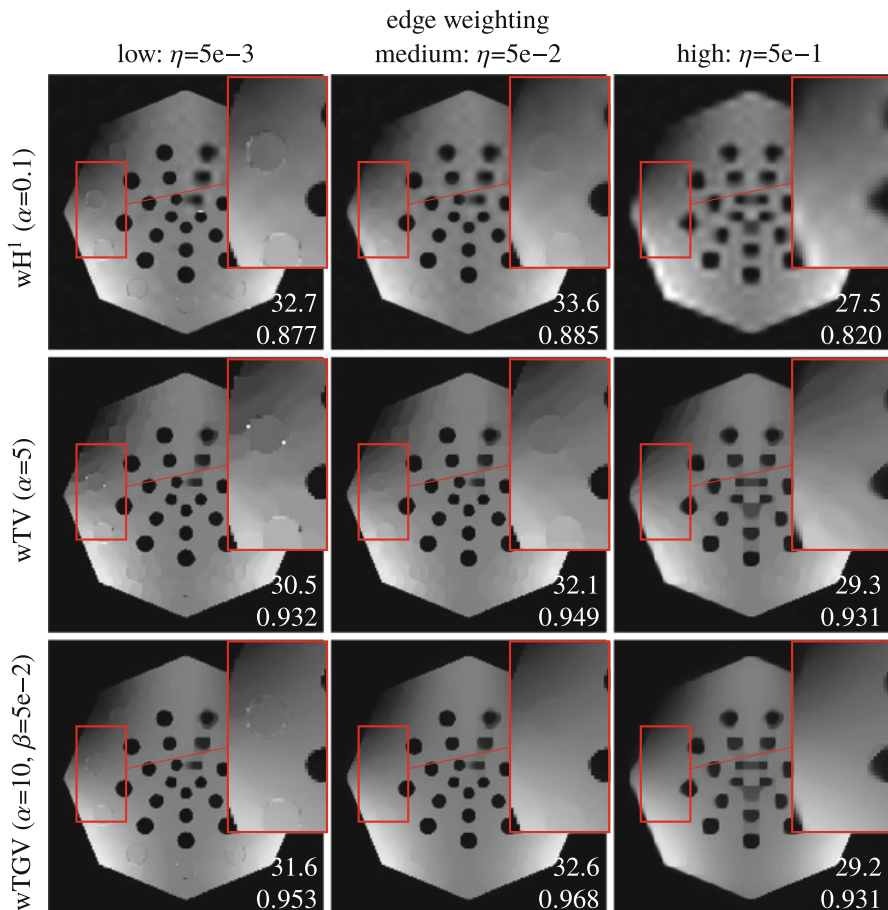


Fig. 16 Effect of edge weighting on locally weighted models for test case super-resolution: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality

Conclusions

This chapter introduced fundamental mathematical concepts on the structure of images and how structural similarity between images can be measured. The fundamental building blocks are the similarity based on edge sets and parallel level sets. These notions lead to several classes of structure-promoting regularizers all of which are convex and thereby lead to tractable optimization problems when used in variational regularization for linear inverse problems with convex data fits. While some of the regularizers are smooth and others are non-smooth, the resulting optimization problem for all of them can be efficiently computed by PDHG. The effectiveness

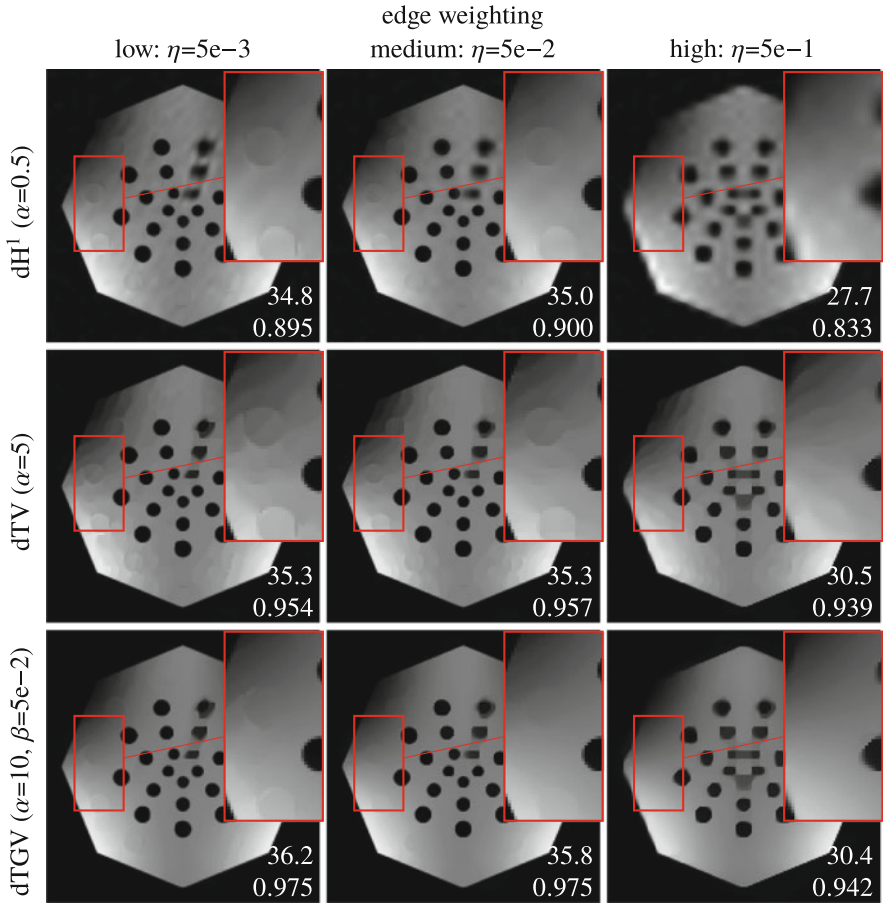


Fig. 17 Effect of edge weighting on directional models for test case super-resolution: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality ($\gamma = 0.9$)

of these regularizers for the promotion of structure has been observed in many applications and was also illustrated in this chapter on two simulation studies.

Open Problems

The mathematical framework for structure-promoting regularizers is by now well established and fairly mature. Open problems reside in practical problems in the translation of these techniques to applications which will also motivate further mathematical research.

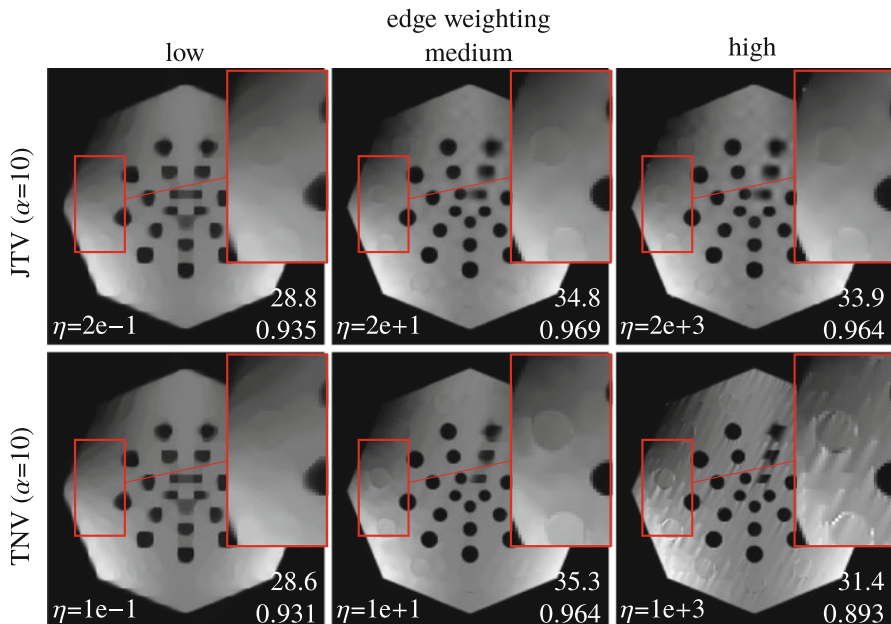


Fig. 18 Effect of edge weighting on joint total variation and total nuclear variation for test case super-resolution: increasing edge parameter η from left to right. All other parameters were tuned to maximize the PSNR and visual image quality

Misregistration The biggest open problem is misregistration. All of the described regularizers assume that both images are perfectly aligned. Even in scanners which have two imaging modalities in the same system such as PET-MR, this assumption is never perfectly fulfilled. This issue has not been addressed much in the literature. In Tsai et al. (2018), the authors proposed an alternating approach between image reconstruction and image registration with some success. In Bungert et al. (2018), the problem was formulated as a blind deconvolution problem so that translations can be compensated with a shifted kernel. A heuristic modification made this approach more robust to large translations (Bungert et al. 2018). A joint reconstruction and affine registration approach was proposed in Bungert and Ehrhardt (2020) which solves the misregistration problem in some cases, e.g., in neurology.

Extensions beyond two modalities It is natural to consider the case that more than one image is available as side information. For instance, in some remote sensing applications, a color photograph with high spatial resolution is available. Similarly, in PET-MR, images of more than one MR sequence might be available. This setting has also been considered in Mehranian et al. (2017) for a purely discrete model. Some of the regularizers to promote structural similarity in this chapter naturally

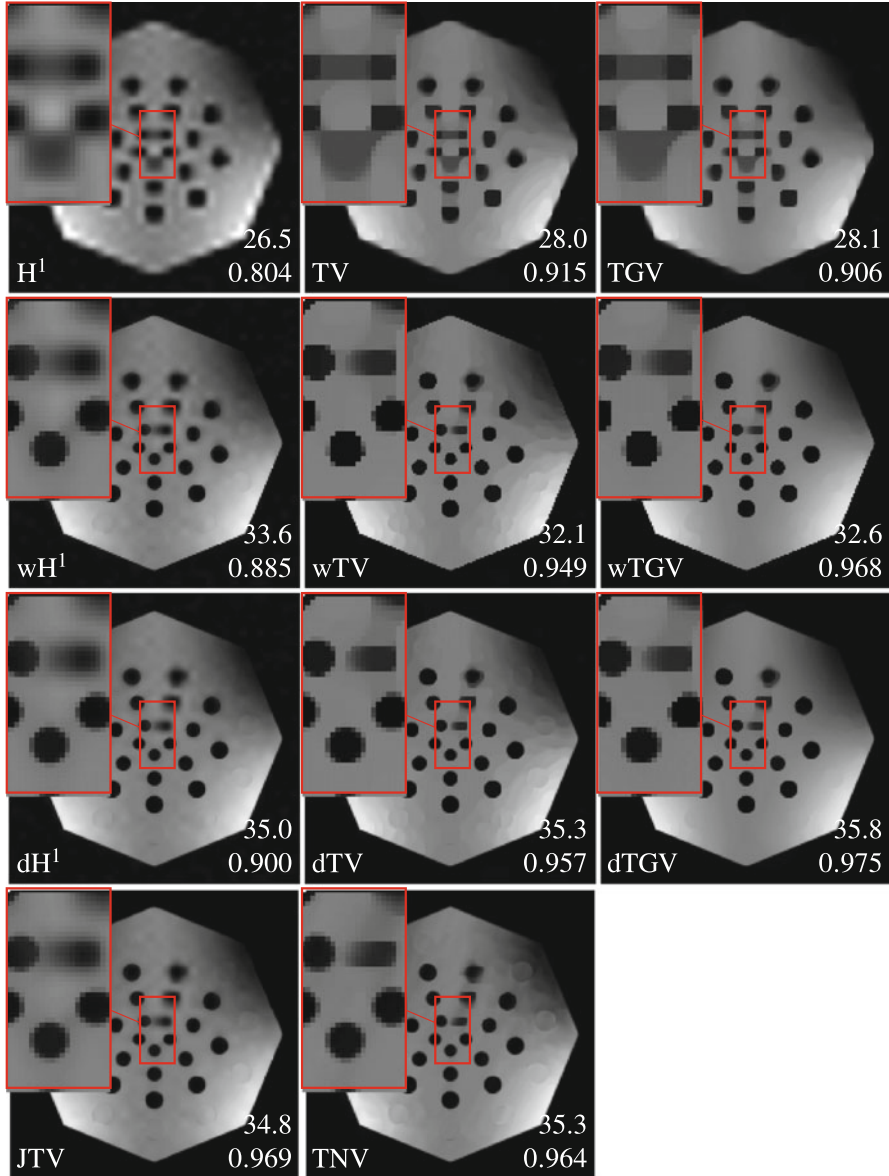


Fig. 19 Comparison of structure-promoting regularizers for test case super-resolution. All parameters were tuned to maximize the PSNR and visual image quality

Table 3 Computing times and PSNR for all tested regularizers

regularizer	Computing time		PSNR	
	x-ray	super-resolution	x-ray	super-resolution
H^1	30.43 s	18.34 s	24.7	26.5
w H^1	34.95 s	22.31 s	29.5	33.6
d H^1	40.87 s	27.80 s	30.9	35.0
TV	32.72 s	18.63 s	28.3	28.0
wTV	38.17 s	22.05 s	32.9	32.1
dTV	44.91 s	29.48 s	37.3	35.3
TGV	71.33 s	52.70 s	28.5	28.1
wTGV	77.67 s	58.44 s	34.3	32.6
dTGV	83.34 s	61.65 s	39.9	35.8
JTV	53.04 s	39.05 s	34.0	34.8
TNV	318.45 s	290.42 s	36.3	35.3

extend to multiple images as side information, but this has not yet been properly investigated.

Applications As we illustrated in section “[Introduction](#),” there are many applications where structure-promoting regularizers were already used or are on the horizon. The list of potential target applications grows steadily with more and more multi-modality scanners being introduced. Next to the misregistration mentioned before, the biggest hurdle in real applications is the interpretation of images that were created by fusing information from several modalities. A common question is “Which edges can I trust?” since often the reconstruction from multi-modality data would be performed on a finer resolution than for the single-modality case. For example, for PET-MR the reconstruction of PET data with an already reconstructed MR image as side information can be performed on the native MRI resolution. The answer might be that such an image should not be interpreted as a PET image, but in fact as a synergistic PET-MR image.

Joint reconstruction Throughout this chapter the focus was on improving the reconstruction of one image with the aid of another modality used as side information. Since the other image is rarely acquired directly, it is natural to aim to reconstruct both images simultaneously rather than sequentially. While conceptually appealing this strategy leads to many more complications than the approach discussed in this chapter which is sometimes referred to as one-sided reconstruction. While the mathematical framework for one-sided reconstruction is quite mature, the framework for joint reconstruction is despite a lot of research effort in the last 10 years still in its infancy. Fundamental problems like computationally tractable and efficient coupling of modalities are still unsolved. The appealing strategy of making use of the solid mathematical foundations of one-sided reconstruction for joint reconstruction in a mathematical sound and computationally tractable way is still not possible to date.

Acknowledgments The author acknowledges support from the EPSRC grant EP/S026045/1 and the Faraday Institution EP/T007745/1. Moreover, the author is grateful to all his collaborators which indirectly contributed to this chapter over the last couple of years.

References

- van Aarle, W., Palenstijn, W.J., Cant, J., Janssens, E., Bleichrodt, F., Dabrovolski, A., De Beenhouwer, J., Joost Batenburg, K., Sijbers, J.: Fast and flexible X-ray tomography using the ASTRA toolbox. *Optics Express* **24**(22), 25129 (2016). <https://doi.org/10.1364/OE.24.025129>
- van Aarle, W., Palenstijn, W.J., De Beenhouwer, J., Altantzis, T., Bals, S., Batenburg, K.J., Sijbers, J.: The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy* **157**, 35–47 (2015). <https://doi.org/10.1016/j.ultramic.2015.05.002>
- Adler, J., Kohr, H., Öktem, O.: Operator Discretization Library (ODL) (2017). <https://doi.org/10.5281/zenodo.249479>
- Ametova, E., Fardell, G., Jørgensen, J.S., Lionheart, W.R.B., Papoutsellis, E., Pasca, E., Sykes, D., Turner, M., Warr, R., Withers, P.J.: Core Imaging Library (CIL) (2019). <https://www.ccpi.ac.uk/cil>
- Arridge, S.R., Burger, M., Ehrhardt, M.J.: Preface to special issue on joint reconstruction and multi-modality/multi-spectral imaging. *Inverse Prob.* **36**, 020302 (2020)
- Arridge, S.R., Kolehmainen, V., Schweiger, M.J.: Reconstruction and regularisation in optical tomography. In: Censor, A., Jiang, Y., Louis, M. (eds.) *Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)*. Scuola Normale Superiore (2008)
- Arridge, S.R., Simmons, A.: Multi-spectral probabilistic diffusion using Bayesian classification. In: ter Haar Romeny, B.M., Florack, L., Koenderink, J.J., Viergever M.A. (eds.) *Scale-Space Theories in Computer Vision*, pp. 224–235. Springer, Berlin (1997). https://doi.org/10.1007/3-540-63167-4_53
- Baete, K., Nuyts, J., Van Paesschen, W., Suetens, P., Dupont, P.: Anatomical-based FDG-PET reconstruction for the detection of hypo-metabolic regions in epilepsy. *IEEE Trans. Med. Imaging* **23**(4), 510–519 (2004). <https://doi.org/10.1109/TMI.2004.825623>
- Bai, B., Li, Q., Leahy, R.M.: Magnetic resonance-guided positron emission tomography image reconstruction. *Semin. Nucl. Med.* **43**, 30–44 (2013). <https://doi.org/10.1053/j.semnuclmed.2012.08.006>
- Ballester, C., Caselles, V., Igual, L., Verdera, J., Rougé, B.: A variational model for P+XS image fusion. *Int. J. Comput. Vis.* **69**(1), 43–58 (2006). <https://doi.org/10.1007/s11263-006-6852-x>
- Bathke, C., Kluth, T., Maass, P.: Improved image reconstruction in magnetic particle imaging using structural a priori information. *Int. J. Magn. Part. Imaging* **3**(1) (2017)
- Bauschke, H.H., Combettes, P.L.: *Convex analysis and monotone operator theory in Hilbert spaces* (2011). <https://doi.org/10.1007/978-1-4419-9467-7>
- Benning, M., Burger, M.: Modern regularization methods for inverse problems. *Acta Numerica* **27**, 1–111 (2018). <https://doi.org/10.1017/S0962492918000016>
- Bilgic, B., Goyal, V.K., Adalsteinsson, E.: Multi-contrast reconstruction with Bayesian compressed sensing. *Magn. Reson. Med.* **66**(6), 1601–1615 (2011). <https://doi.org/10.1002/mrm.22956>
- Blomgren, P., Chan, T.F.: Color TV: Total variation methods for restoration of vector-valued images. *IEEE Trans. Image Process.* **7**(3), 304–309 (1998). <https://doi.org/10.1109/83.661180>
- Bousse, A., Pedemonte, S., Kazantsev, D., Ourselin, S., Arridge, S.R., Hutton, B.F.: Weighted MRI-based Bowsher priors for SPECT brain image reconstruction. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference*, pp. 3519–3522 (2010)
- Bousse, A., Pedemonte, S., Thomas, B.A., Erlandsson, K., Ourselin, S., Arridge, S.R., Hutton, B.F.: Markov random field and Gaussian mixture for segmented MRI-based partial volume correction in PET. *Phys. Med. Biol.* **57**(20), 6681–6705 (2012). <https://doi.org/10.1088/0031-9155/57/20/6681>

- Bowsher, J.E., Johnson, V.E., Turkington, T.G., Jaszczak, R.J., Floyd, C.E., Coleman, R.E.: Bayesian reconstruction and use of anatomical a priori information for emission tomography. *IEEE Trans. Med. Imaging* **15**(5), 673–686 (1996). <https://doi.org/10.1109/42.538945>
- Bowsher, J.E., Yuan, H., Hedlund, L.W., Turkington, T.G., Akabani, G., Badae, A., Kurylo, W.C., Wheeler, C.T., Cofer, G.P., Dewhurst, M.W., Johnson, G.A.: Utilizing MRI information to estimate F18-FDG distributions in rat flank tumors. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference*, pp. 2488–2492 (2004). <https://doi.org/10.1109/NSSMIC.2004.1462760>
- Bredies, K., Dong, Y., Hintermüller, M.: Spatially dependent regularization parameter selection in total generalized variation models for image restoration. *Int. J. Comput. Math.* 1–15 (2012). <https://doi.org/10.1080/00207160.2012.700400>
- Bredies, K., Holler, M.: Regularization of linear inverse problems with total generalized variation. *J. Inverse Ill-Posed Prob.* **22**(6), 871–913 (2014). <https://doi.org/10.1515/jip-2013-0068>
- Bredies, K., Holler, M.: A TGV-based framework for variational image decomposition, zooming, and reconstruction. Part II: Numerics. *SIAM J. Imag. Sci.* **8**(4), 2851–2886 (2015). <https://doi.org/10.1137/15M1023877>
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imag. Sci.* **3**(3), 492–526 (2010). <https://doi.org/10.1137/090769521>
- Bredies, K., Lorenz, D.A.: *Mathematical Image Processing*, 1 edn. Birkhäuser Basel (2018). <https://doi.org/10.1007/978-3-030-01458-2>
- Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Prob. Imaging* **2**(4), 455–484 (2008). <https://doi.org/10.3934/ipi.2008.2.455>
- Bungert, L., Coomes, D.A., Ehrhardt, M.J., Rasch, J., Reisenhofer, R., Schönlieb, C.B.: Blind image fusion for hyperspectral imaging with the directional total variation. *Inverse Prob.* **34**(4), 044003 (2018). <https://doi.org/10.1088/1361-6420/aaaf63>
- Bungert, L., Ehrhardt, M.J.: Robust image reconstruction with misaligned structural information (2020). <http://arxiv.org/abs/2004.00589>
- Bungert, L., Ehrhardt, M.J., Reisenhofer, R.: Robust blind image fusion for misaligned hyperspectral imaging data. In: *Proceedings in Applied Mathematics & Mechanics*, vol. 18, p. e201800033 (2018). <https://doi.org/10.1002/pamm.201800033>
- Burger, M., Osher, S.: A guide to the TV zoo. In: *Level Set and PDE Based Reconstruction Methods in Imaging*, Lecture Notes in Mathematics, vol. 2090, pp. 1–70. Springer (2013). <https://doi.org/10.1007/978-3-319-01712-9>
- Caselles, V., Coll, B., Morel, J.M.: Geometry and color in natural images. *J. Math. Imaging Vision* **16**(Section 2), 89–105 (2002). <https://doi.org/10.1023/A:1013943314097>
- Chambolle, A., Ehrhardt, M.J., Richtárik, P., Schönlieb, C.B.: Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM J. Optim.* **28**(4), 2783–2808 (2018). <https://doi.org/10.1007/s10851-010-0251-1>
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* **40**(1), 120–145 (2011). <https://doi.org/10.1007/s10851-010-0251-1>
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numerica* **25**, 161–319 (2016). <https://doi.org/10.1017/S096249291600009X>
- Chan, C., Fulton, R., Feng, D.D., Cai, W., Meikle, S.: An anatomically based regionally adaptive prior for MAP reconstruction in emission tomography. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference*, pp. 4137–4141 (2007). <https://doi.org/10.1109/NSSMIC.2007.4437032>
- Chan, C., Fulton, R., Feng, D.D., Meikle, S.: Regularized image reconstruction with an anatomically adaptive prior for positron emission tomography. *Phys. Med. Biol.* **54**(24), 7379–400 (2009). <https://doi.org/10.1088/0031-9155/54/24/009>
- Chen, C., Li, Y., Huang, J.: Calibrationless parallel MRI with joint total variation regularization. In: *Medical Image Computing and Computer-Assisted Intervention*, pp. 106–114 (2013). https://doi.org/10.1007/978-3-642-40760-4_14

- Cheng-Liao, J., Qi, J.: PET image reconstruction with anatomical edge guided level set prior. *Phys. Med. Biol.* **56**, 6899–6918 (2011). <https://doi.org/10.1088/0031-9155/56/21/009>
- Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. Springer Optim. Appl. **49**, 185–212 (2011). https://doi.org/10.1007/978-1-4419-9569-8_10
- Comtat, C., Kinahan, P.E., Fessler, J.A., Beyer, T., Townsend, D.W., Defrise, M., Michel, C.J.: Clinically feasible reconstruction of 3D whole-body PET/CT data using blurred anatomical labels. *Phys. Med. Biol.* **47**(1), 1–20 (2002)
- Davies, M., Puy, G., Vandergheynst, P., Wiaux, Y.: A compressed sensing framework for magnetic resonance fingerprinting. *SIAM J. Imag. Sci.* **7**(4), 2623–2656 (2013). <https://doi.org/10.1137/130947246>
- Deidda, D., Karakatsanis, N.A., Robson, P.M., Tsai, Y.J., Efthimiou, N., Thielemans, K., Fayad, Z.A., Aykroyd, R.G., Tsoumpas, C.: Hybrid PET-MR list-mode kernelized expectation maximization reconstruction. *Inverse Prob.* **35**(4) (2019). <https://doi.org/10.1088/1361-6420/ab013f>
- Deligiannis, N., Mota, J.F., Cornelis, B., Rodrigues, M.R., Daubechies, I.: Multi-modal dictionary learning for image separation with application in art investigation. *IEEE Trans. Image Process.* **26**(2), 751–764 (2017). <https://doi.org/10.1109/TIP.2016.2623484>
- Delso, G., Furst, S., Jakoby, B., Ladebeck, R., Ganter, C., Nekolla, S.G., Schwaiger, M., Ziegler, S.I., Fürst, S., Jakoby, B., Ladebeck, R., Ganter, C., Nekolla, S.G., Schwaiger, M., Ziegler, S.I.: Performance measurements of the Siemens mMR integrated whole-body PET/MR scanner. *J. Nucl. Med.* **52**(12), 1914–22 (2011). <https://doi.org/10.2967/jnumed.111.092726>
- Di, Z.W., Leyffer, S., Wild, S.M.: Optimization-based approach for joint X-Ray fluorescence and transmission tomographic inversion. *SIAM J. Imag. Sci.* **9**(1), 1–23 (2016)
- Dong, G., Hintermüller, M., Papafitsoros, K.: Quantitative magnetic resonance imaging: From fingerprinting to integrated physics-based models. *SIAM J. Imag. Sci.* **12**(2), 927–971 (2019). <https://doi.org/10.1137/18M1222211>
- Dong, Y., Hintermüller, M., Rincon-Camacho, M.M.: Automated regularization parameter selection in multi-scale total variation models for image restoration. *J. Math. Imaging Vision* **40**(1), 82–104 (2011). <https://doi.org/10.1007/s10851-010-0248-9>
- Duran, J., Buades, A., Coll, B., Sbert, C., Blanchet, G.: A survey of pansharpening methods with a new band-decoupled variational model. *ISPRS J. Photogramm. Remote Sens.* **125**, 78–105 (2017). <https://doi.org/10.1016/j.isprsjprs.2016.12.013>
- Ehrhardt, M.J.: Joint reconstruction for multi-modality imaging with common structure. Ph.d. thesis, University College London (2015)
- Ehrhardt, M.J., Arridge, S.R.: Vector-valued image processing by parallel level sets. *IEEE Trans. Image Process.* **23**(1), 9–18 (2014). <https://doi.org/10.1109/TIP.2013.2277775>
- Ehrhardt, M.J., Betcke, M.M.: Multi-contrast MRI reconstruction with structure-guided total variation. *SIAM J. Imag. Sci.* **9**(3), 1084–1106 (2016). <https://doi.org/10.1137/15M1047325>
- Ehrhardt, M.J., Markiewicz, P.J., Liljeroth, M., Barnes, A., Kolehmainen, V., Duncan, J., Pizarro, L., Atkinson, D., Hutton, B.F., Ourselin, S., Thielemans, K., Arridge, S.R.: PET reconstruction with an anatomical MRI prior using parallel level sets. *IEEE Trans. Med. Imaging* **35**(9), 2189–2199 (2016). <https://doi.org/10.1109/TMI.2016.2549601>
- Ehrhardt, M.J., Markiewicz, P.J., Schönlieb, C.B.: Faster PET reconstruction with non-smooth priors by randomization and preconditioning. *Phys. Med. Biol.* **64**(22), 225019 (2019). <https://doi.org/10.1088/1361-6560/ab3d07>
- Ehrhardt, M.J., Thielemans, K., Pizarro, L., Atkinson, D., Ourselin, S., Hutton, B.F., Arridge, S.R.: Joint reconstruction of PET-MRI by exploiting structural similarity. *Inverse Prob.* **31**(1), 015001 (2015). <https://doi.org/10.1088/0266-5611/31/1/015001>
- Ehrhardt, M.J., Thielemans, K., Pizarro, L., Markiewicz, P.J., Atkinson, D., Ourselin, S., Hutton, B.F., Arridge, S.R.: Joint reconstruction of PET-MRI by parallel level sets. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference* (2014). <https://doi.org/10.1109/NSSMIC.2014.7430895>
- Elbau, P., Mindrinos, L., Scherzer, O.: Quantitative reconstructions in multi-modal photoacoustic and optical coherence tomography imaging. *Inverse Prob.* **34**(1) (2018). <https://doi.org/10.1088/1361-6420/aa9ae7>

- Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Mathematics and Its Applications. Springer (1996)
- Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.* **3**(4), 1015–1046 (2010). <https://doi.org/10.1137/09076934X>
- Estellers, V., Soatto, S., Bresson, X.: Adaptive regularization with the structure tensor. *IEEE Trans. Image Process.* **24**(6), 1777–1790 (2015). <https://doi.org/10.1109/TIP.2015.2409562>
- Estellers, V., Thiran, J., Bresson, X.: Enhanced compressed sensing recovery with level set normals. *IEEE Trans. Image Process.* **22**(7), 2611–2626 (2013). <https://doi.org/10.1109/TIP.2013.2253484>
- Fang, F., Li, F., Shen, C., Zhang, G.: A variational approach for pan-sharpening. *IEEE Trans. Image Process.* **22**(7), 2822–2834 (2013). <https://doi.org/10.1109/TIP.2013.2258355>
- Fessler, J.A., Elbakri, I., Sukovic, P., Clinthorne, N.H.: Maximum-likelihood dual-energy tomographic image reconstruction. In: *SPIE: Medical Imaging*, vol. 4684, pp. 1–25 (2002). <https://doi.org/doi:10.1117/12.467189>
- Foygel Barber, R., Sidky, E.Y., Gilat Schmidt, T., Pan, X.: An algorithm for constrained one-step inversion of spectral CT data. *Phys. Med. Biol.* **61**(10), 3784–3818 (2016). <https://doi.org/10.1088/0031-9155/61/10/3784>
- Gallardo, L.A., Meju, M.A.: Characterization of heterogeneous near-surface materials by joint 2D inversion of DC resistivity and seismic data. *Geophys. Res. Lett.* **30**(13), 1658 (2003). <https://doi.org/10.1029/2003GL017370>
- Gallardo, L.A., Meju, M.A.: Joint two-dimensional DC resistivity and seismic travel time inversion with cross-gradients constraints. *J. Geophys. Res.* **109**(B3), 1–11 (2004). <https://doi.org/10.1029/2003JB002716>
- Gallardo, L.A., Meju, M.A.: Structure-coupled multiphysics imaging in geophysical sciences. *Rev. Geophys.* **49**, 1–19 (2011). <https://doi.org/10.1029/2010RG000330.1.INTRODUCTION>
- Golbabaee, M., Chen, Z., Wiaux, Y., Davies, M.: CoverBLIP: accelerated and scalable iterative matched-filtering for magnetic resonance fingerprint reconstruction. *Inverse Prob.* **36**(1), 015003 (2020). <https://doi.org/10.1088/1361-6420/ab4c9a>
- Goldluecke, B., Strelakovsky, E., Cremers, D.: The natural vectorial total variation which arises from geometric measure theory. *SIAM J. Imag. Sci.* **5**(2), 537–563 (2012). <https://doi.org/10.1137/110823766>
- Haber, E., Holtzman-Gazit, M.: Model fusion and joint inversion. *Surv. Geophys.* (34), 675–695 (2013). <https://doi.org/10.1007/s10712-013-9232-4>
- Haber, E., Modersitzki, J.: Intensity gradient based registration and fusion of multi-modal images. In: *Medical Image Computing and Computer-Assisted Intervention*, vol. 46, pp. 726–733. Springer, Berlin/Heidelberg (2006). <https://doi.org/10.1160/ME9046>
- Haber, E., Oldenburg, D.W.: Joint inversion: A structural approach. *Inverse Prob.* **13**, 63–77 (1997). <https://doi.org/10.1088/0266-5611/13/1/006>
- Heismann, B., Schmidt, B., Flohr, T.: *Spectral Computed Tomography*. SPIE Press (2012)
- Hintermüller, M., Rincon-Camacho, M.M.: Expected absolute value estimators for a spatially adapted regularization parameter choice rule in L1-TV-based image restoration. *Inverse Prob.* **26**(8), 085005 (2010). <https://doi.org/10.1088/0266-5611/26/8/085005>
- Holt, K.M.: Total nuclear variation and jacobian extensions of total variation for vector fields. *IEEE Trans. Image Process.* **23**(9), 3975–3989 (2014). <https://doi.org/10.1109/TIP.2014.2332397>
- Huang, J., Chen, C., Axel, L.: Fast Multi-contrast MRI reconstruction. *Magn. Reson. Imaging* **32**(10), 1344–52 (2014). <https://doi.org/10.1016/j.mri.2014.08.025>
- Huber, R., Haberfehlner, G., Holler, M., Bredies, K.: Total generalized variation regularization for multi-modal electron tomography. *Nanoscale* 1–38 (2019). <https://doi.org/10.1039/c8nr09058k>
- Ito, K., Jin, B.: *Inverse Problems – Tikhonov Theory and Algorithms*. World Scientific Publishing (2014). <https://doi.org/10.1142/9120>
- Kaipio, J.P., Kolehmainen, V., Vauhkonen, M., Somersalo, E.: Inverse problems with structural prior information. *Inverse Prob.* **15**(3), 713–729 (1999). <https://doi.org/10.1088/0266-5611/15/3/306>

- Kazantsev, D., Arridge, S.R., Pedemonte, S., Bousse, A., Erlandsson, K., Hutton, B.F., Ourselin, S.: An anatomically driven anisotropic diffusion filtering method for 3D SPECT reconstruction. *Phys. Med. Biol.* **57**(12), 3793–3810 (2012). <https://doi.org/10.1088/0031-9155/57/12/3793>
- Kazantsev, D., Jørgensen, J.S., Andersen, M.S., Lionheart, W.R., Lee, P.D., Withers, P.J.: Joint image reconstruction method with correlative multi-channel prior for x-ray spectral computed tomography. *Inverse Prob.* **34**(6) (2018). <https://doi.org/10.1088/1361-6420/aaba86>
- Kazantsev, D., Lionheart, W.R.B., Withers, P.J., Lee, P.D.: Multimodal image reconstruction using supplementary structural information in total variation regularization. *Sens. Imaging* **15**(1), 97 (2014). <https://doi.org/10.1007/s11220-014-0097-5>
- Kimmel, R., Malladi, R., Sochen, N.: Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *Int. J. Comput. Vis.* **39**(2), 111–129 (2000). <https://doi.org/10.1023/A:1008171026419>
- Knoll, F., Holler, M., Koesters, T., Otazo, R., Bredies, K., Sodickson, D.K.: Joint MR-PET reconstruction using a multi-channel image regularizer. *IEEE Trans. Med. Imaging* **36**(1) (2016). <https://doi.org/10.1109/TMI.2016.2564989>
- Knoll, F., Koesters, T., Otazo, R., Boada, F., Sodickson, D.K.: Simultaneous MR-PET reconstruction using multi sensor compressed sensing and joint sparsity. In: *International Society for Magnetic Resonance in Medicine*, vol. 22 (2014)
- Kolehmainen, V., Ehrhardt, M.J., Arridge, S.R.: Incorporating structural prior information and sparsity into EIT using parallel level sets. *Inverse Prob. Imaging* **13**(2), 285–307 (2019). <https://doi.org/10.3934/ipi.2019015>
- Leahy, R.M., Yan, X.: Incorporation of anatomical MR data for improved functional imaging with PET. In: *Information Processing in Medical Imaging*, pp. 105–120. Springer (1991). <https://doi.org/10.1007/BFb0033746>
- Lenzen, F., Berger, J.: Solution-driven adaptive total variation regularization. In: *SSVM*, pp. 203–215 (2015). <https://doi.org/10.1007/978-3-642-24785-9>
- Loncan, L., De Almeida, L.B., Bioucas-Dias, J.M., Briottet, X., Chanussot, J., Dobigeon, N., Fabre, S., Liao, W., Licciardi, G.A., Simoes, M., Tourneret, J.Y., Veganzones, M.A., Vivone, G., Wei, Q., Yokoya, N.: Hyperspectral pansharpening: a review. *IEEE Geosci. Remote Sens. Mag.* **3**(3), 27–46 (2015). <https://doi.org/10.1109/MGRS.2015.2440094>
- Long, Y., Fessler, J.A.: Multi-material decomposition using statistical image reconstruction for spectral CT. *IEEE Trans. Med. Imaging* **33**(8), 1614–1626 (2014). <https://doi.org/10.1109/TMI.2014.2320284>
- Ma, D., Gulani, V., Seiberlich, N., Liu, K., Sunshine, J.L., Duerk, J.L., Griswold, M.A.: Magnetic resonance fingerprinting. *Nature* **495**(7440), 187–92 (2013). <https://doi.org/10.1038/nature11971>
- Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. *IEEE Trans. Med. Imaging* **16**(2), 187–98 (1997). <https://doi.org/10.1109/42.563664>
- Mehranian, A., Belzunce, M., Prieto, C., Hammers, A., Reader, A.J.: Synergistic PET and SENSE MR image reconstruction using joint sparsity regularization. *IEEE Trans. Med. Imaging* **37**(1), 20–34 (2018). <https://doi.org/10.1109/TMI.2017.2691044>
- Mehranian, A., Belzunce, M.A., Niccolini, F., Politis, M., Prieto, C., Turkheimer, F., Hammers, A., Reader, A.J.: PET image reconstruction using multi-parametric anato-functional. *Phys. Med. Biol.* (2017). <https://doi.org/10.1042/BJ20101136>
- Meju, M.A., Mackie, R.L., Miorelli, F., Saleh, A.S., Miller, R.V.: Structurally-tailored 3D anisotropic CSEM resistivity inversion with cross-gradients criterion and simultaneous model calibration. *Geophysics* **84**(6), 1–62 (2019). <https://doi.org/10.1190/geo2018-0639.1>
- Möller, M., Brinkmann, E.M., Burger, M., Seybold, T.: Color Bregman TV. *SIAM J. Imag. Sci.* **7**(4), 2771–2806 (2014). <https://doi.org/10.1137/130943388>
- Möller, M., Wittman, T., Bertozzi, A.L., Burger, M.: A variational approach for sharpening high dimensional images. *SIAM J. Imag. Sci.* **5**(1), 150–178 (2012). <https://doi.org/10.1137/100810356>

- Nuyts, J.: The use of mutual information and joint entropy for anatomical priors in emission tomography. In: IEEE Nuclear Science Symposium and Medical Imaging Conference, pp. 4149–4154. IEEE (2007). <https://doi.org/10.1109/NSSMIC.2007.4437034>
- Obert, A.J., Gutberlet, M., Kern, A.L., Kaireit, T.F., Grimm, R., Wacker, F., Vogel-Claussen, J.: 1H-guided reconstruction of 19F gas MRI in COPD patients. *Magn. Reson. Med.* 1–11 (2020). <https://doi.org/10.1002/mrm.28209>
- Parikh, N., Boyd, S.P.: Proximal algorithms. *Found Trends Optim* 1(3), 123–231 (2014). <https://doi.org/10.1561/24000000003>
- Pedemonte, S., Bousse, A., Hutton, B.F., Arridge, S.R., Ourselin, S.: Probabilistic graphical model of SPECT/MRI. In: *Machine Learning in Medical Imaging*, pp. 167–174 (2011). https://doi.org/10.1007/978-3-642-24319-6_21
- Pluim, J.P.W., Maintz, J.B.A., Viergever, M.A.: Image registration by maximization of combined mutual information and gradient information. *IEEE Trans. Med. Imaging* 19(8), 809–14 (2000). <https://doi.org/10.1109/42.876307>
- Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1762–1769 (2011). <https://doi.org/10.1109/ICCV.2011.6126441>
- Rangarajan, A., Hsiao, I.T., Gindi, G.: A Bayesian joint mixture framework for the integration of anatomical information in functional image reconstruction. *J. Math. Imaging Vision* 12(3), 199–217 (2000). <https://doi.org/10.1023/A:1008314015446>
- Rasch, J., Brinkmann, E.M., Burger, M.: Joint reconstruction via coupled bregman iterations with applications to PET-MR imaging. *Inverse Prob.* 34(1), 014001 (2018a). <https://doi.org/10.1088/1361-6420/aa9425>
- Rasch, J., Kolehmainen, V., Nivajarvi, R., Kettunen, M., Gröhn, O., Burger, M., Brinkmann, E.M.: Dynamic MRI reconstruction from undersampled data with an anatomical prescan. *Inverse Prob.* 34(7) (2018b). <https://doi.org/10.1088/1361-6420/aac3af>
- Rigie, D., La Riviere, P.: Joint reconstruction of multi-channel, spectral CT data via constrained total nuclear variation minimization. *Phys. Med. Biol.* 60, 1741–1762 (2015). <https://doi.org/10.1088/0031-9155/60/4/1741>
- Rigie, D.S., Sanchez, A.A., La Rivière, P.J.: Assessment of vectorial total variation penalties on realistic dual-energy CT data. *Phys. Med. Biol.* 62(8), 3284–3298 (2017). <https://doi.org/10.1088/1361-6560/aa6392>
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenom.* 60(1), 259–268 (1992). [https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F)
- Sapiro, G., Ringach, D.L.: Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Image Process.* 5(11), 1582–1586 (1996). <https://doi.org/10.1109/83.541429>
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: *Variational Methods in Imaging*, vol. 167. Springer, New York/London (2008)
- Schmitt, U., Louis, A.K.: Efficient algorithms for the regularization of dynamic inverse problems: I. Theory. *Inverse Problems* 18(3), 645–658 (2002). <https://doi.org/10.1088/0266-5611/18/3/308>
- Schmitt, U., Louis, A.K., Wolters, C., Vauhkonen, M.: Efficient algorithms for the regularization of dynamic inverse problems: II. Applications. *Inverse Prob.* 18(3), 659–676 (2002). <https://doi.org/10.1088/0266-5611/18/3/308>
- Schramm, G., Holler, M., Rezaei, A., Vunckx, K., Knoll, F., Bredies, K., Boada, F., Nuyts, J.: Evaluation of parallel level sets and Bowsher’s method as segmentation-free anatomical priors for time-of-flight PET reconstruction. *IEEE Trans. Med. Imaging* 62(2), 590–603 (2017). <https://doi.org/10.1109/TMI.2017.2767940>
- Schuster, T., Hahn, B., Burger, M.: Dynamic inverse problems: Modelling – Regularization – numerics. *Inverse Prob.* 34(4) (2018). <https://doi.org/10.1088/1361-6420/aab0f5>

- Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. *IEEE Trans. Image Process.* **7**(3), 310–318 (1998). <https://doi.org/10.1109/83.661181>
- Sodickson, D.K., Feng, L., Knoll, F., Cloos, M., Ben-Eliezer, N., Axel, L., Chandarana, H., Block, K.T., Otazo, R.: The rapid imaging renaissance: Sparser samples, denser dimensions, and glimmerings of a grand unified tomography. In: *Proceedings of SPIE*, vol. 9417, pp. 94170G1–9417014 (2015). <https://doi.org/10.1117/12.2085033>
- Somayajula, S., Panagiotou, C., Rangarajan, A., Li, Q., Arridge, S.R., Leahy, R.M.: PET image reconstruction using information theoretic anatomical priors. *IEEE Trans. Med. Imaging* **30**(3), 537–549 (2011). <https://doi.org/10.1109/TMI.2010.2076827>
- Song, P., Deng, X., Mota, J.F.C., Deligiannis, N., Dragotti, P.L., Rodrigues, M.: Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries. *IEEE Trans. Comput. Imaging* **1**–1 (2019). <https://doi.org/10.1109/tci.2019.2916502>
- Song, P., Weizman, L., Mota, J.F., Eldar, Y.C., Rodrigues, M.R.: Coupled dictionary learning for multi-contrast MRI reconstruction. In: *International Conference on Image Processing*, 2, pp. 2880–2884 (2018). <https://doi.org/10.1109/ICIP.2018.8451341>
- Tang, J., Rahmim, A.: Bayesian PET image reconstruction incorporating anato-functional joint entropy. *Phys. Med. Biol.* **54**(23), 7063–75 (2009). <https://doi.org/10.1088/0031-9155/54/23/002>
- Tang, J., Rahmim, A.: Anatomy assisted PET image reconstruction incorporating multi-resolution joint entropy. *Phys. Med. Biol.* **60**(1), 31–48 (2015). <https://doi.org/10.1088/0031-9155/60/1/31>
- Tang, S., Fernandez-Granda, C., Lannuzel, S., Bernstein, B., Lattanzi, R., Cloos, M., Knoll, F., Asslander, J.: Multicompartment magnetic resonance fingerprinting. *Inverse Prob.* **34**(9) (2018). <https://doi.org/10.1088/1361-6420/aad1c3>
- Tsai, Y.J., Member, S., Bousse, A., Ahn, S., Charles, W., Arridge, S., Hutton, B.F., Member, S., Thielemans, K.: Algorithms for solving misalignment issues in penalized PET/CT reconstruction using anatomical priors. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*. IEEE (2018)
- Tschumperlé, D., Deriche, R.: Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(4), 506–517 (2005). <https://doi.org/10.1109/TPAMI.2005.87>
- Vunckx, K., Atre, A., Baete, K., Reilhac, A., Deroose, C.M., Van Laere, K., Nuyts, J.: Evaluation of three MRI-based anatomical priors for quantitative PET brain imaging. *IEEE Trans. Med. Imaging* **31**(3), 599–612 (2012). <https://doi.org/10.1109/TMI.2011.2173766>
- Wang, G., Zhang, J., Gao, H., Weir, V., Yu, H., Cong, W., Xu, X., Shen, H., Bennett, J., Furth, M., Wang, Y., Vannier, M.: Towards omni-tomography – grand fusion of multiple modalities for simultaneous interior tomography. *PloS one* **7**(6), e39700 (2012). <https://doi.org/10.1371/journal.pone.0039700>
- Wells III, W.M., Viola, P., Atsumi, H., Nakajima, S., Kikinis, R.: Multi-modal volume registration by maximization of mutual information. *Med. Image Anal.* **1**(1), 35–51 (1996)
- Xi, Y., Zhao, J., Bennett, J., Stacy, M., Sinusas, A., Wang, G.: Simultaneous CT-MRI reconstruction for constrained imaging geometries using structural coupling and compressive sensing. *IEEE Trans. Biomed. Eng.* (2015). <https://doi.org/10.1109/TBME.2015.2487779>
- Xiang, L., Chen, Y., Chang, W., Zhan, Y., Lin, W., Wang, Q., Shen, D.: Deep-learning-based multi-modal fusion for fast MR reconstruction. *IEEE Trans. Biomed. Eng.* **66**(7), 2105–2114 (2019). <https://doi.org/10.1109/TBME.2018.2883958>
- Yokoya, N., Grohnfeldt, C., Chanussot, J.: Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geosci. Remote Sens. Mag.* **5**(2), 29–56 (2017). <https://doi.org/10.1109/MGRS.2016.2637824>
- Zhang, Y., Zhang, X.: PET-MRI joint reconstruction with common edge weighted total variation regularization. *Inverse Prob.* **34**(6), 065006 (2018). <https://doi.org/10.1088/1361-6420/aabce9>



Diffraction Tomography, Fourier Reconstruction, and Full Waveform Inversion

8

Florian Faucher, Clemens Kirisits, Michael Quellmalz,
Otmar Scherzer, and Eric Setterqvist

Contents

Introduction	274
Contribution and Outline	276
Experimental Setup	276
Forward Models	278
Incident Plane Wave	279
Modeling the Total Field Using Line and Point Sources	281
Numerical Comparison of Forward Models	282
Modeling the Scattered Field Assuming Incident Plane Waves	283

F. Faucher
Faculty of Mathematics, University of Vienna, Vienna, Austria

Project-Team Makutu, Inria Bordeaux Sud-Ouest, Talence, France
e-mail: florian.faucher@inria.fr; florian.faucher@univie.ac.at

C. Kirisits
Faculty of Mathematics, University of Vienna, Vienna, Austria
e-mail: clemens.kirisits@univie.ac.at

M. Quellmalz
Institute of Mathematics, Technical University Berlin, Berlin, Germany
e-mail: quellmalz@math.tu-berlin.de

O. Scherzer (✉)
Faculty of Mathematics, University of Vienna, Vienna, Austria

Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz, Austria
Christian Doppler Laboratory for Mathematical Modeling and Simulation of Next Generations of
Ultrasound Devices (MaMSi), Vienna, Austria
e-mail: otmar.scherzer@univie.ac.at

E. Setterqvist
Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz, Austria
e-mail: eric.setterqvist@ricam.oeaw.ac.at

Modeling the Total Field Using Line and Point Sources	283
Fourier Diffraction Theorem	284
Rotating the Object	286
Varying Wave Number	287
Rotating the Object with Multiple Wave Numbers	288
Reconstruction Methods	289
Reconstruction Using Full Waveform Inversion	289
Reconstruction Based on the Born and Rytov Approximations	291
Numerical Experiments	294
Reconstruction of Circular Contrast with Various Amplitudes and Sizes	294
Reconstruction of Embedded Shapes: Phantom 1	299
Reconstruction of Embedded Shapes: Phantom 2	304
Computational Costs	305
Conclusion	308
References	309

Abstract

In this chapter, we study the mathematical imaging problem of diffraction tomography (DT), which is an inverse scattering technique used to find material properties of an object by illuminating it with probing waves and recording the scattered waves. Conventional DT relies on the Fourier diffraction theorem, which is applicable under the condition of weak scattering. However, if the object has high contrasts or is too large compared to the wavelength, it tends to produce multiple scattering, which complicates the reconstruction. In this chapter, we give a survey on diffraction tomography and compare the reconstruction of low- and high-contrast objects. We also implement and compare the reconstruction using the full waveform inversion method which, contrary to the Born and Rytov approximations, works with the total field and is more robust to multiple scattering.

Keywords

Diffraction tomography · Mathematical imaging · Fourier diffraction theorem · Full waveform inversion · Born approximation · Rytov approximation · Inverse problems

Introduction

Diffraction tomography (DT) is a technique for reconstructing the scattering potential of an object from measurements of waves scattered by that object. DT can be understood as an alternative to, or extension of, classical computerized tomography. In computerized tomography, a crucial assumption is that the radiation, X-rays, for instance, essentially propagates along straight lines through the object. The attenuated rays are recorded and can be related to material properties f of the object by means of the Radon, or X-ray, transform. A central result for the

inversion of this relation is the Fourier slice theorem. Roughly speaking, it says that the Fourier transformed measurements are equal to the Fourier transform of f evaluated along slices through the origin (Natterer 1986).

The straight ray assumption of computerized tomography can be considered valid as long as the wavelength of the incident field is much smaller than the size of the relevant details in the object. As soon as the wavelength is similar to or greater than those details, for instance, in situations where X-rays are replaced by visible light, diffraction effects are no longer negligible. As an example of a medical application, an optical diffraction experiment in Sung et al. (2009) utilized a red laser of wavelength 633 nm to illuminate human cells of diameter around $10\mu\text{m}$, which include smaller subcellular organelles. One way to achieve better reconstruction quality in such cases is to drop the straight ray assumption and adopt a propagation model based on the wave equation instead.

The theoretical groundwork for DT was laid more than half a century ago (Wolf 1969). The central result derived there, sometimes called the *Fourier diffraction theorem*, says that the Fourier transformed measurements of the scattered wave are equal to the Fourier transform of the scattering potential evaluated along a hemisphere. This result relies on a series of assumptions: (i) the object is immersed in a homogeneous background, (ii) the incident field is a monochromatic plane wave, (iii) the scattered wave is measured on a plane in \mathbb{R}^3 , and (iv) the first Born approximation of the scattered field is valid.

On the one hand, the Born approximation greatly simplifies the relationship between scattered wave and scattering potential. On the other hand, however, it generally requires the object to be weakly scattering, thus limiting the applicability of the Fourier diffraction theorem. An alternative is to assume validity of the first Rytov approximation instead (Iwata and Nagata 1975). While mathematically this amounts to essentially the same reconstruction problem, the underlying physical assumptions are not identical to those of the Born approximation, leading to a different range of applicability in general (Chen and Stamnes 1998; Slaney et al. 1984). Nevertheless, the restriction to weakly scattering objects remains.

Full waveform inversion (FWI) is a different approach that can overcome some of the limitations of the first-order methods, typically at the cost of being computationally more demanding. It relies on the iterative minimization of a cost functional which penalizes the misfit between measurements and forward simulations of the total field, cf. Bamberger et al. (1979), Lailly (1983), Pratt et al. (1998), Tarantola (1984), and Virieux and Operto (2009). Here, the forward model consists of the solution of the full wave equation, without simplification of first-order approximations. It results in a nonlinear minimization problem to be solved, typically with Newton-type methods (Virieux and Operto 2009; Nocedal and Wright 2006).

In practical experiments, there are sometimes only measurements of the intensity, i.e., the absolute value of the complex-valued wave, available. Different phase retrieval methods were investigated, e.g., in Maleki and Devaney (1993), Gbur and Wolf (2002), Horstmeyer et al. (2016), and Beinert and Quellmalz (2022). For this chapter, we assume that both the phase and amplitude information are present, which can be achieved by interferometry, cf. Wedberg and Stamnes (1995).

Contribution and Outline

In this chapter, we present a numerical comparison of three reconstruction approaches for diffraction tomography on simulated data, based on (i) the Born approximation, (ii) the Rytov approximation, and (iii) FWI. The setting we use for this comparison is 2D transmission imaging in a homogeneous background with (approximate) plane wave irradiation. The object is assumed to make a full turn during the experiment, providing measurements for a uniform set of incidence angles. In addition, we investigate how providing additional data by varying the wavelength affects the reconstruction. The scattering potentials considered here are test phantoms of varying sizes, shapes, and contrasts. Moreover, for data generation purposes, we compare several forward models.

For numerical reconstruction under the Born and Rytov approximations, a well-known method is the backpropagation algorithm (Devaney 1982), which is widely used in practice, cf. Müller et al. (2015) and also Fan et al. (2017). Our algorithms rely on the nonuniform discrete Fourier transform (NDFT), which was used in 3D Fourier diffraction tomography yielding better results than discrete backpropagation (Kirisits et al. 2021). Our FWI-based reconstruction uses an iterative Newton-type method on an L^2 distance between data and simulations. Here, the discretization of the partial differential equations associated with the wave propagation uses the hybridizable discontinuous Galerkin method (HDG), Cockburn et al. (2009) and Faucher and Scherzer (2020). It is implemented, together with the inverse procedure, in the open-source parallel software *hawen*,¹ Faucher (2021).

The outline of this chapter is as follows. The conceptual experiment is detailed in section “[Experimental Setup](#)”. Forward models are presented in section “[Forward Models](#)”, and their numerical performance is compared in section “[Numerical Comparison of Forward Models](#)”. The Fourier diffraction theorem is formulated and discussed in section “[Fourier Diffraction Theorem](#)”. Further, section “[Reconstruction Methods](#)” covers the reconstruction algorithms used for the numerical experiments, which are presented in section “[Numerical Experiments](#)”. A concluding discussion of our findings is given in section “[Conclusion](#)”.

Experimental Setup

We consider the tomographic reconstruction of a two-dimensional object taking into account diffraction of the incident field. The object is assumed to be embedded in a homogeneous background and illuminated or insonified by a monochromatic plane wave. In fact, for the computational experiments, we implement and compare several approaches to approximate the incident plane wave; see sections “[Forward Models](#)” and “[Numerical Comparison of Forward Models](#)”. We restrict ourselves to transmission imaging, where the incident field propagates in direction $\mathbf{e}_2 = (0, 1)^\top$,

¹<https://ffaucher.gitlab.io/hawen-website/>

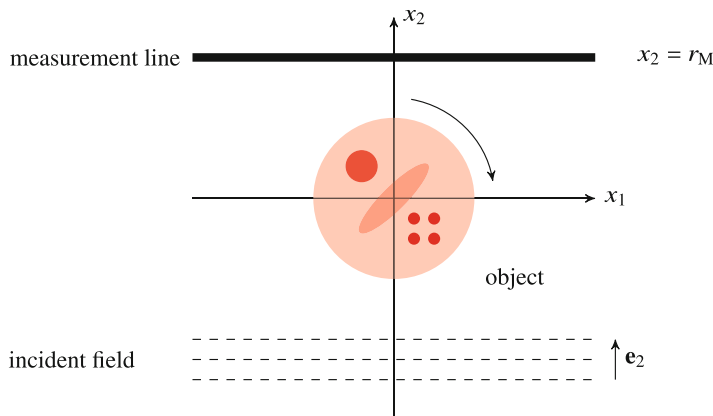


Fig. 1 Experimental setup

and the resulting field is measured on the line $x_2 = r_M$. The distance between the measurement line and the origin, $r_M > 0$, is sufficiently large so that it does not intersect the object. From the measurements, we aim to reconstruct the object's scattering properties. In order to improve the reconstruction quality, we generate additional data by rotating the object or changing the incident field's wavelength. See Fig. 1 for an illustration of the experimental setup.

We now introduce the physical quantities needed subsequently. Let $\lambda > 0$ denote the *wavelength* of the incident wave and $k_0 = 2\pi/\lambda$ the background *wave number*. Furthermore, let $n(\mathbf{x})$ denote the *refractive index* at position $\mathbf{x} \in \mathbb{R}^2$ and n_0 the constant refractive index of the background. From these quantities, we define the wave number

$$k(\mathbf{x}) := k_0 \frac{n(\mathbf{x})}{n_0}.$$

Furthermore, the wave number k can be equivalently expressed in terms of the angular frequency ω and the wave speed c such that

$$k(\mathbf{x}) = \frac{\omega}{c(\mathbf{x})} \quad \text{and} \quad k_0 = \frac{\omega}{c_0}, \quad (1)$$

where c_0 is the constant wave speed in the background. The *scattering potential* f is obtained by subtracting the background wave number k_0 :

$$f(\mathbf{x}) := k^2(\mathbf{x}) - k_0^2 = k_0^2 \left(\frac{n(\mathbf{x})^2}{n_0^2} - 1 \right). \quad (2)$$

Note that, for all practical purposes, f can be assumed to be bounded and compactly supported in the disk $\mathcal{B}_{r_M} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| < r_M\}$.

In our subsequent reconstructions with Born and Rytov approximations, f is the quantity to be reconstructed from the measured data and k_0 is known. On the other hand, with FWI, we reconstruct c ; see Remark 2. These two quantities can be related to each other via

$$c(\mathbf{x}) = \sqrt{\frac{\omega^2}{k_0^2 + f(\mathbf{x})}}. \quad (3)$$

Forward Models

In this section, we propose several forward models for the experiment presented above. For all of them, the starting point is the system of equations

$$\left. \begin{aligned} (-\Delta - k(\mathbf{x})^2) u^{\text{tot}}(\mathbf{x}) &= g(\mathbf{x}), \\ (-\Delta - k_0^2) u^{\text{inc}}(\mathbf{x}) &= g(\mathbf{x}), \\ u^{\text{tot}}(\mathbf{x}) &= u^{\text{inc}}(\mathbf{x}) + u^{\text{sca}}(\mathbf{x}), \end{aligned} \right\} \mathbf{x} \in \mathbb{R}^2. \quad (4)$$

Here, u^{inc} is the given *incident field*, the *total field* u^{tot} is what is recorded on the measurement line $\{\mathbf{x} \in \mathbb{R}^2 : x_2 = r_M\}$, and the difference between the two constitutes the *scattered field* u^{sca} . We describe different sources g in the following subsections. The scattered field u^{sca} is assumed to satisfy the Sommerfeld radiation condition which requires that

$$\lim_{\|\mathbf{x}\| \rightarrow \infty} \sqrt{\|\mathbf{x}\|} \left(\frac{\partial u^{\text{sca}}}{\partial \|\mathbf{x}\|} - ik_0 u^{\text{sca}} \right) = 0$$

uniformly for all directions $\mathbf{x}/\|\mathbf{x}\|$. It guarantees that u^{sca} is an outgoing wave. Further details concerning derivation and analytical properties of problems like Equation 4 can be found, for instance, in Colton and Kress (2013).

The models considered below are based on the following specifications of Equation 4. Their numbers agree with the corresponding subsection numbers, where the models will be explained in more detail.

1. Plane wave No source ($g = 0$) and u^{inc} is an ideal plane wave; see Equation 5.
- 1.1 Born model No source, u^{inc} is an ideal plane wave and u^{sca} satisfies the Born approximation.
- 1.2 Rytov model No source, u^{inc} is an ideal plane wave, and u^{sca} satisfies the Rytov approximation.

In addition, we propose the following two models with sources:

- 2.1 Point source g represents a point source located far from the object.
- 2.2 Line source g represents simultaneous point sources positioned along a straight line. We refer to this configuration as a “line source”.

Section “[Numerical Comparison of Forward Models](#)” contains a numerical comparison of these forward models.

The proposed selection of equations is motivated in part by the availability of methods for their numerical inversion. While the Born and Rytov models can be inverted using nonuniform Fourier methods, the point and line source models are well-suited for FWI.

Incident Plane Wave

Monochromatic plane waves are basic solutions u of the homogeneous Helmholtz equation

$$(-\Delta - k_0^2)u = 0.$$

They take the form $u(\mathbf{x}) = e^{ik_0\mathbf{x}\cdot\mathbf{s}}$, where the unit vector \mathbf{s} specifies the direction of propagation of u . Plane waves are widely studied in imaging applications and theory, and we refer to Colton and Kress (2013), Devaney (2012), and Kak and Slaney (2001) for further information.

In the first model, we consider the incident field is a monochromatic plane wave propagating in direction \mathbf{e}_2

$$u^{\text{inc}}(\mathbf{x}) = e^{ik_0x_2}. \quad (5)$$

In this case, we obtain from Equation 4 the following equation for the scattered field

$$(-\Delta - k(\mathbf{x})^2) u^{\text{sca}}(\mathbf{x}) = f(\mathbf{x}) e^{ik_0x_2}. \quad (6)$$

The Born Approximation

Equation 6 can be written as

$$(-\Delta - k_0^2) u^{\text{sca}}(\mathbf{x}) = f(\mathbf{x}) (e^{ik_0x_2} + u^{\text{sca}}(\mathbf{x})).$$

If the scattered field u^{sca} is negligible compared to the incident field $e^{ik_0x_2}$, we can ignore u^{sca} on the right-hand side and obtain

$$(-\Delta - k_0^2) u^{\text{Born}}(\mathbf{x}) = f(\mathbf{x}) e^{ik_0x_2}, \quad (7)$$

where u^{Born} is the (*first-order*) *Born approximation* to the scattered field. Supplementing this equation with the Sommerfeld radiation condition, we have a unique solution corresponding to an outgoing wave (Colton and Kress 2013). It can be written as a convolution

$$u^{\text{Born}}(\mathbf{x}) = \int_{\mathbb{R}^2} G(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) e^{ik_0 y_2} d\mathbf{y}, \quad (8)$$

where G is the outgoing Green's function for the Helmholtz equation. In \mathbb{R}^2 , it is given by

$$G(\mathbf{x}) = \frac{i}{4} H_0^{(1)}(k_0 \|\mathbf{x}\|), \quad \mathbf{x} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}, \quad (9)$$

where $H_0^{(1)}$ denotes the zeroth-order Hankel function of the first kind; see Colton and Kress (2013, Sect. 3.4). We note that, in spite of a singularity at the origin, G is locally integrable in \mathbb{R}^2 .

The second-order Born approximation can be obtained by replacing the plane wave $e^{ik_0 y_2}$ in Equation 8 by the sum $e^{ik_0 y_2} + u^{\text{Born}}(\mathbf{y})$. Iterating this procedure yields Born approximations of arbitrary order. For more details, we refer to Kak and Slaney (2001, Sect. 6.2.1) and Devaney (2012).

The Rytov Approximation

In this subsection, we derive an alternative approximation for the scattered field. Introducing the complex phases φ^{inc} , φ^{tot} , and φ^{sca} according to

$$u^{\text{tot}} = e^{\varphi^{\text{tot}}}, \quad u^{\text{inc}} = e^{\varphi^{\text{inc}}}, \quad \varphi^{\text{tot}} = \varphi^{\text{inc}} + \varphi^{\text{sca}}, \quad (10)$$

one can derive from Equation 4, with $g = 0$, the following relation

$$(-\Delta - k_0^2)(u^{\text{inc}} \varphi^{\text{sca}}) = \left(f + (\nabla \varphi^{\text{sca}})^2 \right) u^{\text{inc}}, \quad (11)$$

where $(\nabla \varphi^{\text{sca}})^2 = (\partial \varphi^{\text{sca}} / \partial x_1)^2 + (\partial \varphi^{\text{sca}} / \partial x_2)^2$. The details of this derivation can be found, for instance, in Kak and Slaney (2001, Sect. 6.2.2). Neglecting $(\nabla \varphi^{\text{sca}})^2$ in Equation 11, we obtain

$$(-\Delta - k_0^2)(u^{\text{inc}} \varphi^{\text{Rytov}}) = f u^{\text{inc}}, \quad (12)$$

where φ^{Rytov} is the *Rytov approximation* to φ^{sca} . Note that we still assume u^{inc} to be a monochromatic plane wave, as given in Equation 5. Thus, the product $u^{\text{inc}} \varphi^{\text{Rytov}}$ solves the same equation as u^{Born} . If we define the Rytov approximation to the scattered field, u^{Rytov} , in analogy to Equation 10 via

$$u^{\text{Rytov}} = e^{\varphi^{\text{Rytov}} + \varphi^{\text{inc}}} - u^{\text{inc}},$$

and replace φ^{Rytov} by $\frac{u^{\text{Born}}}{u^{\text{inc}}}$, we obtain a relation between the two approximate scattered fields that can be expressed as

$$u^{\text{Born}} = u^{\text{inc}} \log \left(\frac{u^{\text{Rytov}}}{u^{\text{inc}}} + 1 \right). \quad (13)$$

The relation between Born and Rytov in Equation 13 is not unique because of the multiple branches of the complex logarithm. In practical computations, this is addressed by a phase unwrapping as we will see in Equation 30.

Remark 1. There have been many investigations about the validity of the Born and Rytov approximations; see, e.g., Chen and Stamnes (1998), Slaney et al. (1984), or Kak and Slaney (2001, chap. 6). The Born approximation is reasonable only for a relatively (to the wavelength) small object. In particular, for a homogeneous cylinder of radius a , the Born approximation is valid if $a(n - n_0) < \lambda/4$, where λ is the wavelength of the incident wave and n is the constant refractive index inside the object. In contrast, the Rytov approximation only requires that $n - n_0 > (\nabla\varphi^{\text{sca}})^2/k_0^2$, i.e., the phase change of the scattered phase φ^{sca} , see Equation 10, is small over one wavelength, but it has no direct requirements on the object size and is therefore applicable for a larger class of objects. The latter is also observed in numerical simulations in Chen and Stamnes (1998). However, for objects that are small and have a low contrast $n - n_0$, the Born and Rytov approximation produce approximately the same results.

Modeling the Total Field Using Line and Point Sources

As an alternative to ideal incident plane waves, we consider models with one or several point sources. If arranged the right way, the resulting incident field can resemble a monochromatic plane wave. We refer to section “[Numerical Comparison of Forward Models](#)” for a numerical comparison of the different models presented here.

Point Source Far From Object

In this case, the right-hand side in Equation 4 is a Dirac delta function so that we obtain

$$\begin{cases} (-\Delta - k(\mathbf{x})^2) u_{\text{P}}^{\text{tot}}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0), \\ (-\Delta - k_0^2) u_{\text{P}}^{\text{inc}}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0). \end{cases} \quad (14)$$

If the position of the source is given by $\mathbf{x}_0 = -r_0\mathbf{e}_2$ with $r_0 > 0$ sufficiently large, then, after appropriate rescaling, $u_{\text{p}}^{\text{inc}}$ approximates a plane wave with wave number k_0 and propagation direction \mathbf{e}_2 in a neighborhood of $\mathbf{0}$.

Line Source

Alternatively, we let g be a sum of Dirac functions and consider

$$\left\{ \begin{array}{l} (-\Delta - k(\mathbf{x})^2) u_{\text{L}}^{\text{tot}}(\mathbf{x}) = \sum_{j=1}^{N_{\text{sim}}} \delta(\mathbf{x} - \mathbf{x}_j), \\ (-\Delta - k_0^2) u_{\text{L}}^{\text{inc}}(\mathbf{x}) = \sum_{j=1}^{N_{\text{sim}}} \delta(\mathbf{x} - \mathbf{x}_j), \end{array} \right. \quad (15)$$

where the number N_{sim} of simultaneous point sources should be sufficiently large. Moreover, the positions \mathbf{x}_j should be arranged uniformly along a line perpendicular to the propagation direction \mathbf{e}_2 of the plane wave. This is illustrated in section “[Modeling the Total Field Using Line and Point Sources](#)”.

Numerical Comparison of Forward Models

In this section, we numerically compare the forward models presented above. For the discretization of partial differential equations, several approaches exist, we mention, for instance, the finite differences that approximate the problem on a nodal grid (e.g., Virieux 1984), or methods that use the variational formulation, such as finite elements (Monk 2003) or discontinuous Galerkin methods (Hesthaven and Warburton 2007). In our work, we use the *hybridizable discontinuous Galerkin method* (HDG) for (HDG) the discretization and refer to Cockburn et al. (2009), Kirby et al. (2012), and Faucher and Scherzer (2020) for more details. The implementation precisely follows the steps prescribed in Faucher and Scherzer (2020), and it is carried out in the open-source parallel software `hawen`; see Faucher (2021) and Footnote 1. While the propagation is assumed on infinite space, the numerical simulations are performed on a finite discretization domain $\Omega \subset \mathbb{R}^2$, with absorbing boundary conditions (Engquist and Majda 1977) implemented to simulate free-space. It corresponds to the following Robin-type condition applied on the boundary Γ of the discretization domain Ω :

$$-ik(\mathbf{x})u(\mathbf{x}) + \partial_n u(\mathbf{x}) = 0, \quad \text{for } x \text{ on } \Gamma. \quad (16)$$

where $\partial_n u$ denotes the normal derivative of u . The test sample used below is a homogeneous medium encompassing a circular object of radius 4.5 around the origin with contrast $f = 1$. This corresponds to the characteristic function

$$\mathbf{1}_a^{\text{disk}}(\mathbf{x}) := \begin{cases} 1, & \mathbf{x} \in \mathcal{B}_a, \\ 0, & \mathbf{x} \in \mathbb{R}^2 \setminus \mathcal{B}_a, \end{cases} \quad (17)$$

of the disk \mathcal{B}_a with radius $a > 0$. The incident plane wave has wave number $k_0 = 2\pi$.

Modeling the Scattered Field Assuming Incident Plane Waves

We consider the solutions u^{sca} of Equation 6 and u^{Born} of Equation 7, both satisfying boundary condition Equation 16, and simulated following the HDG discretization indicated above. As an alternative for computing the Born approximation u^{Born} , we discretize the convolution Equation 8 with the Green's function G given in Equation 9. In particular, applying an $N \times N$ quadrature on the uniform grid $\mathcal{R}_N = \{-r_M, -r_M + 2r_M/N, \dots, r_M - 2r_M/N\}^2$ to Equation 8, we obtain

$$u^{\text{Born}}(\mathbf{x}) \approx u_{\text{conv},N}^{\text{Born}}(\mathbf{x}) := \left(\frac{2r_M}{N}\right)^2 \sum_{\mathbf{y} \in \mathcal{R}_N} G(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) e^{ik_0 y_2}, \quad \mathbf{x} \in \mathbb{R}^2. \quad (18)$$

In Fig. 2, we illustrate the solutions obtained with the different formulations. We observe that the transmission waves contain the most energy, that is, waves that “cross” the object. On the other hand, the solution has very low amplitude elsewhere, including the reflected waves. In Fig. 2d, we see that the Born approximation leads to an incorrect amplitude of the solution, in particular, the imaginary part. In addition, the imaginary part of $u_{\text{conv},200}^{\text{Born}}$ does not match the one of u^{Born} .

Modeling the Total Field Using Line and Point Sources

The objective is to evaluate how considering line and point sources differs from using incident plane waves and if the data obtained with both approaches are comparable. To compare the scattered fields obtained from Equations 14 and 15 with the solution u^{sca} of Equation 6, one needs to rescale according to

$$u_{\text{P}}^{\text{sca}} = \alpha_{\text{P}} \left(u_{\text{P}}^{\text{tot}} - u_{\text{P}}^{\text{inc}} \right),$$

$$u_{\text{L}}^{\text{sca}} = \alpha_{\text{L}} \left(u_{\text{L}}^{\text{tot}} - u_{\text{L}}^{\text{inc}} \right),$$

where α_{P} and α_{L} are constants depending only on the number and positions of the point sources \mathbf{x}_j . We illustrate in Fig. 3, where the line source is positioned at fixed height $x_2 = -15$ and composed of 441 points between $x_1 = -22$ and $x_1 = 22$. For the case of a point source, we have to consider a very wide domain, namely, $[-500, 500] \times [-500, 500]$, and the point source is positioned in

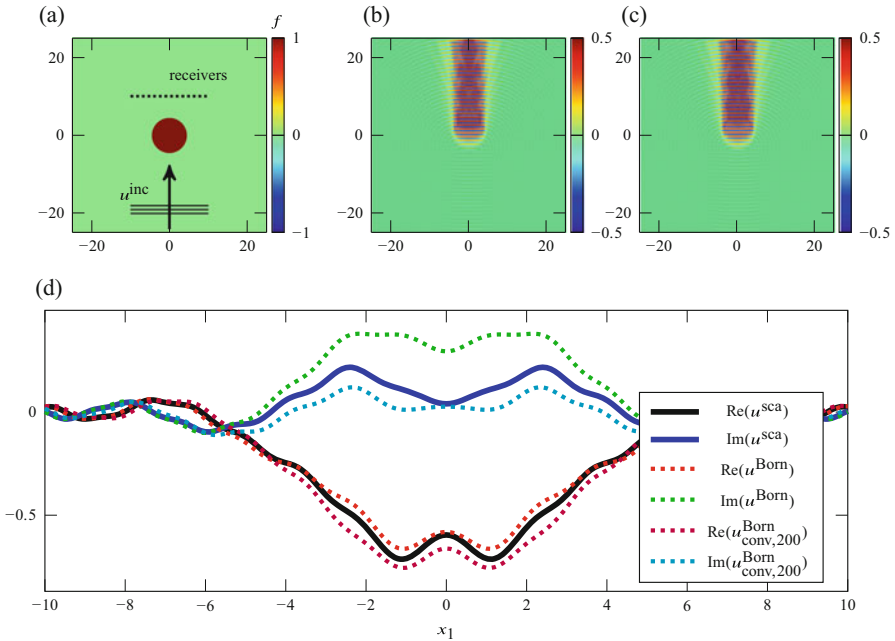


Fig. 2 Comparison of the scattered wave u^{sca} and the Born approximation u^{Born} . The computations are performed on a domain $[-25, 25] \times [-25, 25]$ with boundary conditions given in Equation 16. Further, we display $u^{Born}_{conv,200}$ which is the Born approximation obtained by the convolution Equation 18. (a) Perturbation model $f = \mathbf{1}_{4,5}^{disk}$. (b) Real part of u^{sca} . (c) Real part of u^{Born} . (d) Comparison of the solutions at fixed height $z = 10$

($x_1 = 0, x_2 = -480$). In Fig. 3g, we plot the corresponding solutions on a line at height $x_2 = 10$, i.e., for measurements of transmission waves.

We see that the simulation using the line source is very close to the original solution u^{sca} ; in fact, it is a more accurate representation than the Born approximation pictured in Fig. 2. The simulation using a point source positioned far away is also accurate, except for the middle area of the imaginary part. Furthermore, the major drawback of using a single point source is that it necessitates a very big domain, hence largely increasing the computational cost.

Fourier Diffraction Theorem

In this section, we discuss the inverse problem of recovering the scattering potential from measurements of the scattered wave under the Born or Rytov approximations. Before stating the fundamental result in this context, see Theorem 1 below, we have to introduce further notation.

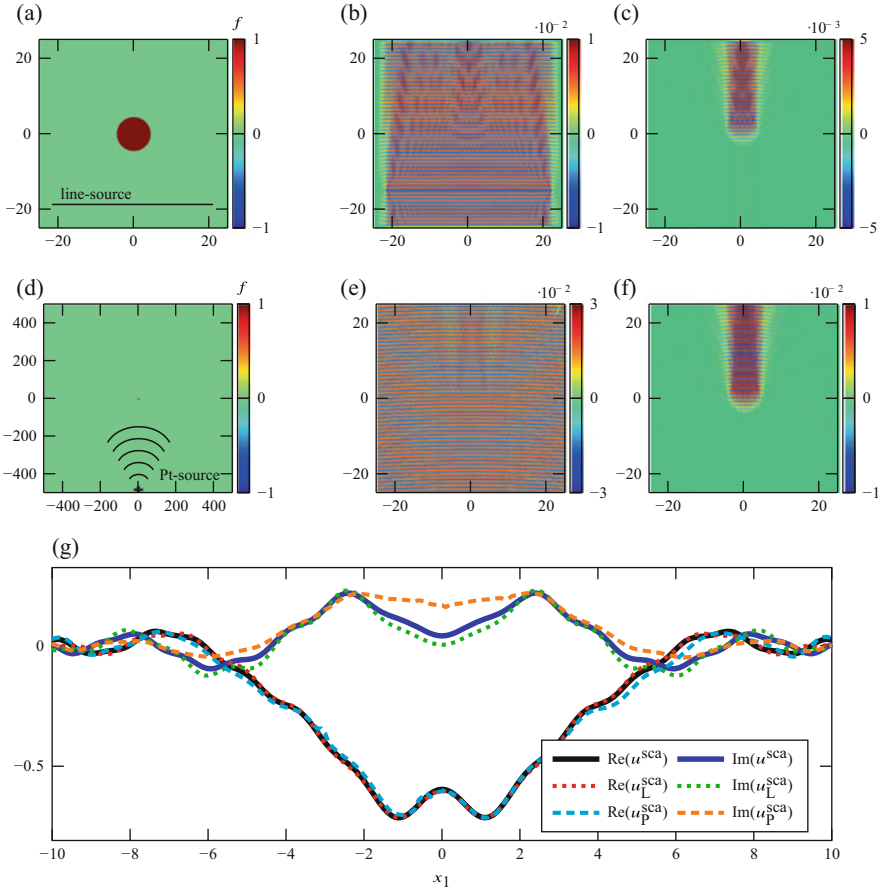


Fig. 3 Total field u_L^{tot} and scattered field u_L^{sca} (line source) and total field u_P^{tot} and scattered field u_P^{sca} (point source). The line source is composed of $N_{\text{sim}} = 441$ points at fixed height $x_2 = -15$. The computational domain for the point source is very large such that the perturbation is barely visible and the source is positioned in $\mathbf{x}_0 = (0, -480)^\top$. (a) Computational domain $[-25, 25]^2$ for line source with perturbation model $f = \mathbf{1}_{4.5}^{\text{disk}}$. (b) Real part of u_L^{tot} . (c) Real part of u_L^{sca} . (d) Computational domain $[-500, 500]^2$ for point source with perturbation model $f = \mathbf{1}_{4.5}^{\text{disk}}$. (e) Real part of u_P^{tot} near origin. (f) Real part of u_P^{sca} near origin. (g) Comparison of the solutions at fixed height $x_2 = 10$

We denote by \mathcal{F} the two-dimensional Fourier transform and by \mathcal{F}_1 the partial Fourier transform with respect to the first coordinate,

$$\mathcal{F}\phi(\mathbf{k}) = (2\pi)^{-1} \int_{\mathbb{R}^2} \phi(\mathbf{x})e^{-i\mathbf{x}\cdot\mathbf{k}} \, d\mathbf{x}, \quad \mathbf{k} \in \mathbb{R}^2,$$

$$\mathcal{F}_1\phi(k_1, x_2) = (2\pi)^{-\frac{1}{2}} \int_{\mathbb{R}} \phi(\mathbf{x})e^{-ik_1x_1} \, dx_1, \quad (k_1, x_2)^\top \in \mathbb{R}^2.$$

For $k_1 \in [-k_0, k_0]$, we define

$$\kappa(k_1) := \sqrt{k_0^2 - k_1^2}.$$

We can now formulate the Fourier diffraction theorem; see, for instance, Kak and Slaney (2001, Sect. 6.3), Natterer and Wübbeling (2001, Thm. 3.1), or Wolf (1969).

Theorem 1. *Let f be bounded with $\text{supp } f \subset \mathcal{B}_{r_M}$. For $k_1 \in (-k_0, k_0)$, we have*

$$\mathcal{F}_1 u^{\text{Born}}(k_1, r_M) = \sqrt{2\pi} \frac{ie^{i\kappa r_M}}{2\kappa} \mathcal{F}f(k_1, \kappa - k_0). \quad (19)$$

According to the Fourier diffraction theorem, Theorem 1, the measurements of the scattered wave u^{Born} can be related to the scattering potential f on a semicircle in k -space. Below we discuss how this so-called *k-space coverage* of the experiment is affected by (i) rotating the object and (ii) varying the wave number k_0 of the incident field u^{inc} .

Rotating the Object

Suppose the object rotates around the origin during the experiment. Then the resulting orientation-dependent scattering potential can be written as

$$f^\alpha(\mathbf{x}) = f(R_\alpha \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2,$$

where α ranges over a (continuous or discrete) set of angles $A \subset [0, 2\pi]$ and

$$R_\alpha = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

is a rotation matrix. If we let u^α be the Born approximation to the wave scattered by f^α , the collected measurements are given by

$$u^\alpha(x_1, r_M), \quad x_1 \in \mathbb{R}, \quad \alpha \in A.$$

Exploiting the rotation property of the Fourier transform, $\mathcal{F}(f \circ R_\alpha) = (\mathcal{F}f) \circ R_\alpha$, we obtain from Equation 19

$$\mathcal{F}_1 u^\alpha(k_1, r_M) = \sqrt{2\pi} \frac{ie^{i\kappa r_M}}{2\kappa} \mathcal{F}f \left(R_\alpha(k_1, \kappa - k_0)^\top \right).$$

Thus, the k -space coverage, that is, the set of all spatial frequencies $\mathbf{y} \in \mathbb{R}^2$ at which $\mathcal{F}f$ is accessible via the Fourier diffraction theorem, is given by

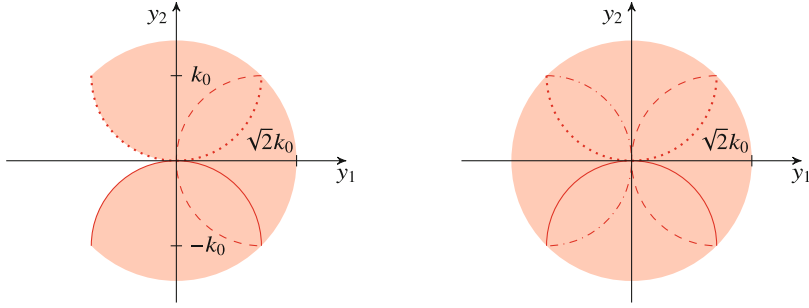


Fig. 4 k-space coverage for a rotating object. Left: half turn, $A = [0, \pi]$. Right: full turn, $A = [0, 2\pi]$. The k-space coverage (light red) is a union of infinitely many semicircles, each corresponding to a different orientation of the object. Some of the semicircles are depicted in red: solid arc ($\alpha = 0$), dashed ($\alpha = \pi/2$), dotted ($\alpha = \pi$), dash-dotted ($\alpha = 3\pi/2$)

$$\mathcal{Y} = \left\{ \mathbf{y} = R_{\alpha}(k_1, \kappa - k_0)^{\top} \in \mathbb{R}^2 : |k_1| < k_0, \alpha \in A \right\}.$$

It consists of rotated versions (around the origin) of the semicircle $(k_1, \kappa - k_0)^{\top}$, $|k_1| < k_0$, see Fig. 4.

Varying Wave Number

Now suppose the object is illuminated or insonified by plane waves with wave numbers ranging over a set $K \subset (0, +\infty)$. Recall the definition of the scattering potential $f_{k_0} = k_0^2(n^2/n_0^2 - 1)$ from Equation 2, but note that we have now added a subscript to indicate the dependence of f on k_0 . If the variation of the object's refractive index n with $k_0 \in K$ is negligible, we can write

$$f_{k_0}(\mathbf{x}) = k_0^2 f_1(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2. \quad (20)$$

If no confusion arises, we may write $f = f_1$. Denoting by u_{k_0} the Born approximation to the wave scattered by f_{k_0} , the resulting collection of measurements is

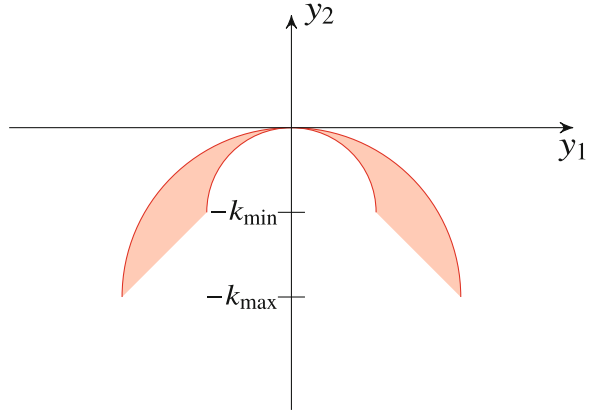
$$u_{k_0}(x_1, r_M), \quad x_1 \in \mathbb{R}, \quad k_0 \in K.$$

Then, according to Equation 19, we have

$$\mathcal{F}_1 u_{k_0}(k_1, r_M) = \sqrt{2\pi} \frac{\mathbf{i}e^{\mathbf{i}k r_M}}{2\kappa} k_0^2 \mathcal{F} f_1(k_1, \kappa - k_0).$$

Notice that now κ also varies with k_0 . The resulting k-space coverage

Fig. 5 k-space coverage for k_0 covering the interval $[k_{\min}, k_{\max}]$



$$\mathcal{Y} = \left\{ \mathbf{y} = (k_1, \kappa - k_0)^\top \in \mathbb{R}^2 : k_0 \in K, |k_1| < k_0 \right\}$$

is a union of semicircles scaled and shifted (in direction of $-\mathbf{e}_2$) according to $k_0 \in K$.

Consider, for example, $K = [k_{\min}, k_{\max}]$. Then, the k-space coverage consists of all points $(y_1, y_2)^\top \in \mathbb{R}^2$ such that $|y_1| \leq k_{\max}$ and

$$\sqrt{k_{\max}^2 - y_1^2} - k_{\max} \geq y_2 \geq \begin{cases} -|y_1|, & |y_1| \geq k_{\min}, \\ \sqrt{k_{\min}^2 - y_1^2} - k_{\min}, & \text{otherwise,} \end{cases}$$

see Fig. 5. Note that, in contrast to the scenarios of section “Rotating the Object,” there are large missing parts near the origin.

Rotating the Object with Multiple Wave Numbers

We combine the two previous observations by picking a finite set of wave numbers $K \subset (0, \infty)$ and performing a full rotation of the object for each $k_0 \in K$. Let $u_{k_0}^\alpha$ be the Born approximation to the wave scattered by $f_{k_0}^\alpha = k_0^2 f_1 \circ R_\alpha$. Then the full set of measurements is given by

$$u_{k_0}^\alpha(x_1, r_M), \quad x_1 \in \mathbb{R}, \quad \alpha \in [0, 2\pi], \quad k_0 \in K, \tag{21}$$

and the Fourier diffraction theorem yields

$$\mathcal{F}_1 u_{k_0}^\alpha(k_1, r_M) = \sqrt{2\pi} \frac{ie^{i\kappa r_M}}{2\kappa} k_0^2 \mathcal{F} f_1 \left(R_\alpha(k_1, \kappa - k_0)^\top \right). \tag{22}$$

We deduce from sections “Rotating the Object” and “Varying Wave Number” that the resulting k-space coverage \mathcal{Y} is the union of disks with radii $\sqrt{2}k_0$, all centered at the origin. Hence, \mathcal{Y} is just the largest disk, that is, the one corresponding to the largest wave number $\max K$. However, smaller disks in k-space are covered more often, which might improve reliability of the reconstruction for noisy data.

Reconstruction Methods

In the following, we assume data generated by line sources according to the setup described in section “Modeling the Total Field Using Line and Point Sources”. We simulate the total fields solutions to Equation 15, which are the synthetic data used for the reconstruction.

Reconstruction Using Full Waveform Inversion

For the identification of the physical properties of the medium, the *Full Waveform Inversion* (FWI) relies on an iterative minimization of a misfit functional which evaluates a distance between numerical simulation and measurements of the total field. The Full Waveform Inversion method arises in the context of seismic inversion for sub-surface Earth imaging, cf. Bamberger et al. (1979), Lailly (1983), Tarantola (1984), Pratt et al. (1998), and Virieux and Operto (2009), where the measured seismograms are compared to simulated waves.

With FWI, we invert with respect to the wave speed c , from which the wave number is defined according to Equation 1. It further connects with the model perturbation f according to Equation 3. In our experiment, c_0 is used as an initial guess (i.e., we start from constant background), and then c is inverted rather than f , as discussed in Remark 2. Given some measurements \mathbf{d} of the total field, the quantitative reconstruction of the wave speed c is performed following the minimization of the misfit functional \mathcal{J} such that

$$\min_c \mathcal{J}(c), \quad \mathcal{J} = \text{dist}(\mathcal{R}u^{\text{tot}}, \mathbf{d}), \quad \text{where } u \text{ solves Equation 15.} \quad (23)$$

Here, $\text{dist}(\cdot)$ is a distance function to evaluate the difference between the measurements and the simulations, and \mathcal{R} is a linear operator to restrict the solution to the positions of the receivers. For simplicity, we do not encode a regularization term in Equation 23 and refer the readers to, e.g., Faucher et al. (2020c), Kaltenbacher (2018), and the references therein.

Several formulations of the distance function have been studied for FWI (in particular, for seismic applications), such as a logarithmic criterion, Shin et al. (2007), the use of the signal’s phase or amplitude, Bednar et al. (2007) and Pyun et al. (2007), the use of the envelope of the signal, Fichtner et al. (2008), criteria

based upon cross-correlation, Luo and Schuster (1991), Van Leeuwen and Mulder (2010), Faucher et al. (2020a), and Faucher et al. (2021), or optimal transport distance, Métivier et al. (2016). Here, we rely on a least-squares approach where the misfit functional is defined as the L^2 distance between the data and simulations:

$$\mathcal{J}(c) := \frac{1}{2} \sum_{\omega \in c_0 K} \sum_{\alpha \in A} \|\mathcal{R}u^{\text{tot}}(c, \omega, \alpha) - \mathbf{d}(\omega, \alpha)\|_{L^2(-l_M, l_M)}^2, \quad (24)$$

where $\mathbf{d}(\omega, \alpha)$ refers to the measurement data of the total field at the measurement plane with respect to the object rotated with angle α , and $u^{\text{tot}}(c, \omega, \alpha)$ is the solution of Equation 15 with $k(\mathbf{x}) = \omega/c(R_\alpha^\top \mathbf{x})$. The last term $R_\alpha^\top \mathbf{x}$ encodes the rotation of the object. We note that a rotation of the object is equivalent of the rotation of both the measurement line and the direction of the incident field. We have encoded a sum over the frequencies ω , which are chosen in accordance with the frequency content available in measurements. In the computational experiments, we further investigate uni- and multifrequency reconstructions.

The minimization of the misfit functional Equation 20 follows an iterative Newton-type method as depicted in Algorithm 1. Due to the computational cost, we use first-order information and avoid the Hessian computation (Virieux and Operto 2009): namely, we rely on the nonlinear conjugate gradient method for the model update, cf. Nocedal and Wright (2006) and Faucher (2017). Furthermore, to avoid the formation of the dense Jacobian, the gradient of the misfit functional

Algorithm 1 Iterative reconstruction of the wave speed model following the minimization of the misfit functional. At each iteration, the total field solution to Equation 15 is computed, and the gradient of the misfit functional is used to update the wave speed model. The algorithm stops when the prescribed number of iterations is performed for all of the frequencies of interest.

Input: Initial wave speed model c_0 .

Initiate global iteration number $\ell := 1$;

for frequency $\omega \in c_0 K$ **do**

for iteration $j = 1, \dots, n_{iter}$ **do**

 Compute the solution to the wave equation using current wave speed model c_ℓ and frequency ω , that is, the solution to Equation 15 with $k(\mathbf{x}) = \omega/c_\ell(\mathbf{x})$;

 Evaluate the misfit functional \mathcal{J} in Equation 24;

 Compute the gradient of the misfit functional using the adjoint-state method;

 Update the wave speed model using nonlinear conjugate gradient method to obtain $c_{\ell+1}$;

 Update global iteration number $\ell \leftarrow \ell + 1$;

end

end

Output: Approximate wave speed c , from which the scattering potential f can be computed via Equations 2 and 1.

is computed using the adjoint-state method, cf. Pratt et al. (1998), Plessix (2006), Barucq et al. (2019), and Faucher and Scherzer (2020). In Algorithm 1, we further implement a progression in the frequency content, which is common to mitigate the ill-posedness of the nonlinear inverse problem, Bunks et al. (1995). We further invert each frequency independently, from low to high, as advocated by Barucq et al. (2019) and Faucher et al. (2020b). For the implementation details using the HDG discretization, we refer to Faucher and Scherzer (2020).

Remark 2. In the computational experiments, the reconstruction with FWI assumes the availability of the total fields which are solutions to Equation 15, and we invert with respect to the (frequency independent) wave speed c defined in Equation 3. We could instead use the representation with relation $k^2 = k_0^2 + f$ and invert with respect to the perturbation f , imposing the (known) smooth background c_0 . Inverting with respect to c rather than f is mainly motivated by consistency with existing literature in FWI (Virieux and Operto 2009), in which the background model (c_0) is usually unknown. Nonetheless, reformulating the minimization with respect to f and imposing c_0 could improve the efficiency of FWI, as advocated by the data-space reflectivity inversion of Clément et al. (2001) and Faucher et al. (2020b).

Reconstruction Based on the Born and Rytov Approximations

In this section, we present numerical methods for the computation of the Born and Rytov approximations from Equations 7 and 12, respectively, as well as the reconstruction of the scattering potential. We concentrate on the case of full rotations of the object using incident waves with different wave numbers $k_0 \in K$; see section “Rotating the Object with Multiple Wave Numbers”. The tomographic reconstruction is based on the Fourier diffraction theorem, Theorem 1, and the nonuniform discrete Fourier transform. Nonuniform Fourier methods have also been applied in computerized tomography (Potts and Steidl 2001), magnetic resonance imaging (Knopp et al. 2007), spherical tomography (Hielscher and Quellmalz 2015, 2016), or surface wave tomography (Hielscher et al. 2018).

In the following, we describe the discretization steps we apply. For $N \in 2\mathbb{N}$, let

$$\mathcal{I}_N := \left\{ -\frac{N}{2} + j : j = 0, \dots, N-1 \right\}.$$

We sample the scattering potential f on the uniform grid $\mathcal{R}_N := \frac{2r_s}{N} \mathcal{I}_N^2$ in the square $[-r_s, r_s]^2$ for some $r_s > 0$. We assume that we are given measurements of the Born approximation

$$u_{k_0}^\alpha(x_1, r_M), \quad x_1 \in [-l_M, l_M],$$

for $\alpha \in A \subset [0, 2\pi]$ and $k_0 \in K$, cf. Equation 21. We want to reconstruct the scattering potential $f = f_1$; recall Equation 20, utilizing Equation 22. We adapt the reconstruction approach of Kirisits et al. (2021), which is written for the 3D case. First, we need to approximate the partial Fourier transform

$$\mathcal{F}_1 u_{k_0}^\alpha(k_1, r_M) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} u_{k_0}^\alpha(x_1, r_M) e^{-ix_1 k_1} dx_1, \quad k_1 \in [-k_0, k_0]. \quad (25)$$

The *discrete Fourier transform* (DFT) of $u(\cdot, r_M)$ on m equispaced points $x_1 \in (2l_M/m)\mathcal{I}_m$ can be defined by

$$\mathbf{F}_{1,m} u(k_1, r_M) := \frac{1}{\sqrt{2\pi}} \frac{2l_M}{m} \sum_{x_1 \in \frac{2l_M}{m} \mathcal{I}_m} u(x_1, r_M) e^{-ix_1 k_1}, \quad k_1 \in \frac{\pi}{l_M} \mathcal{I}_m, \quad (26)$$

which gives an approximation of Equation 25. Then, Equation 22 yields

$$k_0^2 \mathcal{F} f (R_\alpha(k_1, \kappa - k_0)^\top) = -i \sqrt{\frac{2}{\pi}} \kappa e^{-i\kappa r_M} \mathcal{F}_1 u_{k_0}^\alpha(k_1, r_M) \quad (27)$$

for $|k_1| \leq k_0$. Considering that we sample the angle α on the equispaced, discrete grid $A = (2\pi/n_A)\{0, 1, \dots, n_A - 1\}$ and some finite set $K \subset (0, \infty)$, Equation 26 provides an approximation of $\mathcal{F}f$ on the non-uniform grid

$$\mathcal{Y}_{m,n_A} := \left\{ R_\alpha(k_1, \kappa - k_0)^\top : \right. \\ \left. k_1 \in \frac{\pi}{l_M} \mathcal{I}_m, |k_1| \leq k_0, \alpha \in \frac{2\pi}{n_A} \{0, 1, \dots, n_A - 1\}, k_0 \in K \right\}$$

in k -space, from which we want to reconstruct the scattering potential f .

Let M be the cardinality of \mathcal{Y}_{m,n_A} . The two-dimensional *nonuniform discrete Fourier transform* (NDFT) is the linear operator $\mathbf{F}_N: \mathbb{R}^{N^2} \rightarrow \mathbb{R}^M$ defined for the vector $\mathbf{f}_N := (f(\mathbf{x}))_{\mathbf{x} \in \mathcal{R}_N}$ elementwise by

$$\mathbf{F}_N \mathbf{f}_N(\mathbf{y}) := \frac{1}{2\pi} \frac{(2r_s)^2}{N^2} \sum_{\mathbf{x} \in \mathcal{R}_N} f(\mathbf{x}) e^{-i\mathbf{x} \cdot \mathbf{y}}, \quad \mathbf{y} \in \mathcal{Y}_{m,n_A}, \quad (28)$$

see Plonka et al. (2018, Section 7.1). It provides an approximation of the Fourier transform

$$\mathcal{F}f(\mathbf{y}) \approx \mathbf{F}_N \mathbf{f}_N(\mathbf{y}), \quad \mathbf{y} \in \mathcal{Y}_{m,n_A}. \quad (29)$$

solving an equation $\mathbf{F}_N \mathbf{f}_N(\mathbf{y}) = b$ for \mathbf{f}_N amounts to applying an *inverse NDFT*, which usually utilizes an iterative method such as the conjugate gradient method on the normal equations (CGNE); see Kunis and Potts (2007) and Plonka et al. (2018, Section 7.6). One should be aware that the notation regarding conjugate gradient algorithms varies in the literature: the algorithm called CGNE in Hanke (1995) is known as CGNR in Kunis and Potts (2007). Conversely, the algorithm CGME in Hanke (1995) is known as CGNE in Kunis and Potts (2007).

In conclusion, our method for computing f given the Born approximation u^{Born} is summarized in Algorithm 2.

Algorithm 2 Iterative reconstruction of the scattering potential f based on the Born approximation using an inverse NDFT

Input: Measurement data

$$u_{k_0}^\alpha(x_1, r_M), \quad x_1 \in \frac{2l_M}{m} \mathcal{I}_m, \quad \alpha \in A = \frac{2\pi}{n_A} \{0, \dots, n_A - 1\}, \quad k_0 \in K.$$

for $k_0 \in K$ **do**

for $\alpha \in A$ **do**

 Compute $-i\sqrt{\frac{2}{\pi}} \kappa e^{-i\kappa r_M} \mathbf{F}_{1,m} u_{k_0}^\alpha(k_1, r_M)$, $k_1 \in \frac{\pi}{l_M} \mathcal{I}_m$, with a DFT in Equation 26;

end

end

Solve Equation 27 with Equation 29 for \mathbf{f}_N using the conjugate gradient method;

Output: Approximate scattering potential $\mathbf{f}_N \approx (f(\mathbf{x}))_{\mathbf{x} \in \mathcal{R}_N}$.

The Rytov approximation u^{Rytov} , see Equation 12, is closely related to the Born approximation, but it has a different physical interpretation. Assuming that the measurements arise from the Rytov approximation, we apply Equation 13 to obtain u^{Born} from which we can proceed to recover f as shown above. We note that the actual implementation of Equation 13 requires a phase unwrapping because the complex logarithm is unique only up to adding $2\pi i$, cf. Müller et al. (2015). In particular, we use in the two-dimensional case

$$u^{\text{Born}} = u^{\text{inc}} \left(i \text{unwrap} \left(\arg \left(\frac{u^{\text{Rytov}}}{u^{\text{inc}}} + 1 \right) \right) + \ln \left| \frac{u^{\text{Rytov}}}{u^{\text{inc}}} + 1 \right| \right), \quad (30)$$

where \arg denotes the principle argument of a complex number and unwrap denotes a standard unwrapping algorithm. For the reconstruction with the Rytov approximation, we can use Algorithm 2 as well, but we have to preprocess the data u by Equation 30.

Numerical Experiments

In this section, we carry out numerical experiments comparing the reconstruction obtained with FWI (section “[Reconstruction Using Full Waveform Inversion](#)”) and Born and Rytov approximations (section “[Reconstruction Based on the Born and Rytov Approximations](#)”), using single and multi-frequency datasets. We consider different media with varying shapes and amplitude for the embedded objects. Our experiments use synthetic data with added noise: Firstly, synthetic simulations are carried out for the known wave speeds using the software `hawen` (Faucher 2021). The discretization relies on a fine mesh (usually a few hundred thousands cells in the discretized domain) and polynomials of order 5 to ensure accuracy. Then, white Gaussian noise is incorporated in the synthetic data, with a signal-to-noise ratio of 50 dB. The reconstruction with FWI also relies on software `hawen`, but uses different discretization setups to foster the computational time: the discretization mesh is coarser (usually a few tens of thousand cells) and the polynomial order varies with the cells, depending on the (local to the cell) wavelength, in order to remain as small as possible, as detailed in Faucher and Scherzer (2020). The computational cost of FWI is further discussed in section “[Computational Costs](#)”.

We perform a full rotation of the object for a single or for multiple frequencies ω and thus wave numbers k_0 , cf. section “[Rotating the Object with Multiple Wave Numbers](#)”. For different frequencies, the scattering potential is scaled according to Equation 20. We always reconstruct the rescaled scattering potential f_1 , which we will simply denote by f in the following. In all numerical experiments, we rely on forward data generated with the forward model of line sources in section “[Line Source](#)”.

We compare the reconstruction quality based on the *peak signal-to-noise ratio* (PSNR) of the reconstruction \mathbf{g} with respect to the ground truth \mathbf{f} determined by

$$\text{PSNR}(\mathbf{f}, \mathbf{g}) := 10 \log_{10} \frac{\max_{\mathbf{x} \in \mathcal{R}_N} |\mathbf{f}(\mathbf{x})|^2}{N^{-2} \sum_{\mathbf{x} \in \mathcal{R}_N} |\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{x})|^2},$$

where higher values indicate a better reconstruction quality.

Reconstruction of Circular Contrast with Various Amplitudes and Sizes

For the initial reconstruction experiments, we consider a circular object in a homogeneous background, namely, the scattering potential f of Equation 17. We investigate different sizes and contrasts for the object, as shown in Fig. 6. The data are generated for $n_A = 40$ angles of incidence equally partitioned between 0° to 351° , every 9° , and the measurement line is sampled on the 200 point uniform grid $10^{-1} \mathcal{I}_{200} \subset [-l_M, l_M]$ with $l_M = 10$. Let us note that in this context of a circularly

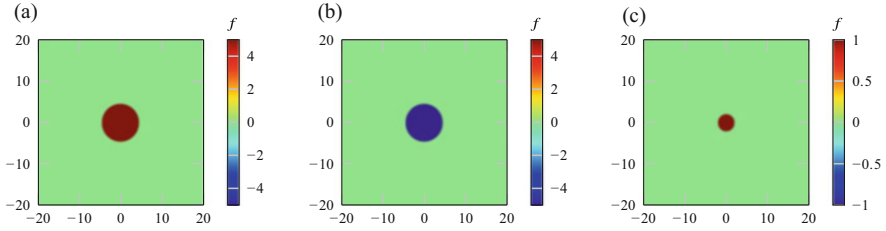


Fig. 6 Different perturbation models f used for the computational experiments, given for frequency $\omega/(2\pi) = 1$, with the relation to the wave speed given in Equation 3. Both the size and contrast vary: we consider two radii (4.5 and 2) and three contrasts (1, 5, and -5 with corresponding wave speeds $c = 0.9876$, $c = 0.9421$, and $c = 1.0701$, respectively), for a total of six configurations. The computations are carried out on the domain $[-50, 50] \times [-50, 50]$, i.e., a slightly larger setup than Fig. 3, and we only picture the area near the origin for clearer visualization. (a) Perturbation f for radius 4.5 and amplitude 5: model $f = 5 \cdot \mathbf{1}_{4.5}^{\text{disk}}$. (b) Perturbation f for radius 4.5 and amplitude -5 : model $f = -5 \cdot \mathbf{1}_{4.5}^{\text{disk}}$. (c) Perturbation f for radius 2 and amplitude 1: model $f = \mathbf{1}_2^{\text{disk}}$

symmetric object, the data of each angle are similar and correspond to that of Fig. 3 for $f = \mathbf{1}_{4.5}^{\text{disk}}$.

Reconstruction Using FWI with Single-Frequency Datasets

We first only use data at frequency $\omega/(2\pi) = 1$, that is, wave number $k_0 = 2\pi$ for the reconstruction of the different perturbations illustrated in Fig. 6. With the background $k_0 = 2\pi$ (i.e., wave speed of 1), it means that we only rely on waves with wavelength 1 when propagating in the (homogeneous) background. Then all measurements in \mathbf{x} are in multiples of the wavelength. In the case of a single frequency, only the inner loop remains in Algorithm 1, and we perform 50 iterations. In Fig. 7, we picture the reconstruction obtained for the six different perturbations f . We observe that the reconstructions of the smaller object of radius 2 (Fig. 7a, b and c) are more accurate, both in terms of the circular shape and in terms of amplitude. In the case of the larger object, the mild amplitude (Fig. 7d) is accurately recovered, while the stronger contrasts (Fig. 7e and f) are only partially retrieved. Here, the outer part of the disk appears, but the amplitude is incorrect with a ring effect and incorrect values in the inner area. Therefore, the reconstruction using single-frequency data is limited and its success depends on two factors: the size of the object and its contrast.

Reconstruction Using FWI with Multiple Frequency Datasets

The difficulty of recovering a large object with a strong contrast can be mitigated by the use of multi-frequency datasets, allowing a multiscale reconstruction (Bunks et al. 1995; Faucher et al. 2020b). We carry out the iterative reconstruction using increasing frequencies, starting with $\omega/(2\pi) = 0.2$ and up to $\omega/(2\pi) = 1$. Following Faucher et al. (2020b), we use a sequential progression, that is, every

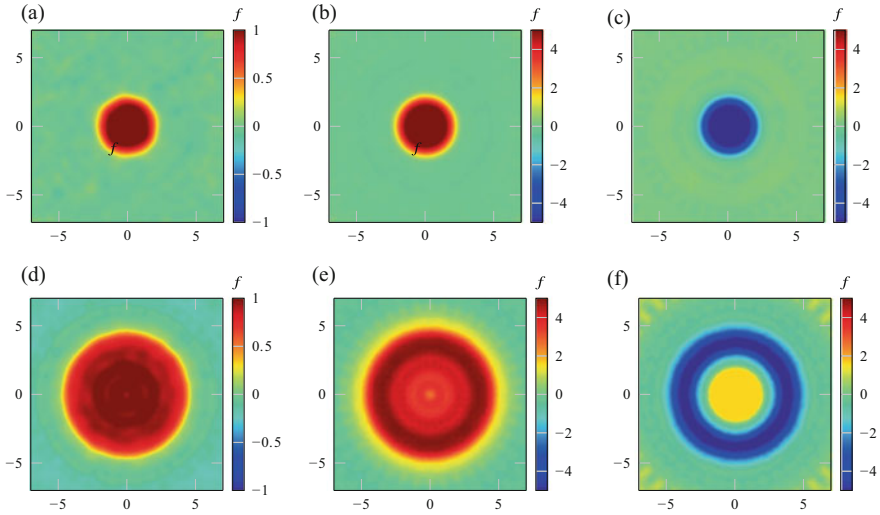


Fig. 7 Reconstruction using iterative minimization using data of frequency $\omega/(2\pi) = 1$ only. In each cases, 50 iterations are performed and the initial model consists in a constant background where $k_0 = 2\pi$. The data consist of $n_A = 40$ different angles of incidence from 0° to 351° (a) Reconstruction for model $f = \mathbf{1}_2^{\text{disk}}$ (PSNR 23.50). (b) Reconstruction for model $f = 5 \cdot \mathbf{1}_2^{\text{disk}}$ (PSNR 24.43). (c) Reconstruction for model $f = -5 \cdot \mathbf{1}_2^{\text{disk}}$ (PSNR 24.25). (d) Reconstruction for model $f = \mathbf{1}_{4,5}^{\text{disk}}$ (PSNR 14.79). (e) Reconstruction for model $f = 5 \cdot \mathbf{1}_{4,5}^{\text{disk}}$ (PSNR 15.71). (f) Reconstruction for model $f = -5 \cdot \mathbf{1}_{4,5}^{\text{disk}}$ (PSNR 10)

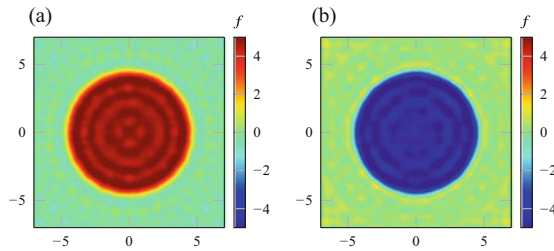


Fig. 8 Reconstruction using multi-frequency data from $\omega/(2\pi) = 0.2$ to $\omega/(2\pi) = 1$. The initial model consists in a constant wave speed $c_0 = 1$. The data consist of $n_A = 40$ different angles of incidence from 0° to 351° (a) Reconstruction for model $f = 5 \cdot \mathbf{1}_{4,5}^{\text{disk}}$ (PSNR 19.43). (b) Reconstruction for model $f = -5 \cdot \mathbf{1}_{4,5}^{\text{disk}}$ (PSNR 19.02)

frequency is inverted separately. The reconstructions for the object of radius 4.5 and contrast $f = \pm 5$ are pictured in Fig. 8. Contrary to the case of a single frequency (see Fig. 7d), the reconstruction is now accurate and stable: the amplitude is accurately retrieved and the circular shape is clear, avoiding the circular artifacts observed in Fig. 7e and f.

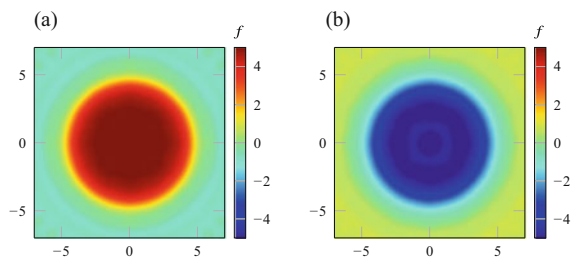


Fig. 9 Reconstruction using frequency $\omega/(2\pi) = 0.7$. The initial model consists in a constant wave speed $c_0 = 1$. The data consist of $n_A = 40$ different angles of incidence from 0° to 351° . **(a)** Reconstruction for model $f = 5 \cdot \mathbf{1}_{4.5}^{\text{disk}}$ (PSNR 15.33). **(b)** Reconstruction for model $f = -5 \cdot \mathbf{1}_{4.5}^{\text{disk}}$ (PSNR 15.03)

Remark 3. It is possible to recover the model with a single frequency, which needs to be carefully chosen depending on the size of the object and the amplitude of the contrast. We have seen in Fig. 7 that the frequency $\omega/(2\pi) = 1$ is sufficient for the object of radius 2, but for the radius 4.5, we need a lower frequency (i.e., larger wavelength) to uncover the larger object. We illustrate in Fig. 9 the reconstruction using data at only $\omega/(2\pi) = 0.7$, where we see that the shape and contrast are retrieved accurately. Nonetheless, it is hard to predict this frequency a priori, and we believe it remains more natural to use multiple frequencies (when available in the data), to ensure the robustness of the algorithm.

Reconstruction Using Born and Rytov Approximations

For the reconstruction with Algorithm 2, which relies on the Born or Rytov approximation, we use the same data as in the above experiment. We use a grid with $N = 240$ and $r_s = N/(8\sqrt{2}) \approx 10$. The numerical results indicate that r_s should not be smaller than l_M . Since we have $k_1^2 \leq k_0^2$ and the distance between two grid points of k_1 is π/l_M , only around $2k_0 l_M/\pi \approx 40$ of them contribute to the data of the inverse NDFT.

In the following reconstructions, we use a fixed number of 20 iteration steps in the conjugate gradient method. Initially, we use the frequency $\omega/(2\pi) = 1$ of the incident wave; therefore, $k_0 = 2\pi$. Reconstructions of the circular model $f = \mathbf{1}_a^{\text{disk}}$ are shown in Fig. 10. We note that all reconstructions are reasonably good, where the Rytov reconstruction looks slightly better inside the object.

For a higher amplitude of the model function f , the limitations of the linear models become apparent. In Fig. 11, we see that the Born reconstruction of the larger object fails, and for the smaller object, only the Rytov approximation yields a good reconstruction, which is consistent with Remark 1. With the Born approximation, we recognize the object's shape but not its amplitude, which is consistent with the observations in Müller et al. (2016). However, as we see in Fig. 7, even the FWI reconstruction makes a considerable error in the object's interior, and we cannot expect the linear models to be better.

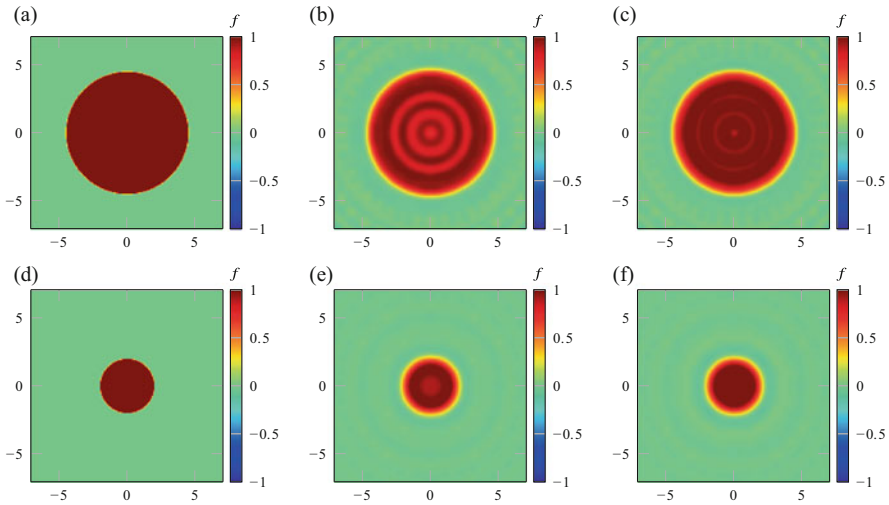


Fig. 10 Reconstructions with the Born and Rytov approximation, where the data $u(\cdot, r_M)$ is generated with the line source model. The incident field has the frequency $\omega/(2\pi) = 1$. Visible is only the cut out center, where we compute the PSNR. (a) Model $f = \mathbf{1}_{4,5}^{\text{disk}}$. (b) Born reconstruction (PSNR 19.35). (c) Rytov reconstruction (PSNR 19.28). (d) Model $f = \mathbf{1}_2^{\text{disk}}$. (e) Born reconstruction (PSNR 24.13). (f) Rytov reconstruction (PSNR 24.01)

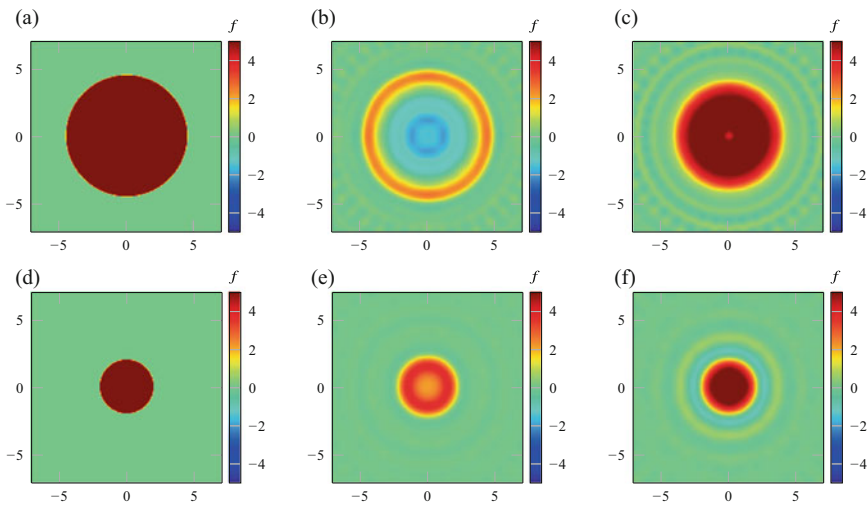
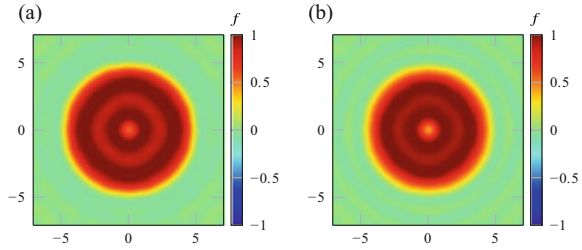


Fig. 11 Same setting as in Fig. 10, but with a higher amplitude of 5 (a) Model $f = 5 \cdot \mathbf{1}_{4,5}^{\text{disk}}$. (b) Born reconstruction (PSNR 4.68). (c) Rytov reconstruction (PSNR 11.91). (d) Model $f = 5 \cdot \mathbf{1}_2^{\text{disk}}$. (e) Born reconstruction (PSNR 19.39). (f) Rytov reconstruction (PSNR 21.31)

Fig. 12 Reconstructions of $\mathbf{1}_{4,5}^{\text{disk}}$, where the incident field has the frequency $\omega/(2\pi) = 0.7$ instead of 1. The rest of the setting is from Fig. 10. (a) Born reconstruction (PSNR 18.05). (b) Rytov reconstruction (PSNR 16.52)



We see that the FWI and the Born/Rytov reconstructions contain different kinds of artifacts. Therefore, a comparison of the visual image quality perception does not necessarily yield the same conclusions as for the computed PSNR values. Furthermore, the size of the object has a considerable effect on the PSNR, e.g., the images in Fig. 11c and f show a comparable visual quality, but the latter’s PSNR is considerably better because of the lower error in the background farther away from the object; see also Huynh-Thu and Ghanbari (2010) for a study on the PSNR.

In Fig. 12, we use the same setup as before, but with the frequency $\omega/(2\pi) = 0.7$ instead of $\omega/(2\pi) = 1$ and thus the wave number $k_0 = \omega$. Apparently, the reconstruction becomes worse with lower frequency, because it provides a smaller k -space coverage.

Reconstruction of Embedded Shapes: Phantom 1

We consider a more challenging scenario with shapes embedded in the background medium. In Fig. 13a, we picture the perturbation f consisting of a disk and heart included in an ellipse, with f varying from 0 to 0.5. The computational domain corresponds to $[-20, 20] \times [-20, 20]$, with line-sources positioned at a distance $R = 10$ and receivers in $r_M = 6$ to capture the data. The data are generated using $n_A = 100$ incidence angles α equispaced on $[0, 2\pi]$, following the steps described in section “Modeling the Total Field Using Line and Point Sources”. This is illustrated in Fig. 13.

Reconstruction Using FWI

We carry out the reconstruction following Algorithm 1, and the results are pictured in Fig. 14, where we compare the use of single and multi-frequency data. In this example, we see that with relatively low-frequency data (i.e., relatively large wavelength), such as for frequency $\omega/(2\pi) = 0.7$ and $\omega/(2\pi) = 1$, the reconstruction is smooth; see Fig. 14a and b, and one needs to use smaller wavelengths to obtain a better reconstruction; see Fig. 14c and d. In Fig. 14e, we see that multi-frequency data gives the best reconstruction, it is also the most robust as one does not need to anticipate the appropriate wavelength before carrying out the reconstruction. Here both the shapes and contrast in amplitude are accurately obtained. We notice some

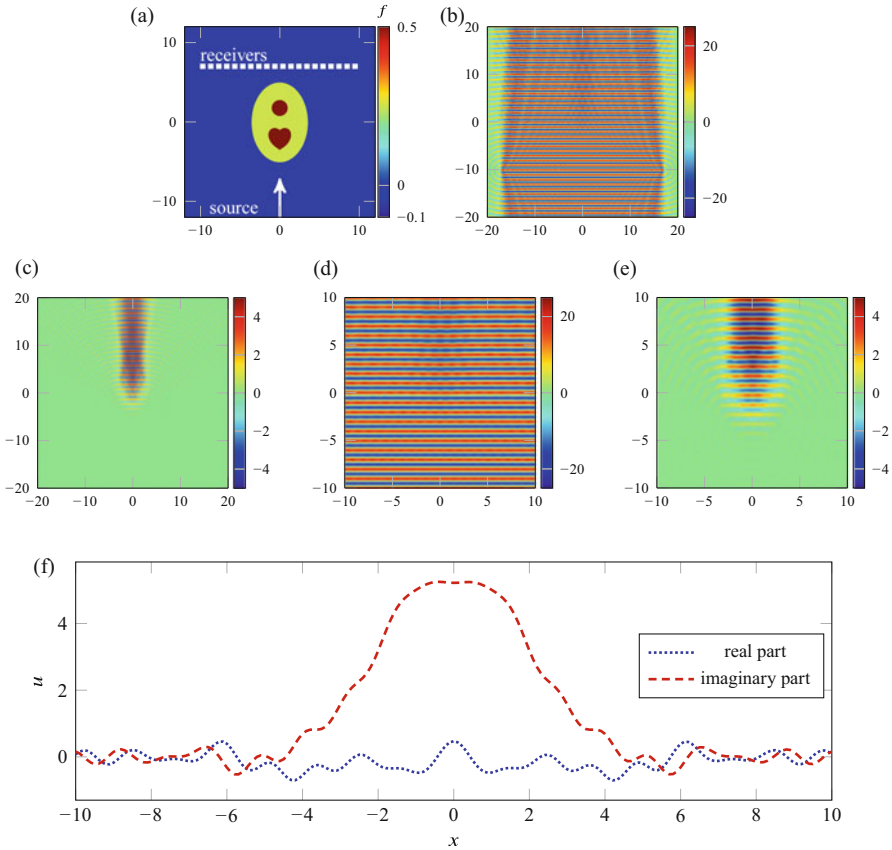


Fig. 13 Illustration of the acquisition setup and generated data. The computations are carried out on the domain $[-20, 20] \times [-20, 20]$. While FWI uses the total field, the reconstruction based upon Born and Rytov approximations use the scattered solutions, obtained after removing a reference solution corresponding to a propagation in an homogeneous medium, cf. section “Forward Models” (a) Perturbation model at frequency $\omega/(2\pi) = 1$, the wave speed is equal to 1 in the background. The positions of the source and the receivers recording transmission data are pictured in white. (b) Real part of the global solution to Equation 15 at frequency $\omega/(2\pi) = 1$, the source is discretized by $N_{\text{sim}} = 1361$ simultaneous excitations at fixed height $x_2 = -10$. (c) Real part of the scattering solution at frequency $\omega/(2\pi) = 1$. (d) Zoom near origin of figure panel (b). (e) Zoom near origin of figure panel (c). (f) Scattered solution measured at the 201 receivers positioned at fixed height $x_2 = 6$

oscillatory noise in the reconstructed models, which could certainly be reduced by incorporating a regularization criterion in the minimization (Faucher et al. 2020c).

In Fig. 15, we conduct a similar computational experiment, but increasing the contrast in the included heart shape where f has now a value of 2; see Fig. 15. We provide single and multi-frequency reconstructions and observe that large wavelengths still provide a smooth reconstruction. The high contrast in the heart is well recovered.

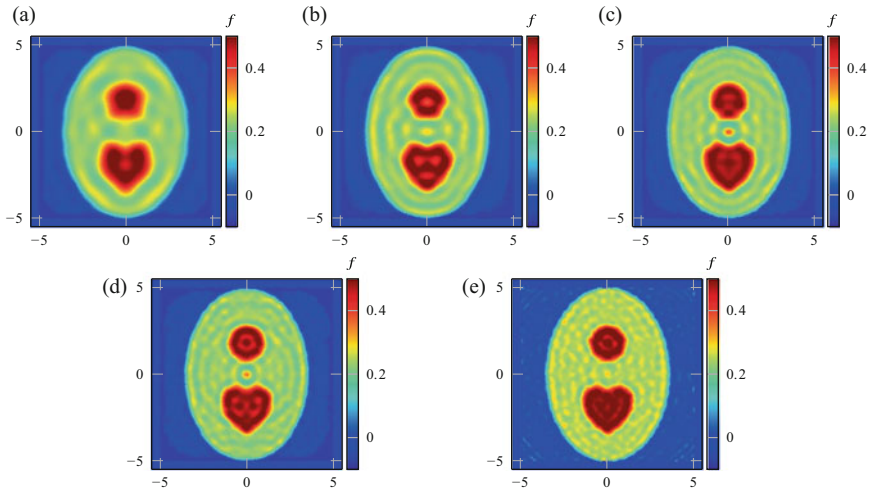


Fig. 14 Reconstruction of the model Fig. 13a with FWI and starting from a homogeneous background with $f = 0$. The models are given at frequency $\omega/(2\pi) = 1$ and the wave speed is equal to 1 in the background. (a) Using single-frequency, $\omega/(2\pi) = 0.7$ (PSNR 22.38). (b) Using single-frequency, $\omega/(2\pi) = 1$. (PSNR 22.91). (c) Using single-frequency, $\omega/(2\pi) = 1.2$. (PSNR 23.10). (d) Using single-frequency, $\omega/(2\pi) = 1.4$. (PSNR 23.31). (e) Using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 27.28)

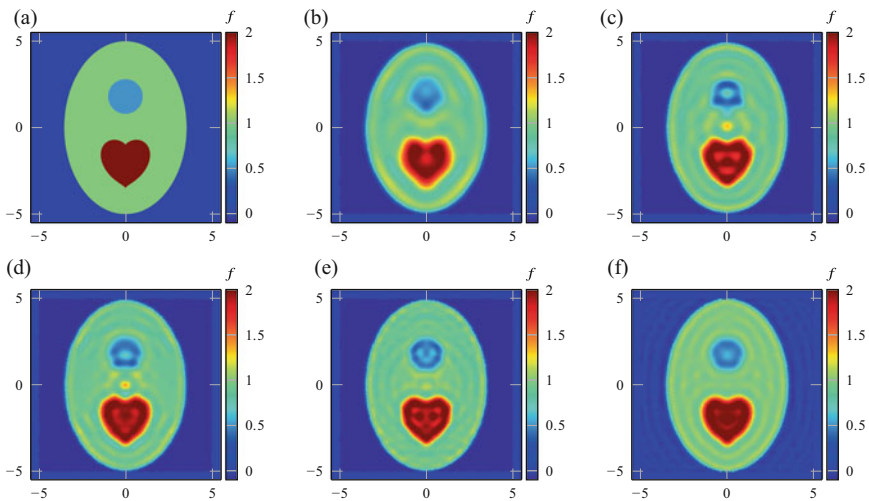


Fig. 15 Reconstruction with FWI starting from a homogeneous background with $f = 0$. The models are given at frequency $\omega/(2\pi) = 1$ and the wave speed is equal to 1 in the background (a) True model. (b) Using single-frequency, $\omega/(2\pi) = 0.7$ (PSNR 25.04). (c) Using single-frequency, $\omega/(2\pi) = 1$ (PSNR 25.91). (d) Using single-frequency, $\omega/(2\pi) = 1.2$ (PSNR 26.19). (e) Using single-frequency, $\omega/(2\pi) = 1.4$ (PSNR 26.75). (f) Using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$ (PSNR 28.76)

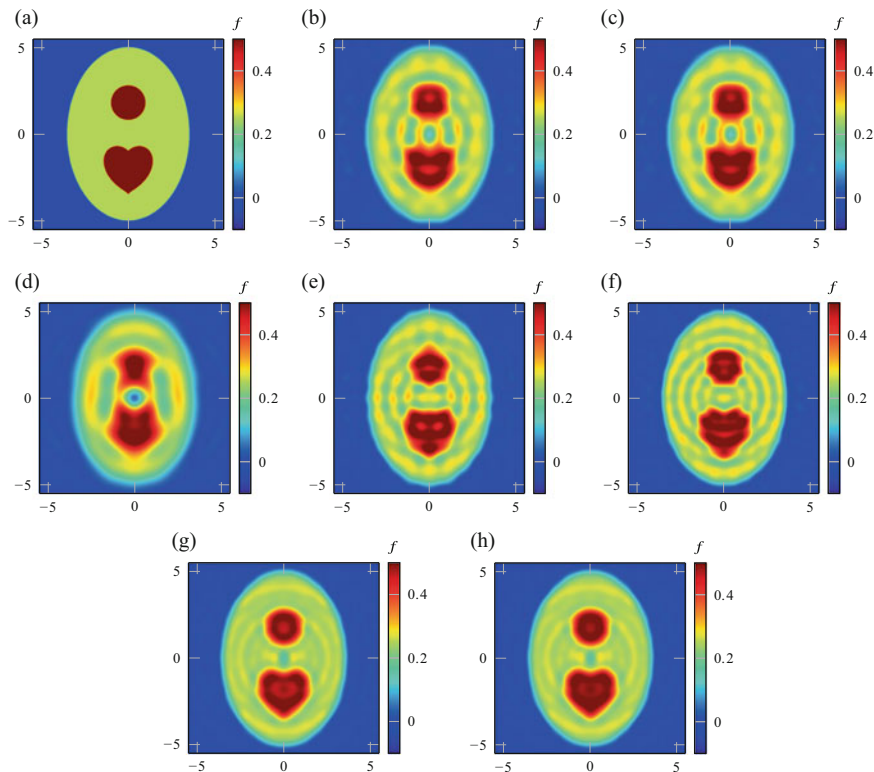


Fig. 16 Reconstructions for different frequencies of the incident wave. The PSNR is computed on the visible part of the grid for the real part of the reconstruction, since we know that f must be real (a) True model f . (b) Born reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 24.69). (c) Rytov reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 24.66). (d) Rytov reconstruction at frequency $\omega/(2\pi) = 0.7$. (PSNR 22.72). (e) Rytov reconstruction at frequency $\omega/(2\pi) = 1.2$. (PSNR 25.32). (f) Rytov reconstruction at frequency $\omega/(2\pi) = 1.4$. (PSNR 26.14). (g) Born reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 26.37). (h) Rytov reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 26.37)

Reconstruction Using Born and Rytov Approximations

We perform the reconstruction with Algorithm 2 of the test model f from Fig. 13. In the following tests, we discretize f on a finer grid of resolution $N = 720$, which covers the radius $r_s = 15/\sqrt{2}$. The PSNR is computed only on the central part of the grid that is visible in the image. Since we know that the f is real-valued, we take only the real part of the reconstruction. For simplicity, we use a constant number of 12 iterations in the conjugate gradient method of the inverse NDFT.

The Born and Rytov reconstructions are shown in Fig. 16, where the data u is the same as in section “Reconstruction Using FWI”. The reconstruction with a higher frequency of the incident wave is more accurate, since it provides a larger k-space

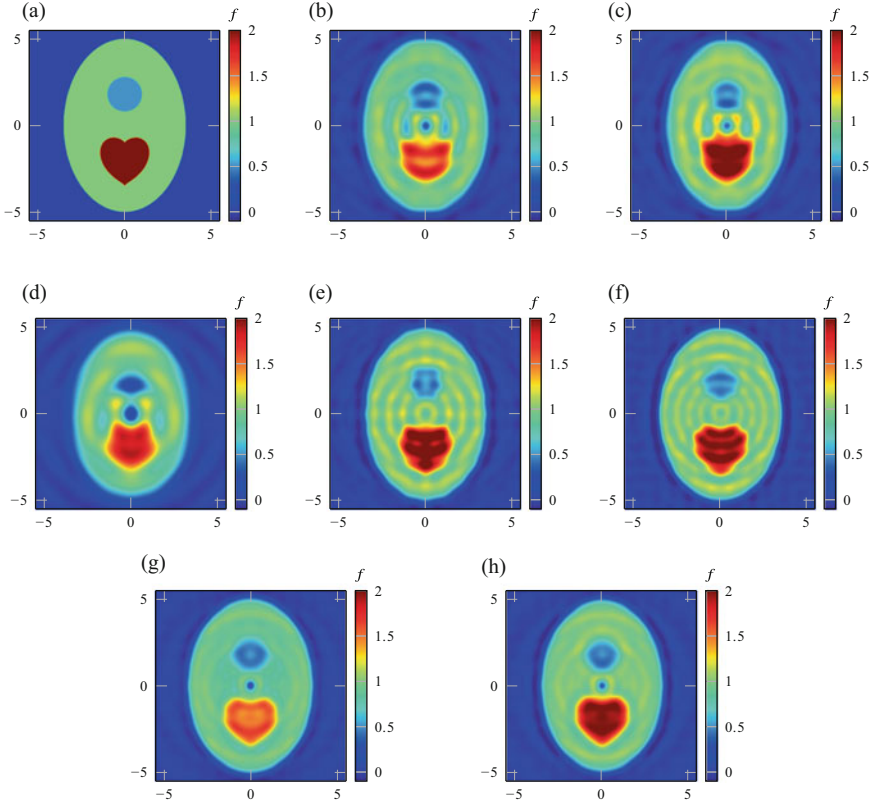


Fig. 17 Reconstructions with a higher contrast, where the rest of the setting is the same as in Fig. 16 (a) True model f . (b) Born reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 23.53). (c) Rytov reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 24.47). (d) Rytov reconstruction at frequency $\omega/(2\pi) = 0.7$. (PSNR 21.78). (e) Rytov reconstruction at frequency $\omega/(2\pi) = 1.2$. (PSNR 25.55). (f) Rytov reconstruction at frequency $\omega/(2\pi) = 1.4$. (PSNR 26.40). (g) Born reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 23.77). (h) Rytov reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 25.92)

coverage, which is the disk of radius $\sqrt{2}k_0 = \sqrt{2}\omega$, see section “[Fourier Diffraction Theorem](#)”. Moreover, the multi-frequency reconstruction is shown in Fig. 16g and h. Even though the multi-frequency setup covers the same disk in k -space, it still seems superior because we have more data points of the Fourier transform $\mathcal{F}f$.

For the similar model from Fig. 15a with a higher contrast, the reconstructions with Born and Rytov approximation differ more from the FWI reconstruction because of the more severe scattering; see Fig. 17. In general, we can expect the FWI reconstruction to be better since it is a numerical approximation of the wave equation, of which the Born or Rytov approximations are just linearizations.

Reconstruction of Embedded Shapes: Phantom 2

We now consider the case with combinations of smaller convex and non-convex shapes included in the background medium.

Reconstruction Using FWI

In Figs. 18 and 19, we show the model perturbation, which consist in small objects buried in the background. FWI is carried out with single and multiple frequencies, while we investigate a mild contrast in Fig. 18 (where f is at most 0.5) and a stronger contrast in Fig. 19 (where f is at most 2). We see that the model can be recovered using a single frequency, which has to be selected depending on the contrast. As an alternative, multi-frequency data appears to be a robust candidate and always provides a good reconstruction, for both the object's shape and amplitude. The reconstruction quality with high contrast in Fig. 19 seems to be of a similar level as with low contrast.

Reconstruction Using Born and Rytov Approximations

In Fig. 20, we show the reconstruction using Born and Rytov approximations. Here, we can clearly see that we need a higher frequency in order to resolve small features of the object. However, the reconstructions are still inferior to the FWI.

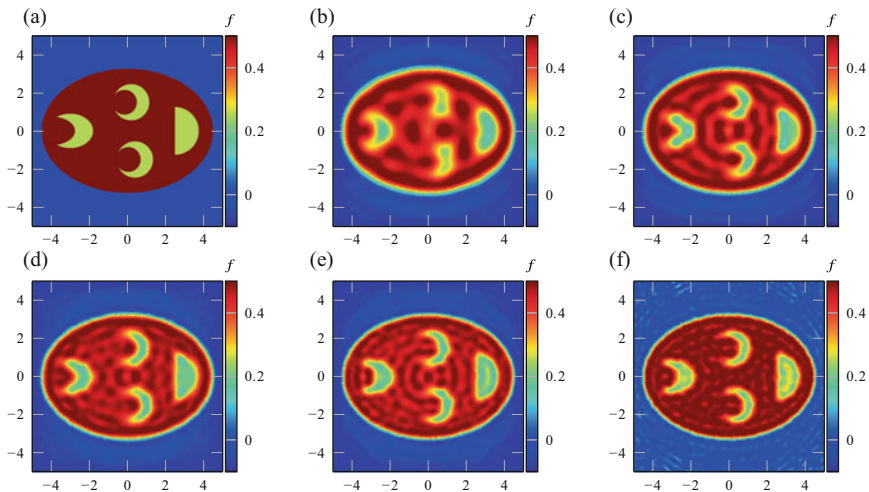


Fig. 18 Reconstruction with FWI starting from a homogeneous background with $f = 0$. The models are given at frequency $\omega/(2\pi) = 1$ and the wave speed is equal to 1 in the background (a) True contrast. (b) Using single-frequency, $\omega/(2\pi) = 0.7$ (PSNR 18.91). (c) Using single-frequency, $\omega/(2\pi) = 1$ (PSNR 19.50). (d) Using single-frequency, $\omega/(2\pi) = 1.2$ (PSNR 19.90). (e) Using single-frequency, $\omega/(2\pi) = 1.4$ (PSNR 20.10). (f) Using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$ (PSNR 23.04)

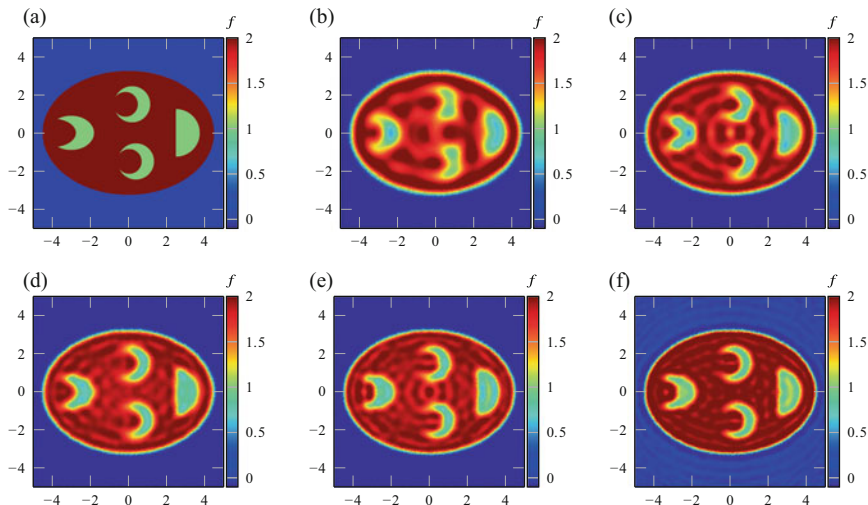


Fig. 19 Reconstruction with FWI starting from a homogeneous background with $f = 0$. The models are given at frequency $\omega/(2\pi) = 1$ and the wave speed is equal to 1 in the background (a) True contrast. (b) Using single-frequency, $\omega/(2\pi) = 0.7$ (PSNR 20.75). (c) Using single-frequency, $\omega/(2\pi) = 1$ (PSNR 21.50). (d) Using single-frequency, $\omega/(2\pi) = 1.2$ (PSNR 22.27). (e) Using single-frequency, $\omega/(2\pi) = 1.4$ (PSNR 22.72). (f) Using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$ (PSNR 22.82)

With a higher contrast, the reconstruction with the Rytov approximation is considerably better than the one with the Born approximation; see Fig. 21. This is consistent with Remark 1. Interestingly, the shapes reconstruction in high contrast barely profits from taking frequencies higher than 1, even though the k-space coverage is larger. In this situation, the Rytov reconstruction is almost comparable to the one with lower contrast, but still worse than the FWI.

Computational Costs

Computational cost of FWI. The computational cost of FWI comes from the discretization and resolution of the wave problem Equation 15 for each of the sources in the acquisition, coupled with the iterative procedure of Algorithm 1. In our numerical experiments, we use the software `hawen` for the iterative inversion, Faucher (2021), Footnote 1, which relies on the Hybridizable discontinuous Galerkin discretization, Cockburn et al. (2009) and Faucher and Scherzer (2020). The number of degrees of freedom for the discretization depends on the number of cells in the mesh and the polynomial order. In the inversion experiments, we use a fixed mesh for all iterations, with about fifty thousand cells. On the other hand, the polynomial order is selected depending on the wavelength on each cell.

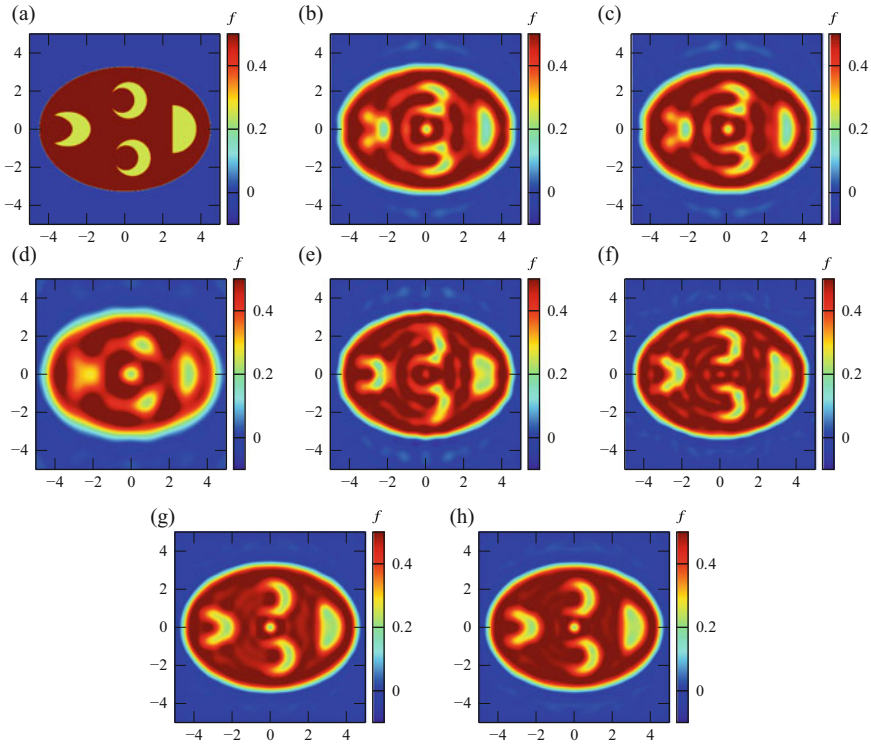


Fig. 20 Reconstructions of the more complicated shapes. The models are given at frequency $\omega/(2\pi) = 1$. (a) True model f . (b) Born reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 20.14). (c) Rytov reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 20.10). (d) Rytov reconstruction at frequency $\omega/(2\pi) = 0.7$. (PSNR 18.00). (e) Rytov reconstruction at frequency $\omega/(2\pi) = 1.2$. (PSNR 21.29). (f) Rytov reconstruction at frequency $\omega/(2\pi) = 1.4$. (PSNR 22.17). (g) Born reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 21.91). (h) Rytov reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 21.93)

That is, each of the cells in the mesh is allowed to have a different order (here between 3 to 7); see Faucher and Scherzer (2020). Then, when the frequency changes, while the mesh remains the same, the order of the polynomial evolves accordingly to the change of wavelength. Once the wave equation, Equation 15, is discretized, we obtain a linear system which size is the number of degrees of freedom that must be solved for the different sources (i.e., the different incident angles). We rely on the direct solver MUMPS, Amestoy et al. (2019), such that once the matrix factorization is computed, the numerical cost of having several sources (i.e., multiple right-hand sides in the linear system) is drastically mitigated, motivating the use of a direct solver instead of an iterative one.

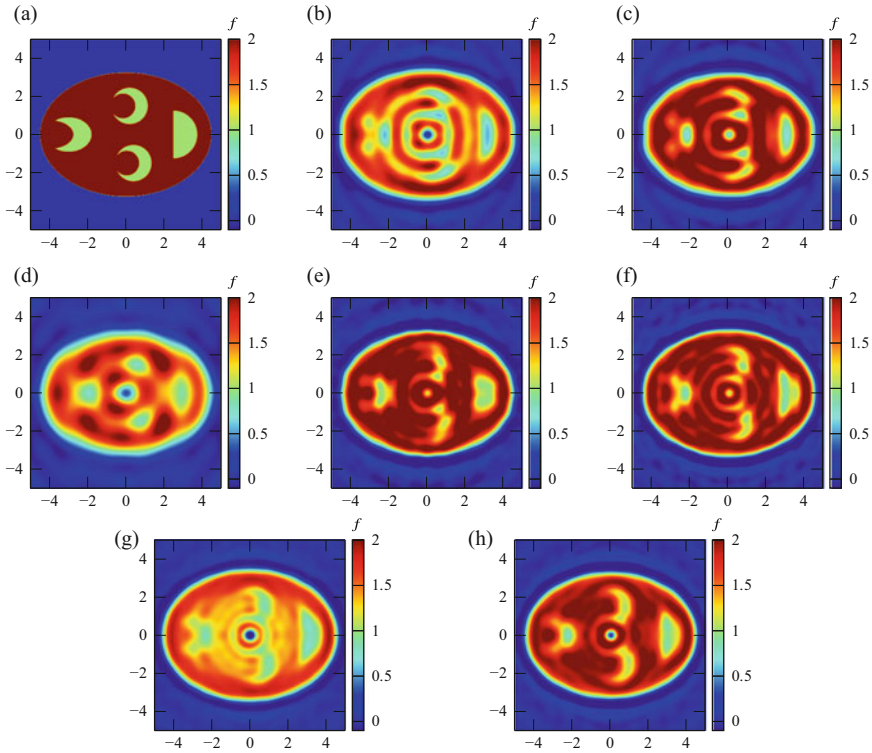


Fig. 21 Reconstructions of the more complicated shapes with a higher contrast than in Fig. 20. The models are given at frequency $\omega/(2\pi) = 1$. (a) True model f . (b) Born reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 17.16). (c) Rytov reconstruction at frequency $\omega/(2\pi) = 1$. (PSNR 18.73). (d) Rytov reconstruction at frequency $\omega/(2\pi) = 0.7$. (PSNR 16.10). (e) Rytov reconstruction at frequency $\omega/(2\pi) = 1.2$. (PSNR 20.14). (f) Rytov reconstruction at frequency $\omega/(2\pi) = 1.4$. (PSNR 21.15). (g) Born reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 16.92). (h) Rytov reconstruction using multi-frequency, $\omega/(2\pi) \in \{0.7, 1, 1.2, 1.4\}$. (PSNR 20.33)

Our numerical experiments have been carried out on the Vienna Scientific Cluster VSC-4,² using 48 cores. For the reconstructions of Figs. 14, 15, 18, and 19, the size of the computational domain is 40×40 , with about 350.000 degrees of freedom. Using single-frequency data, 50 iterations are performed in Algorithm 1, and the total computational time is of about 40 min. In the case of multiple frequencies, we have a total of 120 iterations, and the computational time is of about 1 h 45 min.

Computational cost of Born and Rytov approximations. The conjugate gradient method used in the inverse NDFT requires in each iteration step the evaluation

²<https://vsc.ac.at/>

of an NDFT Equation 28 and its adjoint. We utilize the nonequispaced fast Fourier transform (NFFT) algorithm (Keiner et al. n.d.), implemented in the open-source software library `nfft` (Keiner et al. 2009), which can compute an NDFT in $O(N^2 \log N + M)$ arithmetic operations, which is considerably less than the $O(N^2 M)$ operations of a straightforward implementation of Equation 28.

The numerical simulations of section “[Reconstruction Using Born and Rytov Approximations](#)” have been carried out on a 4-core Intel Core i5-6500 processor. We used 12 iterations of the conjugate gradient method and noted that the reconstruction quality hardly benefits from a higher number of iterations. The reconstruction of an image took about 1 second, with a grid size 720×720 of f and 200×100 data points of u for each frequency. Therefore, the numerical computation of the Born and Rytov approximations is much faster than the FWI.

Conclusion

We study the imaging problem for diffraction tomography, where wave measurements are used to quantitatively reconstruct the physical properties, i.e., the refractive index. The forward operator that describes the wave propagation corresponds with the Helmholtz equation, which, under the assumption of small background perturbations, can be represented via the Born and Rytov approximations.

Firstly, we have compared different forward models in terms of the resulting measured data u . It highlights that, even in the case of a small circular object, the Born approximation is not entirely accurate to represent the total wave field given by the Helmholtz equation. In addition, the source that initiates the phenomenon (e.g., a point source located very far from the object, or simultaneous point source along a line) also plays an important role as it changes the resulting signals, hence leading to systematic differences depending on the choice of forward model. We found that the line source model approximates the plane wave pretty well.

Secondly, we have carried out the reconstruction using data from the total field u^{tot} and compared the efficiency of the Full Waveform Inversion method (FWI) with that of the Born and Rytov approximations. FWI works directly with the Helmholtz problem, Equation 15, hence giving a robust approach that can be implemented in all configurations, however at the cost of possibly intensive computations. On the other hand, the Born and Rytov are computationally cheap, but lack accuracy when the object is too large or when the contrast is too strong. We have also noted that the Rytov approximation gives better results than the Born one. Furthermore, for all reconstruction methods, we have shown that using data of multiple frequencies allows to improve the robustness of the reconstruction by providing information on multiple wavelengths.

Acknowledgments We thank the anonymous reviewer for carefully reading the manuscript and making various suggestions for its improvement. This work is supported by the Austrian Science Fund (FWF) within SFB F68 (“Tomography across the Scales”), Projects F68-06 and F68-07. FF is funded by the Austrian Science Fund (FWF) under the Lise Meitner fellowship M 2791-N.

Funding by the DFG under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, Projektnummer: 390685689) as well as by the DFG project STE 571/19 (Projektnummer: 495365311) is gratefully acknowledged. For the numerical experiments, we acknowledge the use of the Vienna Scientific Cluster VSC-4 (<https://vsc.ac.at/>).

References

- Amestoy, P.R., Buttari, A., L'Excellent, J.-Y., Mary, T.: Performance and scalability of the block low-rank multifrontal factorization on multicore architectures. *ACM Trans. Math. Softw. (TOMS)* **45**(1), 1–26 (2019). <https://doi.org/10.1145/3242094>
- Bamberger, A., Chavent, G., Lailly, P.: About the stability of the inverse problem in the 1-d wave equation. *J. Appl. Math. Optim.* **5**, 1–47 (1979)
- Barucq, H., Chavent, G., Faucher, F.: A priori estimates of attraction basins for nonlinear least squares, with application to Helmholtz seismic inverse problem. *Inverse Probl.* **35**(11), 115004 (2019). <https://doi.org/10.1088/1361-6420>
- Bednar, J.B., Shin, C., Pyun, S.: Comparison of waveform inversion, part 2: phase approach. *Geophys. Prospect.* **55**(4), 465–475 (2007). ISSN: 1365-2478. <https://doi.org/10.1111/j.1365-2478.2007.00618.x>
- Beinert, R., Quellmalz, M.: Total variation-based reconstruction and phase retrieval for diffraction tomography *SIAM J. Imaging Sci.* **15**(3), 1373–1399 (2022). ISSN: 1936-4954. <https://doi.org/10.1137/22M1474382>
- Bunks, C., Saleck, F.M., Zaleski, S., Chavent, G.: Multiscale seismic waveform inversion. *Geophysics* **60**(5), 1457–1473 (1995). <https://doi.org/10.1190/1.1443880>
- Chen, B., Stannnes, J.J.: Validity of diffraction tomography based on the first Born and the first Rytov approximations. *Appl. Opt.* **37**(14), 2996 (1998). <https://doi.org/10.1364/ao.37.002996>
- Clément, F., Chavent, G., Gómez, S.: Migration-based travelttime wave-form inversion of 2-D simple structures: a synthetic example. *Geophysics* **66**(3), 845–860 (2001). <https://doi.org/10.1190/1.1444974>
- Cockburn, B., Gopalakrishnan, J., Lazarov R.: Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.* **47**(2), 1319–1365 (2009). <https://doi.org/10.1137/070706616>
- Colton, D., Kress, R.: *Inverse Acoustic and Electromagnetic Scattering Theory*. Applied Mathematical Sciences, vol. 93, 3rd edn. Springer, Berlin (2013). ISBN: 978-1-4614-4941-6. <https://doi.org/10.1007/978-1-4614-4942-3>
- Devaney, A.: A filtered backpropagation algorithm for diffraction tomography. *Ultrason. Imaging* **4**(4), 336–350 (1982). [https://doi.org/10.1016/0161-7346\(82\)90017-7](https://doi.org/10.1016/0161-7346(82)90017-7)
- Devaney, A.: *Mathematical Foundations of Imaging, Tomography and Wave-Field Inversion*. Cambridge University Press (2012). <https://doi.org/10.1017/CBO9781139047838>
- Engquist, B., Majda, A.: Absorbing boundary conditions for numerical simulation of waves. *Proc. Natl. Acad. Sci.* **74**(5), 1765–1766 (1977)
- Fan, S., Smith-Dryden, S., Li, G., Saleh, B.E.A.: An iterative reconstruction algorithm for optical diffraction tomography. In: *IEEE Photonics Conference (IPC)*, pp. 671–672 (2017). <https://doi.org/10.1109/ipcon.2017.8116276>
- Faucher, F.: Contributions to seismic full waveform inversion for time harmonic wave equations: Stability estimates, convergence analysis, numerical experiments involving large scale optimization algorithms. PhD thesis. Université de Pau et Pays de l'Ardour, pp. 1–400 (2017)
- Faucher, F.: Hawen: time-harmonic wave modeling and inversion using hybridizable discontinuous Galerkin discretization. *J. Open Source Softw.* **6**(57) (2021). <https://doi.org/10.21105/joss.02699>
- Faucher, F., Scherzer, O.: Adjoint-state method for Hybridizable Discontinuous Galerkin discretization, application to the inverse acoustic wave problem. *Comput. Methods Appl. Mech. Eng.* **372**, 113406 (2020). ISSN: 0045-7825. <https://doi.org/10.1016/j.cma.2020.113406>

- Faucher, F., Alessandrini, G., Barucq, H., de Hoop, M., Gaburro, R., Sincich, E.: Full Reciprocity-Gap Waveform Inversion, enabling sparse-source acquisition. *Geophysics* **85**(6), R461–R476 (2020a). <https://doi.org/10.1190/geo2019-0527.1>
- Faucher, F., Chavent, G., Barucq, H., Calandra, H.: A priori estimates of attraction basins for velocity model reconstruction by time-harmonic Full Waveform Inversion and Data-Space Reflectivity formulation. *Geophysics* **85**(3), R223–R241 (2020b). <https://doi.org/10.1190/geo2019-0251.1>
- Faucher, F., Scherzer O., Barucq, H.: Eigenvector models for solving the seismic inverse problem for the Helmholtz equation. *Geophys. J. Int.* (2020c). ISSN: 0956-540X. <https://doi.org/10.1093/gji/ggaa009>
- Faucher, F., de Hoop, M.V., Scherzer, O.: Reciprocitygap misfit functional for Distributed Acoustic Sensing, combining data from passive and active sources. *Geophysics* **86**(2), R211–R220 (2021). ISSN: 0016-8033. <https://doi.org/10.119/geo2020-0305.1>
- Fichtner, A., Kennett, B.L., Igel, H., Bunge, H.-P.: Theoretical back ground for continental- and global-scale full-waveform inversion in the time–frequency domain. *Geophys. J. Int.* **175**(2), 665–685 (2008). <https://doi.org/10.1111/j.1365-246X.2008.03923.x>
- Gbur, G., Wolf, E.: Hybrid diffraction tomography without phase information. *J. Opt. Soc. Am. A* **19**(11), 2194–2202 (2002). <https://doi.org/10.1364/OL27.001890>
- Hanke, M.: Conjugate Gradient Type Methods for Ill-Posed Problems. Pitman Research Notes in Mathematics Series, vol. 327. Longman Scientific & Technical, Harlow (1995)
- Hesthaven, J.S., Warburton, T.: Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer Science & Business Media (2007). <https://doi.org/10.1007/978-0-387-72067-8>
- Hielscher, R., Potts, D., Quellmalz, M.: An SVD in spherical surface wave tomography. In: Hofmann, B., Leitao, A., Zubelli, J.P. (eds.) *New Trends in Parameter Identification for Mathematical Models*. Trends in Mathematics, pp. 121–144. Birkhäuser, Basel (2018). ISBN: 978-3-319-70823-2. https://doi.org/10.1007/978-3-319-70824-9_7
- Hielscher, R., Quellmalz, M.: Optimal mollifiers for spherical de-convolution. *Inverse Probl.* **31**(8), 085001 (2015). <https://doi.org/10.1088/02.665611/31/8/085001>
- Hielscher, R., Quellmalz, M.: Reconstructing a function on the sphere from its means along vertical slices. *Inverse Probl. Imaging* **10**(3), 711–739 (2016). ISSN: 1930-8337. <https://doi.org/10.3934/ipi.2016018>
- Horstmeyer, R., Chung, J., Ou, X., Zheng, G., Yang, C.: Diffraction tomography with Fourier ptychography. *Optica* **3**(8), 827–835 (2016). <https://doi.org/10.1364/OPTICA.3.000827>
- Huynh-Thu, Q., Ghanbari, M.: The accuracy of PSNR in predicting video quality for different video scenes and frame rates. *Telecommun. Syst.* **49**(1), 35–48 (2010). <https://doi.org/10.1007/s112350109351x>
- Iwata, K., Nagata, R.: Calculation of refractive index distribution from interferograms using the Born and Rytov’s approximation. *Jpn. J. Appl. Phys.* **14**(S1), 379–383 (1975). <https://doi.org/10.7567/jjaps.14s1.379>
- Kak, A.C., Slaney M.: Principles of Computerized Tomographic Imaging. Classics in Applied Mathematics, vol. 33. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2001)
- Kaltenbacher, B.: Minimization based formulations of inverse problems and their regularization. *SIAM J. Optim.* **28**(1), 620–645 (2018). <https://doi.org/10.1137/17M1124036>
- Keiner, J., Kunis, S., Potts, D.: Using NFFT3 – a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Softw.* **36**, Article 19, 1–30 (2009). <https://doi.org/10.1145/1555386.1555388>
- Keiner, J., Kunis, S., Potts, D.: NFFT 3.5, C subroutine library (n.d.). <https://www.tu-chemnitz.de/~potts/nfft>
- Kirby, R.M., Sherwin, S.J., Cockburn, B.: To CG or to HDG: a comparative study. *J. Sci. Comput.* **51**(1), 183–212 (2012). <https://doi.org/10.1007/s10915-011-9501-7>

- Kiritsits, C., Quellmalz, M., Ritsch-Marte, M., Scherzer, O., Setterqvist, E., Steidl, G.: Fourier reconstruction for diffraction tomography of an object rotated into arbitrary orientations. *Inverse Probl.* (2021). ISSN: 0266-5611. <https://doi.org/10.1088/1361-6420/ac2749>
- Knopp, T., Kunis, S., Potts, D.: A note on the iterative MRI reconstruction from nonuniform k-space data. *Int. J. Biomed. Imag.* (2007). <https://doi.org/10.1155/2007/24727>
- Kunis, S., Potts, D.: Stability results for scattered data interpolation by trigonometric polynomials. *SIAM J. Sci. Comput.* **29**, 1403–1419 (2007). <https://doi.org/10.1137/060665075>
- Lailly, P.: The seismic inverse problem as a sequence of before stack migrations. In: Bednar, J.B. (ed.) *Conference on Inverse Scattering: Theory and Application*, pp. 206–220. Society for Industrial and Applied Mathematics (1983)
- Luo, Y., Schuster, G.T.: Wave-equation travelttime inversion. *Geophysics* **56**(5), 645–653 (1991). <https://doi.org/10.1190/1.1443081>
- Maleki, M.H., Devaney, A.: Phase-retrieval and intensity-only reconstruction algorithms for optical diffraction tomography. *J. Opt. Soc. Am. A* **10**(5), 1086 (1993). <https://doi.org/10.1364/josaa.10.001086>
- Métivier, L., Brossier, R., Mérigot, Q., Oudet, E., Virieux, J.: Measuring the misfit between seismograms using an optimal transport distance: application to full waveform inversion. *Geophys. J. Int.* **205**(1), 345–377 (2016). <https://doi.org/10.1093/gji/ggw014>
- Monk, P.: *Finite Element Methods for Maxwell's Equations*. Oxford University Press, Oxford (2003)
- Müller, P., Schürmann, M., Guck, J.: ODTbrain: a Python library for full-view, dense diffraction tomography. *BMC Bioinform.* **16**(367) (2015). <https://doi.org/10.1186/s12859-015-0764-0>
- Müller, P., Schürmann, M., Guck, J.: *The Theory of Diffraction Tomography* (2016). arXiv: 1507.00466 [q-bio.QM]
- Natterer, F.: *The Mathematics of Computerized Tomography*, x+222. B. G. Teubner, Stuttgart (1986). ISSN: 3-519-02103-X
- Natterer, F., Wübbeling, F.: *Mathematical Methods in Image Reconstruction. Monographs on Mathematical Modeling and Computation*, vol. 5. SIAM, Philadelphia (2001)
- Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research, 2nd edn. Springer, Berlin (2006)
- Plessix, R.-E.: A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophys. J. Int.* **167**(2), 495–503 (2006). <https://doi.org/10.1111/j.1365-246X.2006.02978.x>
- Plonka, G., Potts, D., Steidl, G., Tasche, M.: *Numerical Fourier Analysis. Applied and Numerical Harmonic Analysis*. Birkhäuser (2018). ISSN: 978-3-030-04305-6. <https://doi.org/10.1007/978-3-030-04306-3>
- Potts, D., Steidl, G.: A new linogram algorithm for computerized tomography. *IMA J. Numer. Anal.* **21**, 769–782 (2001). <https://doi.org/10.1093/imanum/21.3.769>
- Pratt, R.G., Shin, C., Hick, G.J.: Gauss–Newton and full Newton methods in frequency–space seismic waveform inversion. *Geophys. J. Int.* **133**(2), 341–362 (1998). <https://doi.org/10.1046/j.1365-246X.1998.00498.x>
- Pyun, S., Shin, C., Bednar, J.B.: Comparison of waveform inversion, part 3: amplitude approach. *Geophys. Prospect.* **55**(4), 477–485 (2007). ISSN: 1365-2478. <https://doi.org/10.1111/j.1365-2478.2007.00619.x>
- Shin, C., Pyun, S., Bednar, J.B.: Comparison of waveform inversion, part 1: conventional wavefield vs logarithmic wavefield. *Geophys. Prospect.* **55**(4), 449–464 (2007). ISSN: 1365-2478. <https://doi.org/10.1111/j.1365-2478-2007.00617.x>
- Slaney, M., Kak, A.C., Larsen, L.E.: Limitations of imaging with first-order diffraction tomography. *IEEE Trans. Microw. Theory Techn.* **32**(8), 860–874 (1984). <https://doi.org/10.1109/TMTT.1984.1132783>
- Sung, Y., Choi, W., FangYen, C., Badizadegan, K., Dasari, R.R., Feld, M.S.: Optical diffraction tomography for high resolution live cell imaging. *Opt. Express* **17**(1), 266–277 (2009)

- Tarantola, A.: Inversion of seismic reflection data in the acoustic approximation. *Geophysics* **49**, 1259–1266 (1984). <https://doi.org/10.1190/1.1441754>
- Van Leeuwen, T., Mulder, W.: A correlation-based misfit criterion for wave-equation traveltime tomography. *Geophys. J. Int.* **182**(3), 1383–1394 (2010)
- Virieux, J.: SH-wave propagation in heterogeneous media: velocity-stress finite-difference method. *Geophysics* **49**(11), 1933–1942 (1984)
- Virieux, J., Operto, S.: An overview of full-waveform inversion in exploration geophysics. *Geophysics* **74**(6), WCC1–WCC26 (2009). <https://doi.org/10.1190/1.3238367>
- Wedberg, T.C., Stamnes, J.J.: Comparison of phase retrieval methods for optical diffraction tomography. *Pure Appl. Opt.* **4**, 39–54 (1995). <https://doi.org/10.1088/0963-9659/4/1/005>
- Wolf, E.: Three-dimensional structure determination of semi-transparent objects from holographic data. *Opt. Commun.* **1**, 153–156 (1969)



Models for Multiplicative Noise Removal

9

Xiangchu Feng and Xiaolong Zhu

Contents

Introduction	314
Variational Methods with Different Data Fidelity Terms	317
Statistical Property Based Models	318
MAP-Based Models	319
Root and Inverse Transformation-Based Models	320
Variational Methods with Different Regularizers	325
TV Regularization	325
Sparse Regularization	327
Nonconvex Regularization	330
Multitasks	334
Root Transformation	334
Fractional Transformation	335
Nonlocal Methods	335
Indirect Method	336
Direct Method	337
DNN Method	338
Indirect Method	339
Direct Method	342
Conclusion	343
References	343

Abstract

Image denoising is the most important step in image processing for further image analysis. It is an important topic in many applications, such as object recognition, digital entertainment, etc. The digital image can be corrupted with noise during

X. Feng (✉) · X. Zhu
School of Mathematics and Statistics, Xidian University, Xi'an, China
e-mail: xcfeng@mail.xidian.edu.cn; zx1001@aliyun.com

acquisition, storage, and transmission. Noise can be classified as additive noise, multiplicative noise, and non-additive non-multiplicative noise (such as salt and pepper noise, Poisson noise). The main properties of a good image denoising model are that it will remove noise while preserving details of the image.

This chapter aims to present a review of multiplicative denoising models, especially for the multiplicative Gamma noise. Similar to denoising for additive Gaussian noise, these denoising approaches can be categorized as variational methods, non-local methods, and deep neural network-based methods. Due to space constraints, this chapter only discusses some of them. The rest of this chapter is organized as follows. Section “[Introduction](#)” is an introduction and section “[Variational Methods with Different Data Fidelity Terms](#)” describes variational methods with different data fidelity terms. Section “[Variational Methods with Different Regularizers](#)” introduces variational methods with different regularizers. Sections “[Multitasks](#)” to “[DNN Method](#)” describe multitasks, nonlocal, and deep neural network (DNN) methods. Finally, section “[Conclusion](#)” presents our conclusions.

Keywords

Multiplicative denoising · Variational methods · Nonlocal methods · Multitasks · DNN methods

Introduction

The most common noise encountered in real applications is thermal noise. It is additive and follows a Gaussian distribution with zero mean. Many image denoising approaches have been proposed for additive Gaussian noise (Shao et al. 2013; Lebrun et al. 2012). Generally, these approaches can be categorized as spatial domain, transform Domain, and learning-based method. Spatial domain methods include energy function methods, which exploit maximum a posteriori probability (MAP) estimation as the main tool, and nonlocal filters, which exploit the similarities between patches in an image. Transform domain methods consider transforming images into other domains, in which similarities of transformed coefficients are considered. Learning-based methods use sparse representations on a redundant dictionary or train a deep neural network through many training samples. In fact, after so many researches, the denoising results of these methods for additive Gaussian noise are close to the limitation (Chatterjee and Milanfar 2009).

This chapter focuses on reducing multiplicative speckle noise, especially for multiplicative Gamma noise. In many coherent imaging systems, digital images are usually accompanied by speckle noise (Singh and Jain 2016). It is caused by coherent processing of backscattered signals from multiple distributed targets. Speckle noise can be described as random multiplicative noise. It appears in many applications, e.g., in ultrasound imaging, where the noise follows a Rayleigh distribution; in electronic microscopy, where the multiplicative noise is

Poisson noise; and in synthetic aperture radar (SAR), where the noise follows a Gamma distribution. In fact, speckle in a SAR image is caused by constructive and destructive interference of coherent waves reflected by the many elementary scatterers contained within the imaged resolution cell. The magnitude of the complex observations of SAR can usually be modeled as corrupted by multiplicative Rayleigh noise. As a consequence, the noise present in the square of the magnitude, the so-called intensity, is exponentially distributed. To improve the quality of such data, a common approach in SAR imaging is to average independent intensity observations of the same scene to obtain so-called multi look data, which is then contaminated by multiplicative Gamma noise.

The Gamma noise model and Gamma distribution are given below. If we use f to denote the image intensity that the SAR measures for a given pixel whose backscattering coefficient is u , and assume that the SAR image represents an average of L looks (independent samples or pixels), then f is related to u by the multiplicative model

$$f = un \quad (1)$$

where n is the normalized fading random variable in the intensity image, following a Gamma distribution with unit mean and variance $1/L$. The probability density function (PDF) of n is given by

$$p_n(n) = \frac{L^n n^{L-1} e^{-Ln}}{\Gamma(L)}, \quad n \geq 0, \quad L \geq 1. \quad (2)$$

where $\Gamma(\cdot)$ denotes the gamma function.

The natural idea when dealing with this problem is to convert it into an additive problem by applying the logarithm while using a Gaussian distribution to approximate the distribution after the logarithm (Xie et al. 2002). The natural logarithmic transformation converts (1) into

$$\tilde{f} = \tilde{u} + \tilde{n} \quad (3)$$

where $\tilde{f} = \ln(f)$, $\tilde{u} = \ln(u)$ and $\tilde{n} = \ln(n)$. Owing to the monotonic of the logarithmic function, the probability density function of the random variable \tilde{n} can be obtained from

$$p_{\tilde{n}}(\tilde{n}) = p_n(e^{\tilde{n}})e^{\tilde{n}} \quad (4)$$

which leads to

$$p_{\tilde{n}}(\tilde{n}) = \frac{L^L e^{\tilde{n}L} e^{-Le^{\tilde{n}}}}{\Gamma(L)} \quad (5)$$

and the mean of \tilde{n} is given by (Hoekman 1991)

Table 1 Cumulative distribution distance D

Looks L	1	2	3	4	5	6	9	12	16
Intensity	0.242	0.145	0.099	0.078	0.066	0.058	0.045	0.039	0.033
Log(Intensity)	0.070	0.050	0.040	0.034	0.030	0.028	0.023	0.019	0.016

$$E(\tilde{n}) = \psi(L) - \ln(L) \tag{6}$$

where $\psi(\cdot)$ is the Digamma function defined by

$$\psi(\cdot) = \frac{d}{dx} \ln \Gamma(x) \tag{7}$$

The new variance is given by

$$var(\tilde{n}) = \psi(1, L) \tag{8}$$

where $\psi(1, L)$ is known as the first-order Polygamma function of L .

The distance between cumulative distribution is defined as the maximum value of the absolute difference between the two cumulative distribution functions. For evaluating how close a general distribution $p(x)$ is to a Gaussian probability density function $g(x)$, their cumulative distributions are firstly calculated, denoted as $P(x)$ and $G(x)$. Then the distance D is given by (Table 1)

$$D = \max_{-\infty < x < \infty} |P(x) - G(x)| \tag{9}$$

As the number of looks increases, the probability density function of a speckle random variable approaches the Gaussian probability density function. When the number of looks is small, if Gaussian noise is used to approximate, the error is large. Therefore, it is necessary to directly process the multiplicative noise. Because speckle noise is non-Gaussian signal and spatially independent (Ullah et al. 2016; Abolhassani and Rostami 2012; Le and Vese 2003), noise removal is more complex and more challenging than additive noise removal.

As we know, the restoration process is to recover u from the degraded image $f = un$ and preserve image features including edges, point targets, textures, and so on. To achieve this objective, a variety of speckle noise removal methods has been proposed. Some methods have been well-known, such as variational methods, dictionary learning methods, nonlocal methods, deep neural network methods, etc.

A variational model for speckle noise removal normally consists of the regularization term (log-prior) and data fidelity term (log-likelihood). For Gamma noise, a variational model (AA-model) via the maximum a posteriori estimator was derived by (Aubert and Aujol 2008). Motivated by the effectiveness of the inverse scale space, Shi and Osher developed a strictly convex general model (SO-model) for speckle noise removal (Shi and Osher 2008). By applying I-divergence as a

similarity term, an energy function (SST-model) was presented (Steidl and Teuber 2010). By applying an exponential transformation to the AA model, a globally convex model for speckle noise removal has been achieved in Jin and Yang (2010). The regularization term is commonly total variation (TV) and its variations (Xiao et al. 2010; Hu et al. 2013; Na et al. 2018).

Due to the sparse nature of the l_1 norm, TV requires the image to have some sparsity in the gradient domain. We know that the wavelet coefficients, ridgelet coefficients, or curvelet coefficients of a sharp image are sparse. Based on these, Durand et al. gave a hybrid method of curvelet field for removing multiplicative noise in Durand et al. (2010). A combination of total generalized variation filter (which has been proved to be able to reduce the blocky-effects by being aware of high-order smoothness) and shearlet transform (that effectively preserves anisotropic image features such as sharp edges, curves, and so on) was proposed in Ullah et al. (2017). In Huang et al. (2012), dictionary learning is used as a regularization term, and experimental results suggest that in terms of visual quality, peak signal-to-noise ratio, and mean absolute deviation error, the proposed algorithm outperforms many other methods. In addition to variational models, nonlocal methods (Teuber and Lang 2012; Huang et al. 2017) are also proposed. Recently, deep neural network methods (Wang et al. 2017, 2019) are presented, extensive experiments on synthetic and real images show that they achieve significant improvements over the state-of-the-art speckle reduction methods.

Variational Methods with Different Data Fidelity Terms

Usually, a variational method has two terms: a data fidelity term and a regularization term. More specifically, our interest is in recovering a true underlying image u from the noise corrupted observation $f = un$, where n is a random variable following Gamma distribution. To obtain an estimate \hat{u} , (10) is considered

$$\hat{u} = \arg \min_{u \in X} \left\{ E(u) := \phi(u, f) + \lambda \rho(u) \right\} \quad (10)$$

where $\lambda > 0$ is a tuning parameter, X is the space that the solution lies in. Depending on the model, X may be $L^2(\Omega)$, $BV(\Omega)$, etc. In the discrete case, usually $X = R^d$. In general, the data fidelity term ϕ reflects characteristics of the noise corrupting our observation, and the regularization term $\rho(\cdot)$ is a prior on the clean image u . A common choice for $\rho(\cdot)$ is total variation(TV)

$$\rho(u) = \int_{\Omega} |\nabla u| := J(u) \text{ or } |u|_{TV(\Omega)} \quad (11)$$

Statistical Property Based Models

(1) RLO-model

Under the assumption that the mean of the multiplicative noise is equal to 1 and the variance is known, Rudin, Lions, and Osher introduced the following denoising model (RLO) in Rudin et al. (2003):

$$\min_{u \in X} \left\{ J(u) + \lambda_1 \int_{\Omega} \frac{f}{u} dx + \lambda_2 \int_{\Omega} \left(\frac{f}{u} - 1 \right)^2 dx \right\} \tag{12}$$

However, only basic statistical properties, the mean, and the variance of the noise are considered in the RLO model, which somehow limits its denoising ability. We know that the likelihoods of the multiplicative Poisson noise and the likelihood of the multiplicative Rayleigh noise (Setzer et al. 2010; Denis et al. 2009) are

$$\int (u - f \log u) dx \text{ and } \int \left(\frac{1}{2} \left(\frac{f}{u} \right)^2 + \log u \right) dx, \text{ respectively.}$$

Based on the MAP model of Poisson noise and Rayleigh noise, the above model (12) can be generalized into SO-model.

(2) General SO-model

In (2008), Shi and Osher proposed a new general model, which can be fitted in different areas by setting different parameters of a , b and c

$$\int_{\Omega} \left(a \frac{f}{u} + \frac{b}{2} \left(\frac{f}{u} \right)^2 + c \log u \right) dx \tag{13}$$

This is a nonconvex variational problem, coupled with the TV regularization term; it becomes the following problem:

$$\hat{u} = \arg \min_{u \in BV(\Omega)} \left\{ J(u) + \lambda \int \left(a \frac{f}{u} + \frac{b}{2} \left(\frac{f}{u} \right)^2 + c \log u \right) dx \right\} \tag{14}$$

where $c = a + b$ for the Gammer noise. By exponential transformation $u = e^w$, the problem is reduced to a convex one

$$\hat{w} = \arg \min_{w \in BV(\Omega)} \left\{ J(w) + \lambda \int \left(af \exp(-w) + \frac{b}{2} (f)^2 \exp(-2w) + (a + b) w \right) \right\},$$

$$\hat{u} = e^{\hat{w}} \tag{15}$$

The fidelity term $H(w, f) = \int (af \exp(-w) + \frac{b}{2} (f)^2 \exp(-2w) + (a + b) w)$ is globally strictly convex. Using gradient descent and the Euler-Lagrange equation for this total variation-based problem, (16) can be obtained:

$$w_t = \nabla \cdot \frac{\nabla w}{|\nabla w|} + \lambda \left(af \exp(-w) + b (f)^2 \exp(-2w) - (a + b) \right) \tag{16}$$

Shi and Osher extended this convex model to obtain a nonlinear inverse scale space flow and its corresponding relaxed inverse scale space flow. The numerical results of SNR show significant improvement over the RLO model (Shi and Osher 2008).

MAP-Based Models

(3) AA-model

Based on the MAP estimator for multiplicative Gamma noise, Aubert and Aujol (2008) proposed to determine the denoised image as a minimizer in $S(\Omega) = \{u \in BV : u > 0\}$ of the following functional

$$\min_{u \in S(\Omega)} \lambda J(u) + \int_{\Omega} \left(\log u + \frac{f}{u} \right) dx \tag{17}$$

The AA model (17) is nonconvex; finding its global solution is a challenging task. It is known that the convex optimization method has vast applications in image processing. Therefore, many works have been designed to relieve the nonconvex AA model.

(4) SO-model

In (2008), Shi and Osher suggested to keep the data fitting term in (17) but to replace the regularizer $|\nabla u|$ by $|\nabla \log u|$. Moreover, setting as in the log-model $w := \log u$, this results in the convex function

$$\hat{w} = \arg \min_{w \in BV(\Omega)} \int_{\Omega} f e^{-w} + w dx + \lambda J(w), \hat{u} = e^{\hat{w}} \tag{18}$$

In fact, it is the exponential form of the general SO-model (14) when $b = 0$. Furthermore, to better preserve textures and details, this model was extended by Chen and Cheng in 2011, incorporating it with a spatially dependent regularization

$$\min_{w \in BV(\Omega)} \int_{\Omega} \lambda(x) \left(w + f e^{-w} \right) dx + J(w) \tag{19}$$

where $\lambda : \Omega \rightarrow R$ is a spatially varying parameter.

(5) I-divergence model

In connection with deblurring in the presence of multiplicative noise, the I-divergence, also called generalized Kullback-Leibler divergence

$$I(f, u) := \int_{\Omega} f \log \frac{f}{u} - f + u dx \tag{20}$$

is typically used as a data fitting term. The I-divergence is the Bregman distance of the function $F(u) := \int_{\Omega} u \log u - u dx$, i.e., $I(f, u) = F(f) - F(u) - \langle p, f - u \rangle$, where $p \in \partial F(u)$.

Therefore, it shares the useful properties of the Bregman distance, in particular, $I(f, u) \geq 0$. Ignoring the constant terms, the corresponding convex denoising model reads

$$\hat{u} = \arg \min_{u \in BV(\Omega), u > 0} \left\{ \int_{\Omega} u - f \log u dx + \lambda J(u) \right\} \tag{21}$$

The gradient of the data fitting terms in (18) and (21) coincide if we use again the relation $\log \hat{u} = \hat{w}$. Moreover, if we add TV-regularization, then both functions have the same minimizer. Since $\nabla e^w = e^w \nabla w$, for $u = e^w$, we have $\nabla u(x) = 0$ if and only if $\nabla w(x) = 0$. The minimizers \hat{w} and \hat{u} of functions (18) and (21) are unique and given by

$$0 = 1 - f e^{-\hat{w}} - \lambda \operatorname{div} \frac{\nabla \hat{w}}{|\nabla \hat{w}|} \text{ for } |\nabla \hat{w}(x)| \neq 0 \tag{22}$$

$$0 = 1 - \frac{f}{\hat{u}} - \lambda \operatorname{div} \frac{\nabla \hat{u}}{|\nabla \hat{u}|} \text{ for } |\nabla \hat{u}(x)| \neq 0 \tag{23}$$

Since $\frac{\nabla w}{|\nabla w|} = \frac{e^w \nabla w}{e^w |\nabla w|} = \frac{\nabla u}{|\nabla u|}$, we obtain the assertion.

Root and Inverse Transformation-Based Models

(6) m-V model

M -th root transformation was introduced to relax the nonconvexity of the AA model, which was referred to as the m -V model (Yun and Woo 2012). To relax the nonconvexity of the AA-model, they use the m th root transformation ($n_m = \sqrt[m]{n}$, $f_m = \sqrt[m]{f}$, $u_m = \sqrt[m]{u}$). Since the m th root function is monotonically increasing, the gradient operator is applied to m th root transformed images. Then the transformed new variational model is expressed as follows:

$$\begin{aligned}
 u^* &= \arg \min_{u \in \sqrt[m]{U}} \langle m \log u + f u^{-m}, 1 \rangle + \lambda J(u) \\
 \hat{u} &= (u^*)^m
 \end{aligned}
 \tag{24}$$

where $\langle \cdot, \cdot \rangle$ is a usual scalar product in Euclidean spaces, $m \geq 1$. The m -V model can be considered as a generalization of the AA-model (when $m = 1$) and the variational model based on Nakagami distribution (when $m = 2$).

The probability distribution of n_m , which is the m th root of the multiplicative noise n , becomes

$$p(n_m) = \frac{m L^L (n_m)^{mL-1}}{\Gamma(L)} e^{-L(n_m)^m} H(n_m)
 \tag{25}$$

The probability density function (25) is a special case of the generalized Gamma distribution. Hence, the mean value and the variance of n_m are

$$E(n_m) = \frac{\Gamma\left(L + \frac{1}{m}\right)}{\Gamma(L) \sqrt[m]{L}}
 \tag{26}$$

$$var(n_m) = \frac{\Gamma(L) \Gamma\left(L + \frac{2}{m}\right) - \Gamma\left(L + \frac{1}{m}\right)^2}{\Gamma(L)^2 \sqrt[m]{L^2}}
 \tag{27}$$

We know that if $u \in (0, C]$, then the objective function of the m -V model (24) is convex on the set $\left\{ u \mid 0 < u \leq \min \left\{ \sqrt[m]{(m+1)f}, \sqrt[m]{C} \right\} \right\}$. We call this property as conditional convex, which is convex when $m \geq \frac{C}{\min f_j} - 1$.

(7) DZ-model

Since the performance of the m -V model critically depends on the choice of m , a relaxed method was proposed in Kang et al. (2013) to further relax the m -V model. Nevertheless, the method is convex only when m is large enough. In Dong and Zeng (2013), the authors suggested the following model:

$$\min_{u \in \bar{S}(\Omega)} E(u) := \int_{\Omega} \left(\log u + \frac{f}{u} \right) dx + \alpha \int_{\Omega} \left(\sqrt{\frac{u}{f}} - 1 \right)^2 dx + \lambda J(u)
 \tag{28}$$

with the penalty parameter $\alpha > 0$. $\bar{S}(\Omega) := \{v \in BV(\Omega) : v \geq 0\}$ is a closed and convex set, $\log 0 = -\infty$ and $\log \frac{1}{0} = +\infty$ in $\bar{S}(\Omega)$. They proved that if $\alpha \geq \frac{2\sqrt{6}}{9}$, the model (28) is strictly convex.

(8) Exp-model

It was pointed out that the model (28) is mainly suitable for a large value of L . Lu et al. (2016) replace $\sqrt{\frac{u}{f}} - 1$ in the DZ model with $\sqrt{\frac{u}{f}} - \beta 1$, yielding the following optimization problem:

$$\min_{u \in R_+^d} \left\langle \log u + \frac{f}{u}, 1 \right\rangle + \alpha \left\| \sqrt{\frac{u}{f}} - \beta 1 \right\|_2^2 + \lambda J(u) \tag{29}$$

The objective function of this model is strictly convex if $\alpha\beta \geq \frac{2\sqrt{6}}{9}$, where β is no less than 1 and varies with the level of the noise.

Furthermore, owing to the constraint $u > 0$ and the observation that exponent-like models usually provide better quality denoised images than their logarithm-like counterparts, the authors used the log transformation, $w = \log u$, and proposed the following model, called the exp model:

$$\min_{w \in BV(\Omega)} \lambda \int_{\Omega} \left[w + f e^{-w} + \alpha \left(\sqrt{\frac{e^w}{f}} - \beta \right)^2 \right] dx + J(w) \tag{30}$$

The objective function of this model is strictly convex if $\alpha\beta^4 \leq \frac{4096}{27}$.

With a spatially dependent regularization parameter λ , an energy function was presented in Na et al. (2018)

$$\int_{\Omega} w_r(x, y) \left[\frac{f}{\tilde{u}} - \log \frac{f}{\tilde{u}} + \alpha \left(\sqrt{\frac{\tilde{u}}{f}} - \beta \right)^2 \right] (y) dy \tag{31}$$

Next, the log transformation $u = \log(\tilde{u})$ is applied, resulting in

$$S_u^r(x) = \int_{\Omega} w_r(x, y) \bar{q}(u)(y) dy \tag{32}$$

which is the local expected value estimator of the function $\bar{q}(u)$

$$\bar{q}(u) = u + f e^{-u} - \log f + \alpha \left(\sqrt{\frac{e^u}{f}} - \beta 1 \right)^2 \tag{33}$$

Using (32), we can obtain the following TV minimization problem with local constraints:

$$\min_{u \in BV(\Omega)} J(u) = \int_{\Omega} |Du|, \text{ s.t. } S_u^r(x) \leq C \text{ a.e. in } \Omega \tag{34}$$

(9) Convex-model

Another work is the so-called discrete convex model (Zhao et al. 2014)

$$\min_{w, u \in \mathbb{R}^d} \frac{1}{2} \|w - \mu e\|_2^2 + \alpha_1 \|Fw - u\|_1 + \alpha_2 J(u) \tag{35}$$

where μ is a constant, e is a vector of which all the components are valued one, $F = \text{diag}(f)$ is the diagonalization matrix of the noisy image f with main diagonal entries given by f_i , and w is expected as the inverse of the multiplicative noise: $w = \frac{1}{n}$. In fact, from $f = un$ we obtain $fw = u$, and using the matrix form $F = \text{diag}(f)$, we have $Fw = u$. The data fidelity term is $\|Fw - u\|_1$, i.e., $\|\text{diag}(f)w - u\|_1$, which is equivalent to $\|f - un\|_1$. It replaces the nonconvex data fidelity term in the AA model and leads to an unconditional convex problem. Except for the fidelity term and the TV regularization term, the third term $\|w - \mu e\|_2^2$ is introduced to avoid the trivial solution.

(10) m th root transformation model

Based on the statistical analysis of fractional transformation and root transformation, Zhao and Feng first take m th root transformation on the degradation problem $f = un$ (where n obeys the Gamma distribution, set $f_m = \sqrt[m]{f}$, $u_m = \sqrt[m]{u}$ and $\zeta_m = \sqrt[m]{\frac{1}{n}}$), and reformulate the degradation model (Zhao et al. 2018):

$$f_m \zeta_m = u_m \tag{36}$$

Then take L_1 norm $\int_{\Omega} |f_m \zeta_m - u_m| dx$ and TV semi-norm $\int_{\Omega} |\nabla u_m| dx$ as the data fidelity term and the regularization term, respectively, and introduce the quadratic penalty term $\int_{\Omega} (\zeta_m - u_m)^2 dx$ as the prior of noise. Consequently, the proposed model is formulated as (Zhao et al. 2018)

$$\begin{aligned} \text{(a)} \quad & \{\zeta_m^*, u_m^*\} = \arg \min_{\zeta_m, u_m} \int_{\Omega} |f_m \zeta_m - u_m| dx + \frac{\alpha}{2} \int_{\Omega} |\zeta_m - u_m|^2 dx + \lambda \int_{\Omega} |\nabla u_m| dx \\ \text{(b)} \quad & \hat{u} = (u_m^*)^m \end{aligned} \tag{37}$$

where $\{m : m \geq 1, m \in N\}$. α and λ are parameters to control the trade-off among three terms in the objective function. The model is based on the following theorems.

Theorem 1. *Suppose that n follows the Gamma distribution, set $\zeta_m = \frac{1}{\sqrt[m]{n}}$, ($m \geq 1, m \in N$), then*

(i) *The probability density function (PDF) of ζ_m is*

$$p_{\zeta_m}(y) = \frac{L^L m}{\Gamma(L)} y^{-mL-1} e^{-\frac{L}{y^m}} \tag{38}$$

(ii) The means of ζ_m is

$$E(\zeta_m) = \frac{L^{\frac{1}{m}} \Gamma(L - \frac{1}{m})}{\Gamma(L)} \tag{39}$$

and obtain the following trends with fixed $L(L \geq 3)$:

$$\lim_{m \rightarrow +\infty} E(\zeta_m) = 1 \tag{40}$$

$$\lim_{m \rightarrow +\infty} E((\zeta_m - 1)^2) = 0 \tag{41}$$

Theorem 2. Suppose that the random variable n follows Gamma distribution, set $\zeta_m = \frac{1}{n^{\frac{1}{m}}}$, ($m \geq 1, m \in N$); then the KL divergence of ζ_m and $N(\mu_{m,L}, \sigma_{m,L}^2)$ satisfies

$$D_{KL}(\zeta_m \parallel N(\mu_{m,L}, \sigma_{m,L}^2)) = o\left(\frac{1}{m}\right) + o\left(\frac{1}{L^2}\right) \tag{42}$$

where $\mu_{m,L} = E(\zeta_m)$, $\sigma_{m,L}^2 = E((\zeta_m - E(\zeta_m))^2)$.

The proposed model (37) is an unconditional convex problem with a parameter m . It is noted that it reduces to the work in Zhao et al. (2014) when $m = 1$. However, it is known from Fig. 1 that the probability density function of $\zeta = \frac{1}{n}$ ($m = 1$) is far away from the Gaussian distribution, especially for small L . That is to say, the model (Zhao et al. 2014) cannot describe the prior of the noise very well, which

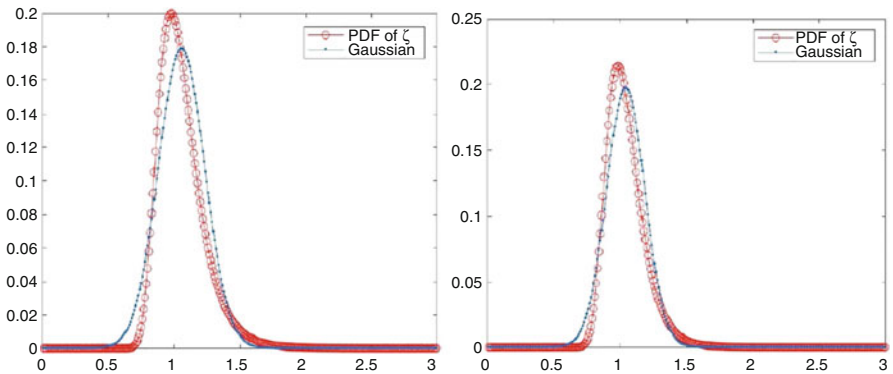


Fig. 1 Plots of the PDFs of ζ_m and $N(\mu_{m,L}, \sigma_{m,L}^2)$ with different m and L (a) $m = 4, L = 3$. (b) $m = 4, L = 4$



Fig. 2 Results of different methods when removing the multiplicative noise with $L = 4$. From the first to the last are original image, noisy image, the restored images of AA, convex, DZ, m -V, and the m th root transformation model, respectively

restricts its denoising performance. Comparatively, the new model is more flexible and extensible.

Moreover, it is worth noting that the data fidelity term in (37) is the L_1 -norm $\int_{\Omega} |f_m \zeta_m - u_m| dx$. The main reason lies in that multiplicative noise mostly presents the corruption as the speckles or outliers onto the image, so L_1 -norm outperforms L_2 -norm or other convex representation as data fidelity term (Zhao et al. 2014) (Fig. 2).

Variational Methods with Different Regularizers

The regularizer $\rho(\cdot)$ has been extensively studied, and there are a few examples widely used in image recovery techniques. The choice of this regularizer depends on the assumptions made about the underlying image structure. Popular choices include the total variation (TV) semi-norm for image gradient sparsity, the l_1 norm for coefficient sparsity in a wavelet basis or other dictionary, and Huber-like functions which are akin to the l_1 norm but smooth. In general, image processors choose a regularizer according to two desiderata: one is that the objective function may be minimized efficiently and the other is that the regularizer accurately reflects image structure. The regularization term can be classified as TV, sparse, and nonconvex regularization.

TV Regularization

A frequently applied regularization term is the total variation (TV) semi-norm suggested in Rudin et al. (1992) by Rudin, Osher, and Fatemi (ROF),

$$|u|_{BV} := \sup_{p \in C_0^1, \|p\|_{\infty} \leq 1} \int_{\Omega} u \operatorname{div} p dx, \text{ which is formally (for sufficiently regular } u)$$

$$J(u) = \int_{\Omega} |\nabla u| dx \tag{43}$$

In the case of additive Gaussian noise, the minimizer \hat{u} of the whole ROF function

$$\frac{1}{2} \int_{\Omega} (f - u)^2 dx + \lambda J(u) \tag{44}$$

has many desirable properties. It preserves important structures such as edges, fulfills a maximum-minimum principle which reads in the discrete n -pixel setting as $f_{\min} \leq \hat{u}_i \leq f_{\max}$, where f_{\min} and f_{\max} denote the minimal and maximal coefficient of f , resp., and preserves the mean value, in the discrete case,

$$\sum_{i=1}^n \hat{u}_i = \sum_{i=1}^n f_i \tag{45}$$

The drawback of the model (44) consists of its staircasing effect so that meanwhile various alternative regularizers were considered.

(1) Non-local TV

The examples given above are standard TV regularization. Non-local TV is a promotion of NL-means. The idea of nonlocal means goes back to Buades et al. (2005) and was incorporated into the variational framework in Gilboa et al. (2006) and Gilboa and Osher (2009). We refer to these papers for further information on NL-means. Based on some pre-computed weights w , the regularization term is given by

$$\rho(u) = \int_{\Omega} |\nabla_w u| dx, \quad |\nabla_w u| := \left(\int_{\Omega} (u(y) - u(x))^2 w(x, y) dy \right)^{1/2}. \tag{46}$$

(2) Weberized TV

Inspiring from the Weberized TV regularization method, a nonconvex Weberized TV regularization-based multiplicative noise removal model was proposed in Xiao et al. (2010):

$$\hat{u} = \arg \min_{u \in X} \left\{ E(u) = \int_{\Omega} \frac{|\nabla u|}{u} dx + \lambda \int_{\Omega} \left(\frac{f}{u} + \log u \right) dx \right\} \tag{47}$$

(3) Modified TV

Another variation of TV is proposed in Hu et al. (2013). When the gradient is small, a log is multiplied, and when the gradient is large, an affine transformation is made. Consider the following variational problem:

$$\min_{u \in X} E(u) := \min_{u \in X} \left\{ \int_{\Omega} \rho(Du) + \lambda \int_{\Omega} \left(\log u + \frac{f}{u} \right) \right\} \tag{48}$$

where $\Omega \subset R^N$ is an open bounded open set with Lipschitz-regular boundary $\partial\Omega$, λ is a constant, $f : \Omega \rightarrow R^+$ is a given function, ρ is an even function from R^N to R having the linear growth

$$\rho(s) = \begin{cases} |s| \log(1 + |s|), & |s| < M \\ b|s| - \frac{M^2}{1+M}, & |s| \geq M \end{cases} \quad (49)$$

where $b = M/(1 + M) + \log(1 + M)$, M is a positive constant, and its value is determined by the size of an image.

(4) TGV

To overcome these staircasing effects, higher-order regularization-based models were suggested in Chambolle and Lions (1997); Chan et al. (2000), and Li et al. (2007). As an early work, an inf-convolution TV (ICTV) model was proposed in Chambolle and Lions (1997), which takes the infimal convolution of TV and second-order TV. Moreover, Li et al. (2007) proposed a denoising model, involving a convex combination of TV and second-order TV as a regularizer. On the other hand, as a generalization of the ICTV, the TGV regularizer was proposed in Bredies et al. (2010). In particular, the second-order TGV is as follows:

$$TGV^2(u) = \min_{p \in P} \int_{\Omega} \alpha_1 |\nabla u - p| + \alpha_0 |\varepsilon(p)| dx \quad (50)$$

where $\varepsilon(p) = \frac{1}{2} (\nabla p + (\nabla p)^T)$ represents the distributional symmetrized derivative, and $\alpha_1, \alpha_0 > 0$ are the weighted parameters that control the balance between the first- and second-order terms. From the formulation (50) of TGV, it can be interpreted that $TGV^2(u)$ can automatically find an appropriate balance between the first- and the second-order derivative of u with respect to α_j .

Sparse Regularization

Due to the sparse nature of the l_1 norm, TV requires the image to have some sparsity in the gradient domain. We know that the wavelet coefficients, ridgelet coefficients, or curvelet coefficients of a sharp image are sparse. Based on these, Durand et al. gave a hybrid method of curvelet field for removing multiplicative noise in Durand et al. (2010).

(5) Curvelet Sparse

Durand et al. considered a hybrid model (DFN) by using the log-image data and a bias correction (Durand et al. 2010). Firstly, they studied the following sparse constraint problem:

$$\hat{\alpha} = \arg \min_{\alpha \in R^d} \left\| W(\log f) - \alpha \right\|_2^2 + \lambda \|\alpha\|_0 \quad (51)$$

where W is the curvelet transform. Secondly, they proposed to minimize a specialized criterion composed of an L_1 data fidelity to $\hat{\alpha}$ and TV regularization in the log-image domain, i.e., the following problem was considered:

$$\hat{x} = \arg \min_{x \in R^d} \left\| \tilde{W}x \right\|_{TV} + \left\| \Lambda(x - \hat{\alpha}) \right\|_1 \quad (52)$$

where \tilde{W} is a left inverse of W , $\Lambda = \text{diag}\{\lambda_{ij}\}$ is some weights. At last step, they restored the image by exponential transformation and bias correction according to the Gamma distribution, e.g.,

$$\hat{u} = \exp(\tilde{W}\hat{x}) \left(1 + \psi_1(L)/2\right) \quad (53)$$

where $\psi_1(z) = \left(\frac{d}{dz}\right)^2 \log \Gamma(z)$, $\Gamma(z) = \int_0^{+\infty} \exp(-t) t^{z-1} dt$. In equation (51), they used curvelet and L_2 loss function to preserve the information of edges. Experimental results in Durand et al. (2010) show that the algorithm can obtain better results than SO (Shi and Osher 2008), AA (Aubert and Aujol 2008), and BS (Chesneau et al. 2010).

(6) Hybrid Model

Hao and Feng introduced dictionary learning instead of curvelet transform (Hao et al. 2012). The authors assumed that the log-image was sparse in the curvelet domain in Durand et al. (2010). However, in practice, it is very difficult to choose the correct dictionary on which the log-image is sparse. So instead of using the pre-selected basis, in the first stage, a dictionary is proposed to train by dictionary learning. Let $\omega : \omega = \log u$, and assume every patch in the log-image can be sparsely represented with the learned dictionary. Hao and Feng propose the following discrete model:

$$\min_{\alpha_{ij}, D, \omega} \left\{ \lambda \sum_{ij} (f \exp(-\omega) + \omega) + \frac{1}{2} \sum_{ij} \|R_{ij}\omega - D\alpha_{ij}\|_2^2 + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 \right\} \quad (54)$$

where λ is a tuning parameter, R_{ij} is an $n \times N$ matrix that extracts the (i, j) th block of size $n \times n$ from the $\sqrt{N} \times \sqrt{N}$ log-image ω , D is a dictionary, α_{ij} is the sparse representation coefficients of the (i, j) th block with dictionary D , μ_{ij} are patch-specific weights, and $\|\alpha_{ij}\|_0$ stands for the count of nonzero entries in α_{ij} . Let ω^* be the minimizer of equation (54). It is amended by L_2 data fidelity and TV regularization in the log-image domain,

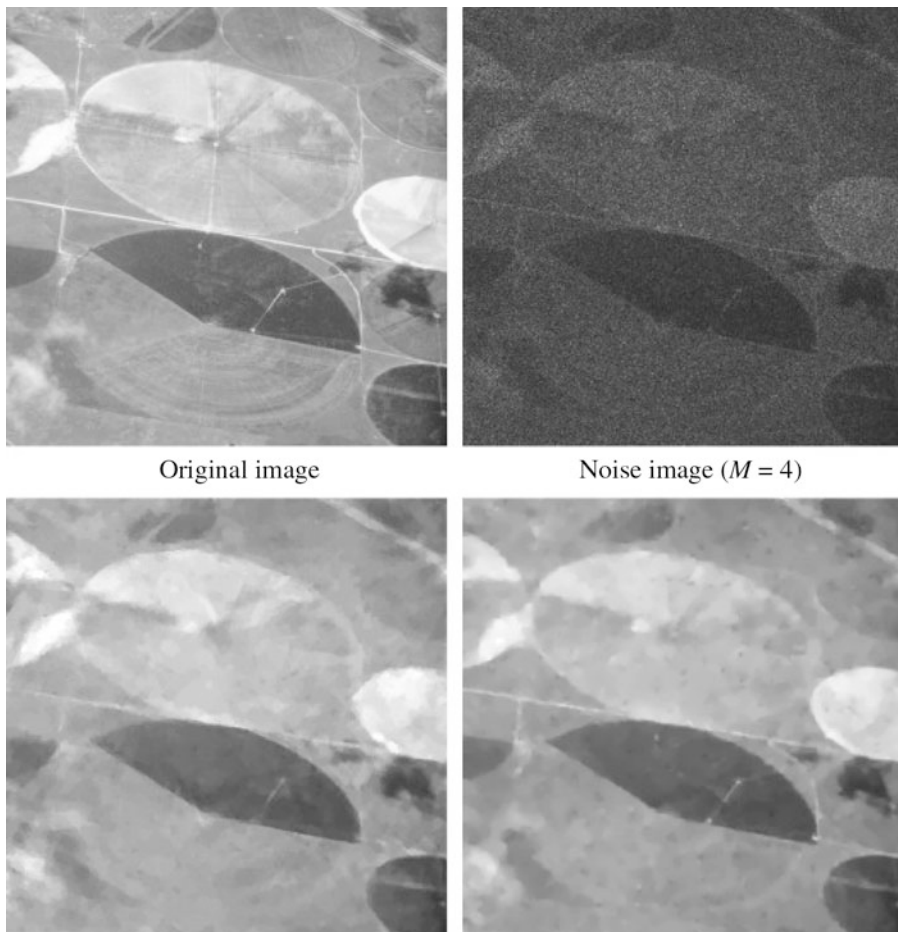


Fig. 3 The denoising experiment on Fields for $L = 4$

$$\min_d \frac{\delta}{2} \|d - \omega^*\|_2^2 + \|d\|_{TV} \tag{55}$$

where δ is a turning parameter.

At the last stage, they transform the result obtained from the second step via an exponential function and bias correction. Let d^* be the solution to (55). d^* can be seen as the estimator of ω^* ; it is prone to bias, which leads to the fact that the restored image is bias too. Using bias correction, we have (Figs. 3 and 4)

$$\hat{u} = \exp(d^*) \left(1 + \psi_1(L)/2\right) \tag{56}$$

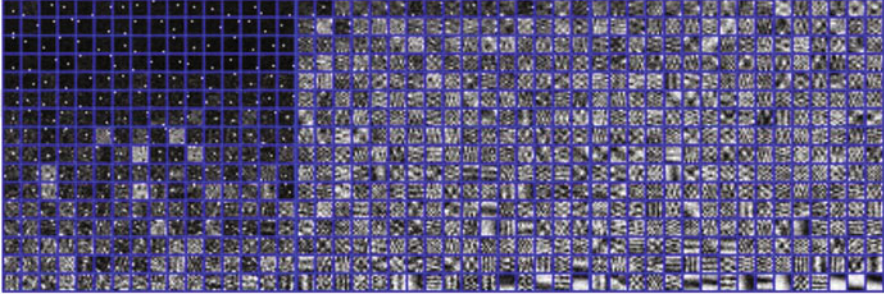


Fig. 4 The trained dictionary on fields for $L = 1, 4, 10$

Differ from this above approach, the following model adds a TV rule for the log domain (Huang et al. 2012).

(7) Dictionary Learning Plus Logarithmic Domain TV

Denoting by 1_Ω the constant 1 over the discrete image domain Ω (a $\sqrt{N} \times \sqrt{N}$ grid) and by $\langle \cdot, \cdot \rangle$ the usual scalar product in Euclidean spaces, the proposed model reads

$$\begin{aligned} \{\hat{D}, \hat{a}_{ij}, \hat{u}\} = \arg \min_{\{D, a_{ij}, u > 0\}} & \lambda \left\langle \log u + \frac{f}{u}, 1_\Omega \right\rangle + \gamma \|\log u\|_{TV} \\ & + \frac{1}{2} \sum_{(i,j) \in P} \|Da_{ij} - R_{ij} \log u\|^2 + \sum_{(i,j) \in P} u_{ij} \|a_{ij}\|_0 \end{aligned} \quad (57)$$

where λ, γ are positive regularization parameters, $P = \{1, 2, \dots, \sqrt{N} - \sqrt{n} + 1\}^2$. $u \in R^N$ is the estimated image. The $\|\cdot\|_{TV}$ term is defined, in the discrete setting, by summing over the image domain Ω the norm of ∇u , the classical 2-neighbors discrete gradient estimate. $R_{i,j} \in R^{n \times N}$ is the matrix corresponding to the extraction of the patch located in (i, j) , and $a_{i,j} \in R^K$ is the sparse vector of coefficients to represent the patch $R_{i,j} \log u$ with the dictionary $D \in R^{n \times K}$. The hidden parameters $(u_{i,j})_{(i,j) \in P}$ are determined by the optimization procedure described in Elad and Aharon (2006).

Nonconvex Regularization

(8) Fractional-Order TV

To enhance the edge-preserving ability of TV, several nonconvex TV regularizers were proposed in Na et al. (2018); Krishnan and Fergus (2009), and Mei et al. (2018), which have the form $\rho(|\nabla u|) = \int_\Omega \varphi(|\nabla u|) dx$, where φ is the nonconvex function defined as

$$\varphi(s) = s^q \quad (0 < q < 1), \quad \frac{\rho s^2}{1 + \rho s^2}, \quad \frac{1}{\rho} (1 + \rho s) \quad (0 < q < 1) \quad (58)$$

Numerical results showed that the nonconvex TV regularizers were better at preserving edges and textures than TV (Nikolova et al. 2010). However, the nonconvex TV regularizers smooth homogeneous regions in the same way as TV. This indicates that they can yield some staircasing artifacts near smooth transition regions in the restored images.

SO model needs finally the nonlinear exponential transformation of the minimizing function, and the Weberized model is strongly dependent on the initialization and the numerical schemes. Tian et al. (2016) generalize the variation order from integer to fraction and obtain a fractional-order I -divergence model as follows:

$$\hat{u} = \arg \min_{u \in BV_\alpha} \left\{ \int_\Omega (u - f \log u) dx + \lambda \int_\Omega |\nabla^\alpha u| dx \right\} \quad (59)$$

where in the corresponding discrete form

$$\begin{cases} \int_\Omega |\nabla^\alpha u| dx = \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} |(\nabla^\alpha u)_{i,j}| \\ |(\nabla^\alpha u)_{i,j}| = \sqrt{((\nabla_1^\alpha u)_{i,j})^2 + ((\nabla_2^\alpha u)_{i,j})^2} \end{cases} \quad (60)$$

Note that the discrete form of the fractional-order gradient $\nabla^\alpha u$ can be evaluated by $(\nabla^\alpha u)_{i,j} = \langle (\nabla_1^\alpha u)_{i,j}, (\nabla_2^\alpha u)_{i,j} \rangle$ with $1 \leq i \leq M, 1 \leq j \leq N$, and

$$\begin{cases} (\nabla_1^\alpha u)_{i,j} = \sum_{k=0}^{K-1} (-1)^k C_k^\alpha u_{i-k,j} \\ (\nabla_2^\alpha u)_{i,j} = \sum_{k=0}^{K-1} (-1)^k C_k^\alpha u_{i,j-k} \end{cases}$$

where $K \geq 3$ is an integer constant, $C_k^\alpha = \Gamma(\alpha + 1) / (\Gamma(k + 1) \Gamma(\alpha - k + 1))$, $\Gamma(\cdot)$ is the gamma function, and u is an image of size $M \times N$.

(9) Nonconvex Sparse Regularizer Model

Following the MAP estimation process, Han and Feng propose a new discrete minimization problem for removing speckle noise (Han et al. 2013):

$$\min_u \left\{ L \sum_{i=1}^n (u_i + e^{f_i - u_i}) + \lambda \sum_{i=1}^n \varphi(|\nabla_i u|) \right\} \quad (61)$$

where $\varphi(s) = \alpha s / (1 + \alpha s)$, and λ is a constant parameter balancing the data term and the regularization term, which are based on the following two points:

Firstly, they point out the advantages of the proposed regularizer in the sparse framework. In fact, both the TV regularizer and the proposed regularizer can be seen as sparse measurements on the gradient modulus of u . The TV regularizer $\sum_{i=1}^n |\nabla_i u|$ is equals to the l_1 -norm of the gradient modulus $\|\nabla_i u\|_1$, while the proposed nonconvex regularizer $\sum_{i=1}^n \varphi(|\nabla_i u|)$ which can be converted into $\sum_{i=1}^n |\nabla_i u| / (|\nabla_i u| + \alpha^{-1})$ tends to be $\|\nabla_i u\|_0$. Note that the parameter α used here should be set large enough. The approximating l_0 -norm is a much sparser measurement than the l_1 -norm. In sparse representation (Daubechies et al. 2010; Candes et al. 2008), the sparse property of the approximating l_0 -norm has been widely used, which will lead to preserving edges of images.

Secondly, they present the underlying reason why the regularizer can protect edges from oversmoothing. This is equivalent to finding out what a good function $\varphi(\cdot)$ should be. On one hand, in order to protect edges from oversmoothing, $\varphi(s)$ should be imposed a “growth” condition of the type $\lim_{s \rightarrow +\infty} \varphi(s) = c$ (c is a constant) so that the contribution of the regularizer would not penalize the formation of strong gradients of u . In other words, the growth condition is used to protect large details of images. On the other hand, at near zero points ($s \rightarrow 0^+$), $\varphi(s)$ is preferable to have the same behavior as the TV regularizer so that u can be better smoothed in homogeneous regions of images. To make a balance between preserving edges and smoothing homogeneous regions, necessarily $\varphi(s)$ should have a nonconvex shape like the type $\varphi(s) = \alpha s / (1 + \alpha s)$. Three different choices of $\varphi(s)$ are shown in Fig. 5. Therefore, the nonconvex sparse regularization is better than convex TV because the TV regularizer does not satisfy the growth condition (Fig. 6).

(10) Nonconvex TGV

Recently, Ochs et al. (2015) proposed a nonconvex extension of the TGV regularizer as follows:

$$NTGV(u) = \min_{p \in P} \int_{\Omega} \alpha_1 \varphi(|\nabla u - p|) + \alpha_0 \varphi(|\varepsilon(p)|) dx \tag{62}$$

where $\varphi(x) = \frac{1}{\rho} \log(1 + \rho x)$ with the parameter $\rho > 0$ controlling the nonconvexity of the regularization term. This regularization takes advantage of both nonconvex regularization and TGV regularization.

The authors propose the following model Na et al. (2018) for the removal of heavy multiplicative noise, which utilizes an NTGV and $\lambda : \Omega \rightarrow R_+$:

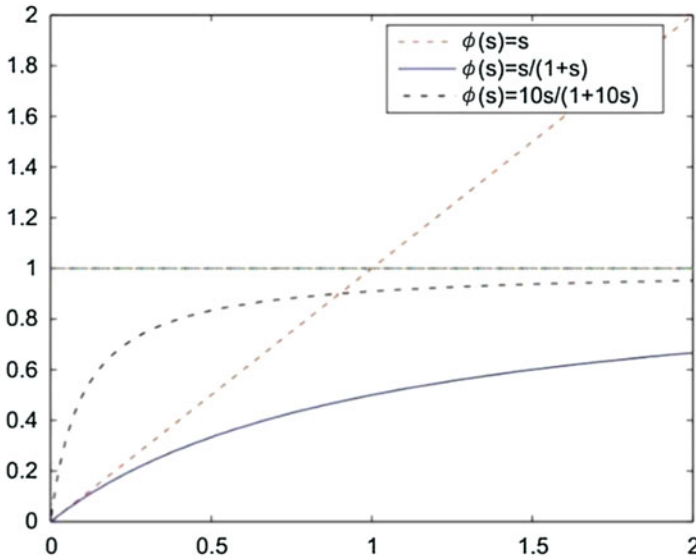


Fig. 5 Nonconvex and convex functions $\varphi(\cdot)$. The nonconvex function $\varphi(s) = s/(1 + s)$ (resp. $\varphi(s) = 10s/(1 + 10s)$) corresponds to $\alpha = 1$ (resp. $\alpha = 10$). Both of their limits are 1 as $s \rightarrow +\infty$. The convex function $\varphi(s) = s$ corresponds to the case of the TV regularizer



Fig. 6 Local enlarged denoising results. From left to right, the clean image, the denoising results of the AA model, the BF model, and the nonconvex sparse regularizer model are listed

$$\min_{u \in X} \int_{\Omega} \lambda(x) \left[u + f e^{-u} + \alpha \left(\sqrt{\frac{e^u}{f}} - \beta 1 \right)^2 \right] dx + NTGV(u), \tag{63}$$

with the NTGV defined as

$$NTGV(u) = \min_{p \in P} \int_{\Omega} \alpha_1 \varphi_1(|\nabla u - p|) + \alpha_0 \varphi_0(|\varepsilon(p)|) dx, \tag{64}$$

where φ_i ($i = 0, 1$) are the nonconvex log functions, $\varphi_i(x) = \frac{1}{\rho_i} \log(1 + \rho_i x)$, where $\rho_i > 0$ control the nonconvexity of regularization terms. The parameters $\alpha > 0$ and $\beta \geq 0$ satisfy the condition $\alpha\beta^4 \leq \frac{4096}{27}$ to enforce the convexity of the data fidelity term. X and P are the corresponding solution spaces.

Multitasks

One of the advantages of using the variational method to build a model is that it can be easily extended to multitasking situations.

Root Transformation

The degraded image f is given by

$$f = (Au) n, \tag{65}$$

where A is a known linear and continuous blurring operator and $n \in L^2(\Omega)$ represents multiplicative noise with mean 1. Here, f is obtained from u , which is blurred by the blurring operator A and then is corrupted by the multiplicative noise n , assuming that $f > 0$. Until the past decade, a few variational methods have been proposed to handle the restoration problem with the multiplicative noise. Given the statistical properties of the multiplicative noise n , in Rudin et al. (2003) the recovery of the image \hat{u} was based on solving the following constrained optimization problem:

$$\begin{aligned} & \min_{u \in S(\Omega)} \int_{\Omega} |Du| \\ & \text{s.t. } \int_{\Omega} \frac{f}{Au} dx = 1, \\ & \int_{\Omega} \left(\frac{f}{Au} - 1 \right)^2 dx = \theta^2, \end{aligned} \tag{66}$$

where θ^2 denotes the variance of n , $S(\Omega) = \{v \in BV(\Omega) : v \geq 0\}$, and $BV(\Omega)$ is the space of functions of bounded variation. In (66), only basic statistical properties, the mean and the variance, of the noise n are considered, which somehow limits the restored results. For this reason, based on the Bayes rule and Gamma distribution with mean 1, by using MAP estimator, Aubert and Aujol (2008) introduced a variational model as follows:

$$\min_{u \in S(\Omega)} \int_{\Omega} \left(\log(Au) + \frac{f}{Au} \right) dx + \lambda \int_{\Omega} |Du| \tag{67}$$

A quadratic penalty term is introduced in (67), which turns out to be

$$\min_{u \in \bar{S}(\Omega)} E_A(u) := \int_{\Omega} \left(\log(Au) + \frac{f}{Au} \right) dx + \alpha \int_{\Omega} \left(\sqrt{\frac{Au}{f}} - 1 \right)^2 dx + \lambda \int_{\Omega} |Du| \tag{68}$$

where $\bar{S}(\Omega) := \{v \in BV(\Omega) : v \geq 0\}$.

Proposition 1. *If $\alpha \geq \frac{2\sqrt{6}}{9}$, then the model (68) is convex.*

Inspired by Dong and Zeng’s model (Dong and Zeng 2013), the following TGV regularized model was presented in Shama et al. (2016)

$$\min_{u \in L^p(\Omega)} E(u) = \int_{\Omega} \left(\log Hu + \frac{f}{Hu} \right) dx + \beta \int_{\Omega} \left(\sqrt{\frac{Hu}{f}} - 1 \right)^2 dx + TGV_{\alpha}^2(u) \tag{69}$$

where $\beta \geq \frac{2\sqrt{6}}{9}$, $p \in (1, \infty)$ and $p \leq d/(d - 1)$, and $d = 2$ for the two-dimensional case.

Fractional Transformation

Zhao et al. (2014) introduced a new convex total variation-based model for restoring images contaminated with multiplicative noise and blur. The main notion is to reformulate a blur and multiplicative noise equation such that both the image variable and noise variable are decoupled. As a result, the concluding energy function involves the total variational filter, the term of the variance of the inverse of noise, the l_1 -norm of the data fidelity term among the observed image, noise, and image variables. The convex optimization model is given by

$$\min_{w, u \in R^d} \frac{1}{2} \|w - \mu e\|_2^2 + \alpha_1 \|Fw - Hu\|_1 + \alpha_2 \|Du\|_2 \tag{70}$$

where α_1 and α_2 are two positive regularization parameters to control the balance between the three terms in the objective function, μ can be set to be the mean value of w , and e is a vector with all entries equal to 1.

Nonlocal Methods

Non-local means (NLM) is an algorithm in image processing for image denoising, which estimates each pixel based on the weighted average of all pixels inside a search window. The weight of a contributing pixel is evaluated on the basis of

“similarity measure” of a neighborhood between the contributing and the target pixels. The NLM algorithm produces Gaussian denoised results with a higher peak signal-to-noise ratio (PSNR) value as well as good perceptual quality. It is natural to extend it to the non-Gaussian noise removal setting (Laus and Steidl 2019).

Indirect Method

In Huang et al. (2017), the Box-Cox transform is used to transform the random variable into an approximately normal distribution, and then the similar block BM3D method is used to denoise. We know Box-Cox transformation (Box and Cox 1964) can effectively transform a random variable and force it to follow normal distribution exactly or approximately if a suitable transformation parameter is selected. Furthermore, BM3D (Dabov et al. 2007) proposed by Dabov et al. is a rather novel method for additive Gaussian white noise removal. Therefore, inspired by the work proposed in Makitalo and Foi (2010, 2014), the authors proposed to transform the multiplicative noise removal to additive Gaussian noise removal by applying the Box-Cox transformation in Huang et al. (2017), and the images are finally recovered by an unbiased denoising algorithm. The Box-Cox transformation parameter is determined through a maximum likelihood method. After applying the Box-Cox transformation to the observed images, the BM3D method is utilized to restore the transformed image, and an unbiased improvement is performed so that the recovered image can finally be obtained.

Applying Box-Cox transformation with parameter λ to each pixel variable of $f=un$ to get

$$f^{(\lambda)} = \frac{(un)^\lambda - 1}{\lambda} \quad (71)$$

Suppose that u and n are independent, the expectation of $f^{(\lambda)}$ reads as

$$E\left(f^{(\lambda)}\right) = E\left(\frac{(un)^\lambda - 1}{\lambda}\right) = \frac{\Gamma(L + \lambda)}{\lambda L^\lambda \Gamma(L)} u^\lambda - \frac{1}{\lambda} \quad (72)$$

If λ is selected appropriately, $f^{(\lambda)}$ should follow or be close to Gaussian distribution, and it can be expressed as

$$f^{(\lambda)} = \frac{\Gamma(L + \lambda)}{\lambda L^\lambda \Gamma(L)} u^\lambda - \frac{1}{\lambda} + \varepsilon \quad (73)$$

where the random variable $\varepsilon \sim N(0, \sigma^2)$ is based on the assumptions in the Box-Cox transformation. In (73), if we consider $\frac{\Gamma(L+\lambda)}{\lambda L^\lambda \Gamma(L)} u^\lambda - \frac{1}{\lambda}$ as the original image and $f^{(\lambda)}$ as the observed image, the additive noise removal methods can be applied

to (73) and a denoised approximation w of $\frac{\Gamma(L+\lambda)}{\lambda L^\lambda \Gamma(L)} u^\lambda - \frac{1}{\lambda}$ can be recovered. Finally, the reconstructed image can be obtained.

$$\hat{u} = L \left(\frac{\Gamma(L) (\lambda w + 1)}{\Gamma(L + \lambda)} \right)^{\frac{1}{\lambda}} \tag{74}$$

Direct Method

If we use nonlocal mean directly, the key is how to correctly estimate similar blocks under multiplicative noise. Now, to measure whether $u_1 = u_2$ by the noisy observations f_1, f_2 , Deledalle et al. (2009) suggest using an approximate

$$s_{DDT}(f_1, f_2) := \int_S p_{f_1|u_1}(f_1|u) p_{f_2|u_2}(f_2|u) du \tag{75}$$

of the conditional density

$$p_{u_1=u_2|(f_1, f_2)}(0|f_1, f_2) = \frac{\int_S p_{u_1}(u) p_{u_2}(u) p_{f_1|u_1}(f_1|u) p_{f_2|u_2}(f_2|u) du}{p_{f_1}(f_1) p_{f_2}(f_2)} \tag{76}$$

as a measure of similarity. s_{DDT} is equal to the NL-mean filter under additive noise.

When the above method is generalized to multiplicative noise, the conditional density

$$\begin{aligned} p_{u_1=u_2|(f_1, f_2)}(0|f_1, f_2) &= \int_S \frac{p_{u_1}(u) p_{u_2}(u)}{p_{f_1}(f_1) p_{f_2}(f_2)} p_{f_1|u_1}(f_1|u) p_{f_2|u_2}(f_2|u) du \\ &= \int_S \frac{p_{u_1}(u) p_{u_2}(u)}{p_{f_1}(f_1) p_{f_2}(f_2)} \frac{1}{u^2} p_{n_1}\left(\frac{f_1}{u}\right) p_{v_2}\left(\frac{f_2}{u}\right) du \end{aligned} \tag{77}$$

is approximated by s_{DDT} .

For multiplicative Gamma noise,

$$\begin{aligned} s_{DDT}(f_1, f_2) &= L \frac{\Gamma(2L-1)}{\Gamma(L)^2} \frac{(f_1 f_2)^{L-1}}{(f_1 + f_2)^{2L-1}} \\ &= L \frac{\Gamma(2L-1)}{\Gamma(L)^2} \frac{1}{f_1 + f_2} \frac{1}{\left(2 + \frac{f_1}{f_2} + \frac{f_2}{f_1}\right)^{L-1}} \end{aligned} \tag{78}$$

However, this measure does not seem to be optimal for multiplicative noise.

Next, a logarithmic transformation is done, and then the approximation of the conditional density is used. Considering the logarithmically transformed random variables $\tilde{f}_i = \ln(f_i)$, where

$$\ln(\tilde{f}_i) = \ln(u_i n_i) = \underbrace{\ln(u_i)}_{\tilde{u}_i} + \underbrace{\ln(n_i)}_{\tilde{n}_i}, \quad i = 1, 2. \tag{79}$$

Lemma 1. For $f_1, f_2 > 0$ with $p_{f_i}(f_i)$ and $S = \text{supp}(p_{\tilde{u}_i})$, it holds that

$$\begin{aligned} & p_{\tilde{u}_1 - \tilde{u}_2}(\tilde{f}_1, \tilde{f}_2) \left(0 | \ln(f_1), \ln(f_2) \right) \\ &= \int_{\tilde{S}} \frac{p_{\tilde{u}_1}(t) p_{\tilde{u}_2}(t) p_{\tilde{f}_1|\tilde{u}_1}(\ln(f_1)|t) p_{\tilde{f}_2|\tilde{u}_2}(\ln(f_2)|t)}{p_{\tilde{f}_1}(\ln(f_1)) p_{\tilde{f}_2}(\ln(f_2))} dt \\ &= p_{\frac{u_1}{u_2}}(f_1, f_2) \left(0 | f_1, f_2 \right) \end{aligned} \tag{80}$$

Suppose $n_i, i = 1, 2$, be Gamma distributed random variables, for $f_1, f_2 > 0$, we can obtain

$$\begin{aligned} s(f_1, f_2) &= \frac{L^{2L}}{\Gamma(L)^2} (f_1 f_2)^L \int_0^{+\infty} \frac{1}{u^{2L+1}} \exp\left(-L \frac{f_1+f_2}{u}\right) du \\ &= \frac{\Gamma(2L)}{\Gamma(L)^2} \frac{(f_1 f_2)^L}{(f_1+f_2)^{2L}} = \frac{\Gamma(2L)}{\Gamma(L)^2} \frac{1}{\left(2 + \frac{f_1}{f_2} + \frac{f_2}{f_1}\right)^L} \end{aligned} \tag{81}$$

which has a maximum of $c = \frac{\Gamma(2L)}{\Gamma(L)^2} \frac{1}{4^L}$.

Then we can use the similarity (81) to calculate the weight function required by the NLM to determine Nonlocal filters for multiplicative Gamma noise.

DNN Method

At present, neural network-based methods have achieved great success in data processing and have also been applied to additive denoising and recovery, such as MLP, CSF, TNRD, and DnCNN (Chen and Pock 2016; Burger et al. 2012; Zhang et al. 2017). This induces us to generalize it to multiplicative noise removal.

Indirect Method

The splitting method is used to solve the variational problem, and one of the subproblems is replaced by DNN, which is essentially a plug-and-play model. Wang et al. (2019) propose a model for general multiplicative noise removal in (82).

$$\begin{aligned} (u^{k+1}, w^{k+1}) &= \arg \min E(u, w) \\ &= \left\{ \int_{\Omega} \left(afe^{-w} + \frac{b}{2}f^2e^{-2w} + cw \right) dx + \frac{\theta_1}{2} \int_{\Omega} (u - e^w - d^k)^2 dx + \lambda \int_{\Omega} \Phi(u) dx \right\} \end{aligned} \quad (82)$$

where θ_1 is the balance parameter. The second term $\int_{\Omega} (u - e^w - d^k)^2 dx$ makes $u = e^w$, and d^k is the Bregman distance. The last term is the deep CNN denoiser prior. Getting the solution of (82) directly is hard because of the term of $\Phi(u)$. Using the split method, we can import auxiliary variable $z = u$. Then (82) can be transformed into (83).

$$\begin{aligned} (u^{k+1}, w^{k+1}, z^{k+1}) &= \arg \min E(u, w, z) \\ &= \left\{ \int_{\Omega} \left(afe^{-w} + \frac{b}{2}f^2e^{-2w} + cw \right) dx + \frac{\theta_1}{2} \int_{\Omega} (u - e^w - d^k)^2 dx \right. \\ &\quad \left. + \lambda \int_{\Omega} \Phi(z) dx + \frac{\theta_2}{2} \int_{\Omega} (z - u)^2 dx \right\} \end{aligned} \quad (83)$$

For Gaussian noise, the parameters can be set as $c = 0, b = 1, a = -1$. Then (83) is changed into the form of (84).

$$\begin{aligned} (u^{k+1}, w^{k+1}, z^{k+1}) &= \arg \min E(u, w, z) \\ &= \left\{ \int_{\Omega} \left(\frac{1}{2}f^2e^{-2w} - fe^{-w} \right) dx + \frac{\theta_1}{2} \int_{\Omega} (u - e^w - d^k)^2 dx \right. \\ &\quad \left. + \lambda \int_{\Omega} \Phi(z) dx + \frac{\theta_2}{2} \int_{\Omega} (z - u)^2 dx \right\} \end{aligned} \quad (84)$$

Each variable will be solved separately after dissociation; by using the alternating optimization strategy, the optimization (84) can be divided into the following subproblems on (u, w, z) :

$$w^{k+1} = \arg \min_w \left\{ E(w) = \int_{\Omega} \left(\frac{1}{2} f^2 e^{-2w} - f e^{-w} \right) dx + \frac{\theta_1}{2} \int_{\Omega} \left(u - e^w - d^k \right)^2 dx \right\} \quad (85)$$

$$z^{k+1} = \arg \min_z \left\{ E(z) = \lambda \int_{\Omega} \Phi(z) dx + \frac{\theta_2}{2} \int_{\Omega} (z - u)^2 dx \right\} \quad (86)$$

$$u^{k+1} = \arg \min_u \left\{ E(u) = \frac{\theta_1}{2} \int_{\Omega} \left(u - e^w - d^k \right)^2 dx + \frac{\theta_2}{2} \int_{\Omega} (z - u)^2 dx \right\} \quad (87)$$

For calculating w , we can deduce the corresponding Euler-Lagrange equation:

$$f e^{-w} - f^2 e^{-2w} - \theta_1 \left(u^{k+1} - e^w - d^k \right) = 0 \quad (88)$$

We can solve w via gradient descent method as

$$w^{k+1} = w^k - \Delta t S^k \quad (89)$$

where Δt represents the time step, and

$$S^k = f e^{-w} - f^2 e^{-2w} - \theta_1 \left(u^{k+1} - e^w - d^k \right) \quad (90)$$

For calculating z , (86) can be changed into (91)

$$\min_z \left\{ E(z) = \int_{\Omega} \Phi(z) dx + \frac{1}{2 \left(\sqrt{\frac{\lambda}{\theta_2}} \right)^2} \int_{\Omega} (z - u)^2 dx \right\} \quad (91)$$

According to Bayesian theory, (91) is the Gaussian denoiser and the noise variance is λ/θ_2 . In Wang et al. (2019), the authors use the CNN Gaussian denoiser for solving (91) by considering the performance and discriminative image prior modeling. The reason for using CNN is that it has achieved great success in Gaussian denoising and better performance (such as PSNR results outperforms BM3D's (Dabov et al. 2007)) than a model-based method. By incorporating CNN Gaussian denoiser into the model, we need not retrain the multiplicative noise removal model for different types of noise. We can deal with different types of noise only by changing the data fidelity.

For solving u , the corresponding Euler-Lagrange equation of (87) is shown in (92).

$$\theta_1 (u - e^w - d^{k+1}) + \theta_2 (u - z) = 0 \quad (92)$$

So, we can get u by using (93).

$$u^{k+1} = \frac{\theta_1 (e^{w^{k+1}} + d^{k+1}) + \theta_2 z^{k+1}}{\theta_1 + \theta_2} \quad (93)$$

The Bregman distance can be expressed as $d^{k+1} = d^k + e^{w^{k+1}} - u^{k+1}$.

Taking into account the above equations, we obtain the complete iteration used in the algorithm for multiplicative noise removal (Fig. 7).

Algorithm of multiplicative noise removal

1. Initialization: $u^0 = f$, $d^0 = 0$, $w^0 = \log u^0$, $k = 0$
2. Repeat
3. Compute w^k using Eq. (89);
4. Compute z^k using Eq. (91);
5. Compute u^k using Eq. (93);
6. Compute Bergman parameter d

using $d^{k+1} = d^k + e^{w^{k+1}} - u^{k+1}$;

1. $k = k + 1$;
 2. Until k achieved the presetting value.
-

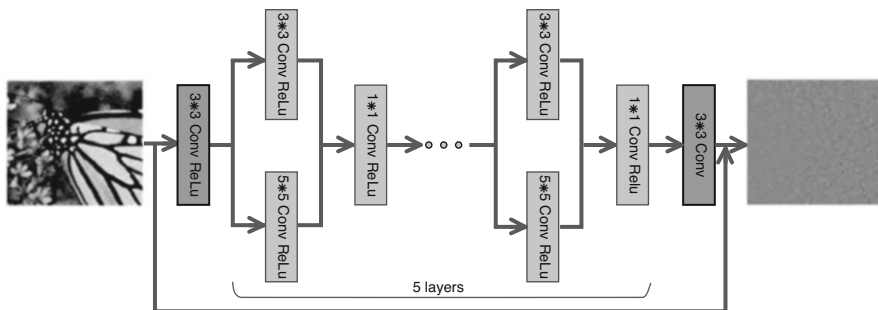


Fig. 7 The architecture of the proposed denoiser network

Direct Method

Let $F \in R^{W \times H}$ be the observed image intensity, $U \in R^{W \times H}$ be the noise free image, and $N \in R^{W \times H}$ be the speckle noise. Then assuming that the SAR image is an average of L looks, the observed image F is related to U by the following multiplicative model (Ulaby et al. 2019)

$$F = UN \tag{94}$$

One common assumption on N is that it follows a Gamma distribution with unit mean and variance $\frac{1}{L}$ and has the following probability density function (Ulaby et al. 2019)

$$p(N) = \frac{1}{\Gamma(L)} L^L N^{L-1} e^{-LN} \tag{95}$$

where $\Gamma(\cdot)$ denotes the Gamma function and $N \geq 0, L \geq 1$

The noise-estimating part of the ID-CNN network consists of eight convolutional layers (along with batch normalization and ReLU activation functions), with appropriate zero-padding to make sure that the output of each layer shares the same dimension with that of the input image (Wang et al. 2017). Each convolutional layer (except for the last convolutional layer) consists of 64 filters with the stride of one. Then the division residual layer with skip connection divides the input image by the estimated speckle noise. A hyperbolic tangent layer is stacked at the end of the network which serves as a nonlinear function. Here, L1 and L8 stand for the sequence of Conv-ReLU layers as depicted in Fig. 8. Similarly, L2 to L7 denote Conv-BNReLU layers. Some estimated results are shown in Table 2.

$$L_E(\varphi_E) = \frac{1}{WH} \sum_{w=1}^W \sum_{h=1}^H \left\| \varphi(F^{w,h}) - U^{w,h} \right\|_2^2 \tag{96}$$

$$L_{TV} = \sum_{w=1}^W \sum_{h=1}^H \sqrt{(\hat{U}^{w+1,h} - \hat{U}^{w,h})^2 + (\hat{U}^{w,h+1} - \hat{U}^{w,h})^2} \tag{97}$$

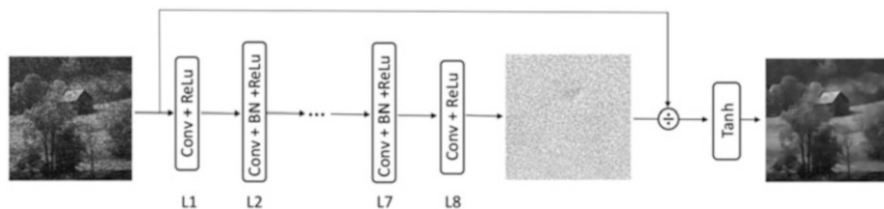


Fig. 8 Proposed ID-CNN network architecture for image despeckling

Table 2 The estimated results on real SAR images

# chip	PPB	SAR-BM3D	CNN	SAR-CNN	ID-CNN
1	42.49	69.26	32.32	50.76	89.43
2	8.63	10.95	7.50	8.93	13.90
3	103.25	127.38	31.65	99.13	193.00
4	34.84	63.83	7.65	43.13	69.40

Finally, the overall loss function is defined as follows:

$$L = L_E + \lambda_{TV} L_{TV} \quad (98)$$

Conclusion

This chapter presents a review of restoration models in the case of multiplicative noise. We introduce the main ideas for multiplicative denoising models and focus on the variational methods with different data fidelity terms, variant methods with different regularizers, multitasks methods, nonlocal methods, and DNN methods. We hope this chapter can provide some help for relevant researchers. We did not give the corresponding optimization method, although it is very important. The complete description probably needs twice as many pages as there are now. The interested reader can refer to the corresponding literature. We think Chambolle and Pock (2016) is a good review article for reference.

References

- Abolhassani, M., Rostami, Y.: Speckle noise reduction by division and digital processing of a hologram. *Optik* **123**(10), 937–939 (2012)
- Aubert, G., Aujol, J.-F.: A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.* **68**(4), 925–946 (2008)
- Box, G.E.P., Cox, D.R.: An analysis of transformations. *J. R. Stat. Soc.: Ser. B (Methodol.)* **26**(2), 211–243 (1964)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
- Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol 2, pp. 60–65. IEEE (2005)
- Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2392–2399. IEEE (2012)
- Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Appl.* **14**(5–6), 877–905 (2008)
- Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numer. Math.* **76**(2), 167–188 (1997)

- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
- Chan, T., Marquina, A., Mulet, P.: High-order total variation-based image restoration. *SIAM J. Sci. Comput.* **22**(2), 503–516 (2000)
- Chatterjee, P., Milanfar, P.: Is denoising dead? *IEEE Trans. Image Process.* **19**(4), 895–911 (2009)
- Chen, D.-Q., Cheng, L.-Z.: Spatially adapted total variation model to remove multiplicative noise. *IEEE Trans. Image Process.* **21**(4), 1650–1662 (2011)
- Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1256–1272 (2016)
- Chesneau, C., Fadili, J., Starck, J.-L.: Stein block thresholding for image denoising. *Appl. Comput. Harmon. Anal.* **28**(1), 67–88 (2010)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Daubechies, I., DeVore, R., Fornasier, M., Güntürk, C.S.: Iteratively reweighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math.: J. Issued Courant Inst. Math. Sci.* **63**(1), 1–38 (2010)
- Deledalle, C.-A., Denis, L., Tupin, F.: Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Process.* **18**(12), 2661–2672 (2009)
- Denis, L., Tupin, F., Darbon, J., Sigelle, M.: SAR image regularization with fast approximate discrete minimization. *IEEE Trans. Image Process.* **18**(7), 1588–1600 (2009)
- Dong, Y., Zeng, T.: A convex variational model for restoring blurred images with multiplicative noise. *SIAM J. Imaging Sci.* **6**(3), 1598–1625 (2013)
- Durand, S., Fadili, J., Nikolova, M.: Multiplicative noise removal using L_1 fidelity on frame coefficients. *J. Math. Imaging Vis.* **36**(3), 201–226 (2010)
- Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **15**(12), 3736–3745 (2006)
- Gilboa, G., Darbon, J., Osher, S., Chan, T.: Nonlocal convex functionals for image regularization. *UCLA CAM-report*, pp. 06–57 (2006)
- Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2009)
- Han, Y., Feng, X.-C., Baciú, G., Wang, W.-W.: Nonconvex sparse regularizer based speckle noise removal. *Pattern Recogn.* **46**(3), 989–1001 (2013)
- Hao, Y., Feng, X., Xu, J.: Multiplicative noise removal via sparse and redundant representations over learned dictionaries and total variation. *Signal Process.* **92**(6), 1536–1549 (2012)
- Hoekman, D.H.: Speckle ensemble statistics of logarithmically scaled data (radar). *IEEE Trans. Geosci. Remote Sens.* **29**(1), 180–182 (1991)
- Hu, X., Wu, Y.H., Li, L.: Analysis of a new variational model for image multiplicative denoising. *J. Inequal. Appl.* **2013**(1), 568 (2013)
- Huang, Y.-M., Moisan, L., Ng, M.K., Zeng, T.: Multiplicative noise removal via a learned dictionary. *IEEE Trans. Image Process.* **21**(11), 4534–4543 (2012)
- Huang, Y.-M., Yan, H.-Y., Zeng, T.: Multiplicative noise removal based on unbiased box-cox transformation. *Commun. Comput. Phys.* **22**(3), 803–828 (2017)
- Jin, Z., Yang, X.: Analysis of a new variational model for multiplicative noise removal. *J. Math. Anal. Appl.* **362**(2), 415–426 (2010)
- Kang, M., Yun, S., Woo, H.: Two-level convex relaxed variational model for multiplicative denoising. *SIAM J. Imaging Sci.* **6**(2), 875–903 (2013)
- Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-Laplacian priors. In: *Advances in Neural Information Processing Systems*, pp. 1033–1041 (2009)
- Laus, F., Steidl, G.: Multivariate myriad filters based on parameter estimation of Student- t distributions. *SIAM J. Imaging Sci.* **12**(4), 1864–1904 (2019)
- Le, T., Vese, L.: Additive and multiplicative piecewise-smooth segmentation models in a variational level set approach. *UCLA CAM Report 03-52*, University of California at Los Angeles, Los Angeles (2003)

- Lebrun, M., Colom, M., Buades, A., Morel, J.-M.: Secrets of image denoising cuisine. *Acta Numer.* **21**, 475 (2012)
- Li, F., Shen, C., Fan, J., Shen, C.: Image restoration combining a total variational filter and a fourth-order filter. *J. Vis. Commun. Image Represent.* **18**(4), 322–330 (2007)
- Lu, J., Shen, L., Xu, C., Xu, Y.: Multiplicative noise removal in imaging: an exp-model and its fixed-point proximity algorithm. *Appl. Comput. Harmon. Anal.* **41**(2), 518–539 (2016)
- Makitalo, M., Foi, A.: Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. *IEEE Trans. Image Process.* **20**(1), 99–109 (2010)
- Makitalo, M., Foi, A.: Noise parameter mismatch in variance stabilization, with an application to Poisson–Gaussian noise estimation. *IEEE Trans. Image Process.* **23**(12), 5348–5359 (2014)
- Mei, J.-J., Dong, Y., Huang, T.-Z., Yin, W.: Cauchy noise removal by nonconvex ADMM with convergence guarantees. *J. Sci. Comput.* **74**(2), 743–766 (2018)
- Na, H., Kang, M., Jung, M., Kang, M.: Nonconvex TGV regularization model for multiplicative noise removal with spatially varying parameters. *Inverse Probl. Imaging* **13**(1), 117 (2018)
- Na, H., Kang, M., Jung, M., Kang, M.: An exp model with spatially adaptive regularization parameters for multiplicative noise removal. *J. Sci. Comput.* **75**(1), 478–509 (2018)
- Nikolova, M., Ng, M.K., Tam, C.-P.: Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. *IEEE Trans. Image Process.* **19**(12), 3073–3088 (2010)
- Ochs, P., Dosovitskiy, A., Brox, T., Pock, T.: On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM J. Imaging Sci.* **8**(1), 331–372 (2015)
- Rudin, L., Lions, P.-L., Osher, S.: Multiplicative denoising and deblurring: theory and algorithms. In: *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pp. 103–119. Springer, New York (2003)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1–4), 259–268 (1992)
- Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. *J. Vis. Commun. Image Represent.* **21**(3), 193–199 (2010)
- Shama, M.-G., Huang, T.-Z., Liu, J., Wang, S.: A convex total generalized variation regularized model for multiplicative noise and blur removal. *Appl. Math. Comput.* **276**, 109–121 (2016)
- Shao, L., Yan, R., Li, X., Liu, Y.: From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms. *IEEE Trans. Cybern.* **44**(7), 1001–1013 (2013)
- Shi, J., Osher, S.: A nonlinear inverse scale space method for a convex multiplicative noise model. *SIAM J. Imaging Sci.* **1**(3), 294–321 (2008)
- Singh, P., Jain, L.: A review on denoising of images under multiplicative noise. *Int. Res. J. Eng. Technol. (IRJET)* **03**(04), 574–579 (2016)
- Steidl, G., Teuber, T.: Removing multiplicative noise by Douglas-Rachford splitting methods. *J. Math. Imaging Vis.* **36**(2), 168–184 (2010)
- Teuber, T., Lang, A.: A new similarity measure for nonlocal filtering in the presence of multiplicative noise. *Comput. Stat. Data Anal.* **56**(12), 3821–3842 (2012)
- Tian, D., Du, Y., Chen, D.: An adaptive fractional-order variation method for multiplicative noise removal. *J. Inf. Sci. Eng.* **32**(3), 747–762 (2016)
- Ulaby, F., Dobson, M.C., Álvarez-Pérez, J.L.: *Handbook of Radar Scattering Statistics for Terrain*. Artech House, Norwood (2019)
- Ullah, A., Chen, W., Khan, M.A.: A new variational approach for restoring images with multiplicative noise. *Comput. Math. Appl.* **71**(10), 2034–2050 (2016)
- Ullah, A., Chen, W., Khan, M.A., Sun, H.: A new variational approach for multiplicative noise and blur removal. *PLoS One* **12**(1), e0161787 (2017)
- Wang, P., Zhang, H., Patel, V.M.: SAR image despeckling using a convolutional neural network. *IEEE Signal Process. Lett.* **24**(12), 1763–1767 (2017)
- Wang, G., Pan, Z., Zhang, Z.: Deep CNN Denoiser prior for multiplicative noise removal. *Multimed. Tools Appl.* **78**(20), 29007–29019 (2019)
- Xiao, L., Huang, L.-L., Wei, Z.-H.: A Weberized total variation regularization-based image multiplicative noise removal algorithm. *EURASIP J. Adv. Signal Process.* **2010**, 1–15 (2010)

- Xie, H., Pierce, L.E., Ulaby, F.T.: Statistical properties of logarithmically transformed speckle. *IEEE Trans. Geosci. Remote Sens.* **40**(3), 721–727 (2002)
- Yun, S., Woo, H.: A new multiplicative denoising variational model based on m th root transformation. *IEEE Trans. Image Process.* **21**(5), 2523–2533 (2012)
- Zhao, X.-L., Wang, F., Ng, M.K.: A new convex optimization model for multiplicative noise and blur removal. *SIAM J. Imaging Sci.* **7**(1), 456–475 (2014)
- Zhao, C.-P., Feng, X.-C., Jia, X.-X., He, R.-Q., Xu, C.: Root-transformation based multiplicative denoising model and its statistical analysis. *Neurocomputing* **275**, 2666–2680 (2018)
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)



Recent Approaches to Metal Artifact Reduction in X-Ray CT Imaging

10

Soomin Jeon and Chang-Ock Lee

Contents

Introduction	348
Background: CT Image Formation and Metal Artifacts	350
Methods	353
Normalized Metal Artifact Reduction (NMAR)	353
Surgery-Based Metal Artifact Reduction (SMAR)	355
Convolutional Neural Network-Based MAR (CNN-MAR)	357
Industrial Application: 3D Cone Beam CT	360
Simulations and Results	365
Simulation Conditions	365
NMAR vs. SMAR: Patient Image Simulations	365
SMAR vs. CNN-MAR	369
NMAR vs. SMAR for 3D CBCT	371
Conclusion	374
References	375

Abstract

Metal artifacts severely degrade image quality by generating streak artifacts in X-ray computed tomography (CT) images. Metal artifact reduction (MAR) has long been an important issue because metal artifacts interfere with the acquisition of accurate contrast images, limiting the various applications of CT imaging. In this

S. Jeon

Department of Radiology, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA

e-mail: sjeon3@mgh.harvard.edu

C.-O. Lee (✉)

Department of Mathematical Sciences, KAIST, Daejeon, Republic of Korea

e-mail: colee@kaist.edu

work, three recently developed CT MAR methods are introduced: normalized MAR, surgery-based MAR, and convolutional neural network-based MAR. Also, a MAR method for industrial cone beam CT is presented as an industrial application.

Keywords

Computed tomography (CT) · Convolutional neural network (CNN) · Normalized metal artifact reduction (NMAR) · Sinogram · Surgery based metal artifact reduction (SMAR)

Introduction

X-ray computed tomography (CT) is one of the most widely used tomographic imaging techniques for non-destructive visualization of structures inside objects. X-ray CT uses radiation from X-rays whose energy is absorbed according to the attenuation coefficients of the tissues in its path (Deans 2007). The cross-sectional image is reconstructed slice by slice from the measured X-ray data at different angles around the scanned object.

X-ray CT produces detailed, high-quality images, and its applicability is promising, but there are various artifacts that severely degrade the quality of CT images: beam hardening artifacts, scattering artifacts, and artifacts due to partial volume effects, photon starvation, undersampling, etc. (Barrett and Keat 2004). Artifacts in CT images are defined as system-induced discrepancies between the reconstructed CT image and the ground truth. These artifacts can be classified according to their causes: (i) physics-based artifacts arising from the physical processes involved during CT data acquisition; (ii) patient-based artifacts caused by factors such as patient movement or the presence of metallic objects in or on the patient; (iii) scanner-based artifacts due to defects in certain scanner functions; and (iv) others such as helical and multi-section artifacts. Among the various causes, implanted metals such as chest screws, dental fillings, and hip prostheses bring the most serious artifacts in CT images. They can also be classified according to their shape as streak artifacts, ring artifacts, cupping artifacts, etc.

The term metal artifact is a generic term for all artifacts caused by metallic objects such as dental implants and surgical clips which lead to various effects such as beam hardening, photon starvation, scattering, and noise increases (Boas and Fleischman 2012). Metal artifacts spread over the entire image in a bright and shadowy crown shape, damaging the quality of CT images and preventing accurate diagnoses. For this reason, as CT imaging becomes more popular, the importance of metal artifact reduction (MAR) technique increases.

Various studies have been attempted to understand metal artifacts, and several approaches have been proposed to reduce them. Existing MAR methods can be roughly classified into three categories: inpainting methods in the projection domain, iterative reconstruction methods, and other methods. For methods based

on projection domain inpainting, sinogram data is calibrated with various types of inpainting techniques such as polynomial interpolation (Abdoli et al. 2010; Kalender et al. 1987; Klotz et al. 1990; Mahnken et al. 2003; Wei et al. 2004), wavelets (Zhao et al. 2002), Euler's elastica model (Gu et al. 2006), interpolation using adjacent pixel values (Kim et al. 2010), and forward projection (Bal and Spies 2006; Prell et al. 2009). However, these methods generate additional artifacts in the reconstructed CT image due to the inconsistency of the calibrated sinogram after inpainting. These extra artifacts deteriorate the quality of the X-ray CT image. In iterative reconstruction methods, the image is updated in a feedback manner through forward projection and back projection, for example, adding physics knowledge such as acquisition process and photon statistics (De Man et al. 2001; Kano and Koseki 2016; Lemmens et al. 2009; Wang et al. 1996). Iterative reconstruction methods achieve better image quality than inpainting-based methods. However, they are usually much slower due to the high computational cost. The category of other methods includes filtering methods (Bal et al. 2005; Kachelrieß et al. 2001), methods based on the wave front set (Park et al. 2016), and those with total variation minimization (Verburg and Seco 2012). There are also hybrid methods that combine different MAR methods (Watzke and Kalender 2004; Zhang et al. 2013). The first clinical application of the iterative MAR algorithm was achieved by Philips Health Care (Philips Healthcare 2012) though it was only applied to orthopedic implant cases (Zhang et al. 2020). The algorithm used by Phillips Health Care was based on the work by Timmer and Koehler (Koehler et al. 2012; Timmer 2008), whose methods were applied experimentally only for simple-shaped phantoms, with the result showing that residual artifacts still existed.

One of the state-of-the-art MAR algorithms is the normalized MAR (NMAR) algorithm (Meyer et al. 2010). It inpaints the corrupted part of the sinogram using normalization technique in conjunction with the prior image. In the first step, the metal is segmented in the image domain by a threshold. The forward projection then identifies the metal traces in the original projection. Before interpolation, the projection data is normalized based on the forward projection of the prior image. The prior image is acquired, for example, by a multi-threshold segmentation of the initial image. The original raw data is divided by the projection data of the prior image and then denormalized after interpolation.

Recently, a new metal artifact reduction algorithm based on sinogram surgery has been proposed to reduce metal artifacts without additional ones (Jeon and Lee 2018). The area around the metal region with similar CT numbers is extracted using the reconstructed CT numbers from the given sinogram. Then, the metal region and its surroundings are filled with the average CT number of the surrounding area to obtain a modified CT image. Using the forward projection of the modified CT image, a sinogram containing information about the anatomical structure is generated, and the sinogram surgery is performed using this and then back-projected to regenerate the CT image. The reconstructed CT image contains structural information around the metal region even if the original CT image includes severe artifacts near metallic objects. Unlike other interpolation-based MAR methods, the proposed algorithm uses this structural information to correct the corruption in

the sinogram. The sinogram completion process is iteratively performed using the basic principles of CT image reconstruction to remove the metal effect from the sinogram.

Meanwhile, attempts are underway to exploit deep learning in almost all fields of science and technology. In particular, the concept of deep learning was also introduced to MAR in Ghani and Karl (2020), Gjestebj et al. (2017), Hwang et al. (2018), and Zhang and Yu (2018). Among these, the convolutional neural network (CNN)-based MAR (CNN-MAR) method (Zhang and Yu 2018) is best known as a general open framework. The CNN-MAR method consists of two phases: CNN phase and surgery phase with a prior image. In the CNN phase, CNN is used as an information fusion tool to produce a reduced artifact image by combining the uncorrected CT image and two pre-corrected ones from some model-based MAR methods as the input data of the neural network. The surgery phase further reduces the remaining artifacts by adding seamless surgery process with a prior image based on tissue classification.

This work introduces the NMAR algorithm (Meyer et al. 2010), the surgery-based MAR (SMAR) algorithm (Jeon and Lee 2018), and the CNN-MAR methods (Zhang and Yu 2018). It also reviews a methodology for reducing metal artifacts in three-dimensional industrial cone beam CT systems (Jeon et al. 2021).

Background: CT Image Formation and Metal Artifacts

In an X-ray CT system, the X-ray source and detector rotate simultaneously at regular angular intervals. A single projection data is obtained for each angle, and a stack of these projection data is called a sinogram. A cross-sectional image of an object can be obtained from the sinogram through a reconstruction process called filtered back projection (FBP). Depending on the geometry of photon spread, there are different types of X-ray CT systems, such as parallel beam CT, fan beam CT, cone beam CT, and helical CT. This work assumes using a parallel beam CT.

Let $f_E(\mathbf{x})$ denote the X-ray attenuation coefficient at a point \mathbf{x} when the X-ray energy level is E . The Beer-Lambert law (Klotz et al. 1990) describes the attenuation of an X-ray along the path through which the X-ray passes a physical substance composed of a single species of uniform concentration by the first-order ordinary differential equation

$$\frac{dI}{dt}(\mathbf{x}) = -f_E(\mathbf{x})I(\mathbf{x}), \quad \mathbf{x} = s\Theta + t\Theta^\perp, \quad \Theta = (\cos \theta, \sin \theta),$$

where I is the intensity of X-ray, s the distance along the detector, and t the distance along the path of X-ray; see Fig. 1. Solving the above equation gives the formula for I at the detector:

$$I_\theta(E, s) = I_0(E)e^{-\mathcal{R}_\theta f_E(s)},$$

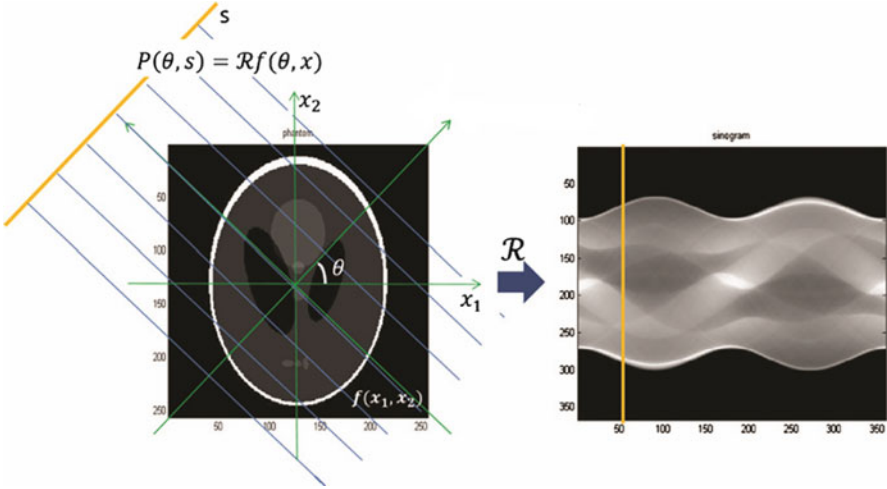


Fig. 1 Illustration of the Radon transform and the sinogram. (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

where $I_0(E)$ is the initial intensity of the X-ray with energy level E . Here, the projection data $\mathcal{R}_\theta f_E(s)$ is the Radon transform of f_E defined by

$$\mathcal{R}_\theta f_E(s) := \int_{\mathbb{R}^2} f_E(\mathbf{x}) \delta(\Theta \cdot \mathbf{x} - s) d\mathbf{x},$$

which means the integral along the line

$$L_{\theta,s} := \left\{ \mathbf{x} \in \mathbb{R}^2 : \Theta \cdot \mathbf{x} = s, \Theta = (\cos \theta, \sin \theta) \right\},$$

where δ is the Dirac delta function.

Since the sinogram is a stack of projection data, it can be expressed as $\mathcal{R}f_E(\theta, s) = [\mathcal{R}_\theta f_E(s)]_\theta := [\mathcal{R}_{\theta_1} f_E(s), \mathcal{R}_{\theta_2} f_E(s), \dots]$; see Fig. 1. Assuming that the X-ray is monochromatic, there is a relation

$$\mathcal{R}f_E(\theta, s) = \left[-\ln \left(\frac{I_\theta(E, s)}{I_0(E)} \right) \right]_\theta.$$

Then, a two-dimensional X-ray CT image is reconstructed by the inverse Radon transform of the sinogram $\mathcal{R}f_E$:

$$\begin{aligned} f_E(\mathbf{x}) &= \mathcal{R}^{-1} \{ \mathcal{R}f_E \} \\ &= \frac{1}{4\pi^2} \int_0^\pi \int_{-\infty}^\infty |\omega| \mathcal{F}_s[\mathcal{R}f_E(\theta, s)](\omega) e^{i\omega \mathbf{x} \cdot \Theta} d\omega d\theta, \end{aligned} \tag{1}$$

where \mathcal{F}_s is the 1D Fourier transform with respect to s .

In practice, since X-ray CT uses polychromatic X-rays, the measured X-ray intensity is given by

$$I_\theta(s) = \int_{E_{\min}}^{E_{\max}} I_\theta(E, s) dE = \int_{E_{\min}}^{E_{\max}} I_0(E) e^{-\mathcal{R}_\theta f_E(s)} dE, \tag{2}$$

where E_{\min} and E_{\max} are the minimum and maximum energy levels of the X-ray, respectively. Then the sinogram $\mathcal{P}f_E$ is given by

$$\mathcal{P}f_E = [\mathcal{P}_\theta f_E]_\theta,$$

for

$$\mathcal{P}_\theta f_E(s) = -\ln\left(\frac{I_\theta(s)}{I_0}\right), \tag{3}$$

where $I_0 = \int_{E_{\min}}^{E_{\max}} I_0(E) dE$. Then, the CT image is reconstructed from the sinogram using (1) with $\mathcal{P}f_E$ instead of $\mathcal{R}f_E$. The CT image reconstruction is shown in Fig. 2.

A smooth function is called a Schwartz function if all its derivatives including itself decay at infinity faster than the inverse of any polynomial. A function $g(\theta, s)$ defined on $[0, 2\pi) \times \mathbb{R}$ is said to satisfy the homogeneous polynomial condition if for $k = 1, 2, \dots$, the integral

$$\int_{\mathbb{R}} g(\theta, s) s^k ds \tag{4}$$

can be written as a k -th degree homogeneous polynomial in $\Theta = (\cos \theta, \sin \theta)$. By the Schwartz theorem for the Radon transform (Helgason 1965), $g = \mathcal{R}f$ for

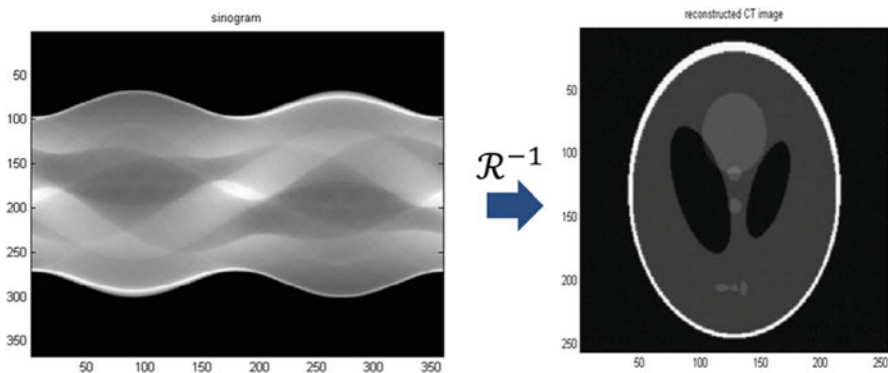


Fig. 2 Illustration of the reconstruction of CT image from sinogram with inverse Radon transform

some Schwartz function f if and only if g satisfies the homogeneous polynomial condition. In particular, when $k = 0$, (4) produces the same value regardless of θ for all sinograms generated by a single-energy X-ray CT machine. This is called the consistency condition of the sinogram.

Metal artifacts are mainly due to the beam hardening effect of polychromatic X-ray beam. When a polychromatic X-ray beam passes through an object, low-energy photons are preferentially absorbed, and thus the mean energy gradually increases (we say that the beam is hardened). The harder the beam, the less it attenuates. Therefore, the total attenuation is no longer proportional to the object thickness, unlike the monochromatic X-ray case. Hence, the generated sinogram becomes inconsistent. Because a monochromatic X-ray is assumed in the reconstruction (1), artifacts occur when a CT image is reconstructed from an inconsistent sinogram. If the target is a non-metal such as human tissue, these artifacts are not significant because the energy dependence of the attenuation coefficient is not high. However, the X-ray attenuation coefficient of materials such as metal with a high CT number is extremely energy dependent and produces erroneous projection data that is the source of metal artifacts.

Methods

Normalized Metal Artifact Reduction (NMAR)

Inpainting of Metal Traces in the Normalized Sinogram

The simplest inpainting-based MAR method is the LI method (Kalender et al. 1987) that fills the metal trace in the uncorrected sinogram by the linear interpolation of its neighboring unaffected projections in each projection view. In fact, the projection image called sinogram is made along the sine curve. Therefore, since the sinogram after interpolation is not consistent and even not smooth at the boundaries of the metal traces, new artifacts are necessarily introduced, and the structure near metals is distorted. However, such interpolation is less problematic in homogeneous regions, as interpolation on nearly flat sinogram provides a certain level of smoothness at the boundaries of the metal traces. The idea of normalization is to transform the sinogram so that it is comparatively flat.

Here, as a way to transform a sinogram into a more flat one, the method in Müller and Buzug (2009) is introduced. In the first step, the metal trace is determined by the forward projection of the metal extracted by the thresholding from the uncorrected CT image. A normalized sinogram is then created by dividing each pixel value of the given sinogram by the thickness of the object that the X-ray passes through. The metal trace determines where in the normalized sinogram is replaced by the inpainting (e.g., simple linear interpolation per projection view (Kalender et al. 1987)). Subsequently, the corrected sinogram is obtained by denormalizing the interpolated sinogram by multiplying it by the thickness of the object. Reconstructing a CT image with this corrected sinogram produces the corrected image.

This method gives excellent results in the absence of high contrast. If bones or metals are present, normalization with thickness cannot produce a very flat sinogram, resulting in new artifacts. To extend this idea to objects composed of bones, metals, and other high-contrast materials, NMAR uses a prior image that takes these materials into account. Through denormalization, NMAR restores traces of high-contrast objects buried in metal shadows. This is because the shape information of these objects is contained in the sinogram of the prior image. NMAR ensures a certain level of smoothness at the boundaries of the metal traces in the corrected sinogram and recovers traces of objects contained in the prior image.

NMAR Algorithm

Figure 3 provides a diagram of the different steps of NMAR algorithm. An uncorrected image is reconstructed from the original sinogram p . The metal image is then obtained by thresholding. The prior image f^{prior} is created by segmenting soft tissues and bones. Forward projection produces the corresponding sinograms. The original sinogram p is then normalized by division by p^{prior} projected from f^{prior} . The division is only performed on pixels where the divisor is greater than a

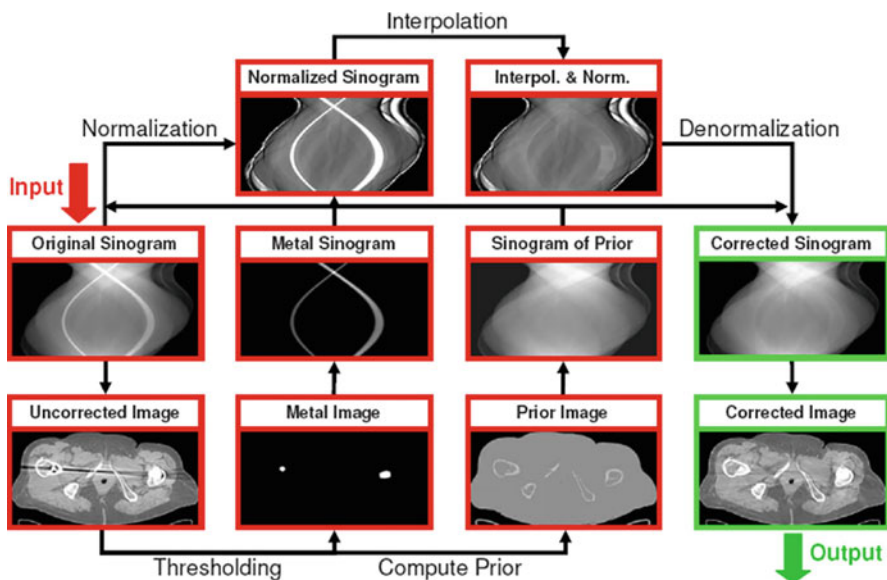


Fig. 3 Scheme of NMAR algorithm – from the original sinogram, an uncorrected image is reconstructed. By thresholding, the metal image and the prior image are obtained. Forward projection yields the corresponding sinograms. The normalized sinogram is then obtained by dividing the original sinogram by the sinogram of the prior image. The metal projections determine where data in the normalized sinogram are replaced by interpolation. The interpolated and normalized sinogram is denormalized by multiplying it with the sinogram of the prior image. Reconstruction yields the corrected image. (Reprinted from Meyer et al. (2010) with permission from John Wiley and Sons)

small positive value to avoid division by zero. A simple interpolation operation \mathcal{M}_{int} is performed on the normalized sinogram p^{norm} to obtain a sinogram with metal traces removed. Subsequently, the corrected sinogram p^{corr} is obtained through denormalization, which multiplies $\mathcal{M}_{\text{int}}p^{\text{norm}}$ by p^{prior} :

$$\begin{aligned} p^{\text{prior}} &= \mathcal{R}f^{\text{prior}}, \\ p^{\text{norm}} &= \frac{p}{p^{\text{prior}}}, \\ p^{\text{corr}} &= p^{\text{prior}} \mathcal{M}_{\text{int}}p^{\text{norm}}. \end{aligned}$$

In this step, the structure information from the prior image is brought back into the metal trace because traces of high-contrast objects are included in the sinogram of the prior image. Normalization and multiplication procedures ensure that there is no difference between the original sinogram and corrected sinogram, except for metal trace. Hence, only sinogram values around metal traces are needed for normalization and denormalization. After reconstruction, the metal is inserted back into the corrected image.

An important step in NMAR algorithm is finding a good prior image. It should be modeled as close as possible to the uncorrected image, but should not contain artifacts. To achieve this, it is necessary to identify air regions, soft tissue regions, and bone regions. After smoothing the image with Gaussian, simple thresholding can be applied to segment air, soft tissue, and bone. It is also useful to smooth the streak structure as described in Müller and Buzug (2009) to reduce streak artifacts before segmentation. See Meyer et al. (2010) for more details.

Surgery-Based Metal Artifact Reduction (SMAR)

Even though NMAR algorithm removes metal artifacts very well, it still generates streaking artifacts because the corrected sinogram is not consistent. Recently, a new metal artifact reduction algorithm called SMAR, based on sinogram surgery, was proposed to reduce metal artifacts by calibrating the sinogram to be nearly consistent (Jeon and Lee 2018).

SMAR algorithm consists of two steps: a preprocessing step and an iterative reconstruction step. In the preprocessing step, the metal part from the given CT image is extracted, and then its metal trace is determined by the forward projection as in the NMAR algorithm. In the iterative reconstruction step, in order to moderate metal artifacts, several processes are performed such as average fill-in, sinogram surgery, and reconstruction from the updated sinogram. Detailed descriptions of each of these are given below.

Preprocessing Step

(1) Metal extraction: The metal region M can be extracted by simple thresholding.

- (2) Surgery region designation: Once the metal region M has been extracted, its forward projection using the Radon transform \mathcal{R} establishes the surgery region

$$M_{\text{proj}} = \text{supp}\{\mathcal{R}\chi_M\},$$

where χ is the characteristic function

$$\chi_M(x) = \begin{cases} 1 & \text{for } x \in M, \\ 0 & \text{otherwise.} \end{cases}$$

This region coincides with the corrupted part of the sinogram due to metal.

Iterative Reconstruction Step

The iterative reconstruction step has three steps: average fill-in, sinogram surgery, and reconstruction of the updated sinogram.

- (1) Average fill-in: For the reconstructed CT image from the previous step, $f^{(n-1)}$, a connected region C is segmented which is surrounding M . Using $v^{(n-1)}$, the average of the attenuation coefficients $f^{(n-1)}$ of the region C , the average fill-in step is evaluated as

$$\tilde{f}^{(n-1)} = v^{(n-1)}\chi_{C \cup M} + f^{(n-1)}(1 - \chi_{C \cup M}),$$

which moderates the streak structure of the CT image.

- (2) Projection and sinogram surgery: By forward projection of $\tilde{f}^{(n-1)}$, a new sinogram

$$\tilde{p}^{(n-1)} = \mathcal{R}\tilde{f}^{(n-1)}$$

is obtained. Using $\tilde{p}^{(n-1)}$ a new sinogram

$$p^{(n)} = \tilde{p}^{(n-1)}\chi_{M_{\text{proj}}} + p^{(0)}(1 - \chi_{M_{\text{proj}}}).$$

is produced. This is referred to as a sinogram surgery, where the corrupted sinogram part of a given sinogram $p^{(0)}$ is replaced with the newly generated sinogram $\tilde{p}^{(n-1)}$ which is generated from the CT image with moderate metal artifact structure.

- (3) CT image reconstruction: The new CT image is reconstructed by FBP

$$f^{(n)} = \mathcal{R}^{-1}p^{(n)}$$

The resulting image has less streak artifacts compared to $f^{(n-1)}$. Here, other sophisticated reconstruction methods can be also applied for the image quality improvement.

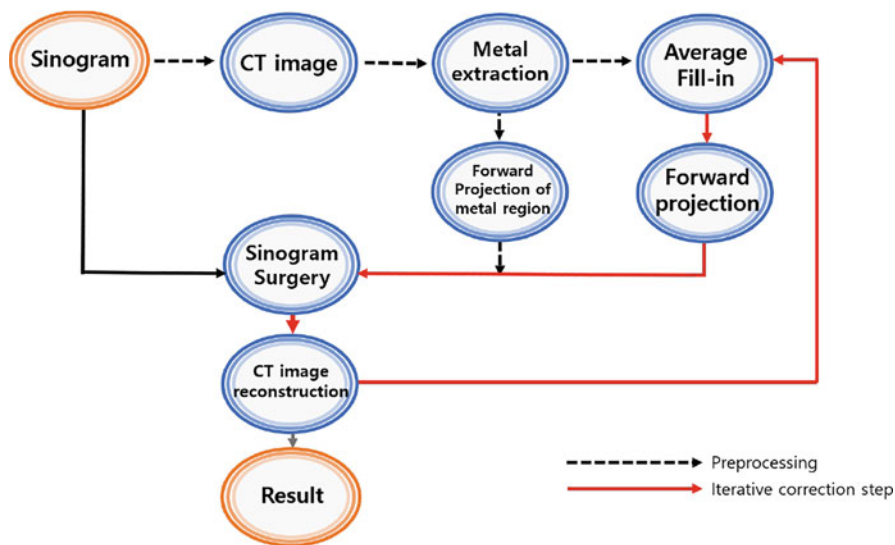


Fig. 4 Diagram for schematic description of SMAR algorithm

As the iterative reconstruction step is repeated, streak artifacts are reduced gradually because the missing data is complementarily replaced for both the sinogram and the reconstructed CT image. The iterative reconstruction step is terminated when the relative difference between the sinogram data becomes less than the tolerance level. The convergence of the SMAR algorithm is given empirically in the Appendix of Jeon and Lee (2018).

Figure 4 provides a schematic diagram of SMAR algorithm.

Convolutional Neural Network-Based MAR (CNN-MAR)

In this section, CNN-MAR method (Zhang and Yu 2018) is introduced as one of the most successful deep learning algorithms for metal artifact reduction. The CNN-MAR method uses two pre-corrected auxiliary images from the BHC method and the LI method, which were used in the hybrid MAR method in Zhang et al. (2013). BHC method is a model-based reconstruction method using total variation minimization (Verburg and Seco 2012). In this work, CNN-MAR methods with BHC and LI methods and with SMAR and LI methods are considered.

Training of the Convolutional Neural Network

The main goal of CNN training is to find optimal parameters that minimize the loss function. From a metal-free reference image, a set of images is generated: an image with metal inserted; an image with metal artifacts, which is used as a raw image;

and two pre-corrected auxiliary images. The loss function $Loss: \{\mathbf{U}, \mathbf{V}, W\} \rightarrow \mathbb{R}$ is defined by

$$Loss = \frac{1}{N} \sum_{n=1}^N \|C_L(\mathbf{u}_n, W) - \mathbf{v}_n\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and N is the number of input data, $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$ the input data where each \mathbf{u}_i consists of a raw image and two auxiliary images, $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$ a target data of reference images, and C_L a CNN containing a parameter set W . To optimize the loss function, stochastic gradient descent with momentum (SGDM) is used. SGDM is a variant of stochastic gradient descent (SGD) by adding momentum to accelerate the SGD algorithm which updates parameters randomly in order to avoid the situation “trap in local minima.” SGD is based on traditional gradient descent (GD) algorithm. SGDM is formulated as

$$\begin{aligned} \Delta W^{(k)} &= \mu \Delta W^{(k-1)} - \alpha \nabla Loss(W^{(k)}), \\ W^{(k+1)} &= W^{(k)} + \Delta W^{(k)}, \end{aligned}$$

where μ is a momentum value ($= 0.9$), ΔW the direction vector, W the parameters, and α the learning rate and $\nabla Loss$ denotes the stochastic gradient. Therefore, ΔW is updated with remembering the past directions. Thanks to the momentum term, it is expected that the probability that W is trapped in local minima is reduced.

As in Zhang and Yu (2018), the CNN is constructed as follows: L convolutional layers are used with $\text{ReLU}(x) = \max(0, x)$ for nonlinear activation function. The first $L - 1$ layers are formulated as

$$\begin{aligned} C_0(\mathbf{u}) &= \mathbf{u}_0, \\ C_m(\mathbf{u}) &= \text{ReLU}(\mathbf{W}_m * C_{m-1}(\mathbf{u}) + \mathbf{b}_m), \quad m = 1, \dots, L - 1, \\ C_L(\mathbf{u}) &= \mathbf{W}_L * C_{L-1}(\mathbf{u}) + \mathbf{b}_L, \end{aligned}$$

where $*$ stands for convolution, \mathbf{W}_m is the m -th kernel, and \mathbf{b}_m is the bias in the m -th layer. Each layer consists of 32 channels. The last layer generates an image that is close to the target. The convolutional kernel is 3×3 in each layer. Zero padding is used in each layer to maintain the size of output data as the same as input data. Whole architecture of the CNN is shown in Fig. 5.

CNN-MAR Method

The CNN-MAR algorithm consists of the following five steps:

- (1) Two auxiliary MAR images are obtained.
- (2) A CT image is obtained with reduced artifacts by the trained CNN.

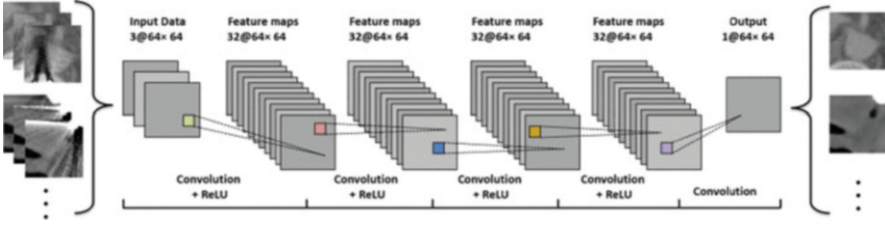


Fig. 5 Architecture of the CNN for metal artifact reduction. (Reprinted from Zhang and Yu (2018) with permission from IEEE)

- (3) A CNN prior image is generated using tissue processing.
- (4) A corrected sinogram is produced by replacing the metal traces in the sinogram of the CNN image using the sinogram of the CNN prior image.
- (5) A corrected CT image is obtained using the inverse Radon transform.

Here, the details of Steps 2–4 are provided.

First, from the uncorrected image, two corrected auxiliary MAR images are obtained by the BHC and LI methods or by the SMAR and LI methods. Then these are combined as a three-channel image $\mathbf{u}^{\text{input}}$, and the CNN-corrected image f^{CNN} is obtained through the CNN processing:

$$f^{\text{CNN}} = C_L(\mathbf{u}^{\text{input}}).$$

Here, the parameters in C_m have been found in advance from the CNN training. In implementation, $L = 5$ was used.

Even after the CNN processing, f^{CNN} still has considerable artifacts. Therefore, additional process is applied to reduce these artifacts; a prior image is generated from f^{CNN} by the tissue processing in Zhang and Yu (2018). First, because the water-equivalent tissues have similar attenuations and are accounted for a dominant proportion in a patient, the pixels corresponding to these tissues are assigned uniform values. For simple calculation, it is assumed that f^{CNN} consists of bone, water, and air. Using the k -means clustering on f^{CNN} , two thresholds are determined; one threshold is the bone-water threshold, and the other is the water-air threshold. Then, a binary image B is obtained with the water region set to 1 and the rest set to 0.

To replace the metal trace of the sinogram, a distance image D is introduced, which is made from the binary image B as follows. The pixel value of D is set to the distance between the pixel and its nearest 0 pixel if it is not greater than 5 and set to 5 if it is greater than 5. Hence, in the image $D = \{D_i\}$, most of the water pixels have the value 5, and there are 5-pixel transition regions, while the other pixels are zero. We compute the weighted average of the water pixel values:

$$\bar{f}^{\text{water}} = \frac{\sum_i D_i f_i^{\text{CNN}}}{\sum_i D_i}.$$

Then, the prior image is obtained:

$$f_i^{\text{prior}} = \frac{D_i}{5} \bar{f}^{\text{water}} + \left(1 - \frac{D_i}{5}\right) f_i^{\text{CNN}}.$$

This prior image $f^{\text{prior}} = \{f_i^{\text{prior}}\}$ is smoother than f^{CNN} . Using the prior image, sinogram correction and image reconstruction are performed as follows. First, let the metal trace occupy from the $(j_n + 1)$ -th pixel to the $(j_n + \Delta_n)$ -th pixel in the n -th projection view according to θ . Then the metal trace is replaced by the following:

$$p_{\theta, k_n}^{\text{corr}} = \frac{\left(\mathcal{R}_\theta f_{j_n + \Delta_n + 1}^{\text{CNN}} - \mathcal{R}_\theta f_{j_n + \Delta_n + 1}^{\text{prior}}\right) - \left(\mathcal{R}_\theta f_{j_n}^{\text{CNN}} - \mathcal{R}_\theta f_{j_n}^{\text{prior}}\right)}{\Delta_n + 1} (k_n - j_n) \\ + \mathcal{R}_\theta f_{k_n}^{\text{prior}} + \left(\mathcal{R}_\theta f_{j_n}^{\text{CNN}} - \mathcal{R}_\theta f_{j_n}^{\text{prior}}\right), \quad j_n \leq k_n \leq j_n + \Delta_n + 1$$

and the other part of p_θ^{corr} is kept in $\mathcal{R}_\theta f^{\text{CNN}}$. This produces a new projection data $p^{\text{corr}} = [p_\theta^{\text{corr}}]_\theta$, which connects the correction of the metal trace to the surrounding unaffected projection data. It is kind of a seamless surgery of sinogram. Finally, a corrected CT image is reconstructed by the FBP algorithm for p^{corr} , and metals are inserted back into the corrected image. Note that this seamless surgery can also be used for the SMAR algorithm.

There are two key factors for the success of the CNN-MAR method: selection of the appropriate pre-corrected auxiliary CT images and preparation of training data. The first factor provides information to help CNN distinguish between tissue structures and artifacts. The second factor ensures the generality of the trained CNN by including as many kinds of metal artifact cases as possible.

Industrial Application: 3D Cone Beam CT

Industrial X-ray CT is used in various areas of industry as an internal inspection for manufactures such as flaw detection, failure analysis, metrology, and so on. Particularly, the reconstructed image obtained from three-dimensional cone beam CT (CBCT) provides ideal testing techniques to locate and measure volumetric details in three dimensions. However, a potential drawback with CT imaging is the possibility of artifacts due to physical phenomenon such as beam hardening effect.

In industry, computer-aided design (CAD) data is available for in-line inspection systems because most products are designed in the form of CAD data. In Jeon et al. (2021), a metal artifact reduction algorithm was proposed using the shape prior information given by CAD format. It is an extended version of SMAR algorithm in

Section “[Surgery-Based Metal Artifact Reduction \(SMAR\)](#)” to 3D, and CAD data is adopted as a shape prior information. In the SMAR algorithm, it is essential to accurately segment the average fill-in region for the success of the algorithm, and for this purpose, a registration algorithm is proposed to register the CAD data to the reconstructed CT volume.

Data Preparation

First, using the given CAD data, a binary volume data V_{CAD} such as

$$V_{CAD}(x) = \begin{cases} 1, & x \in \text{inside of the object} \\ 0, & x \in \text{outside of the object.} \end{cases}$$

is generated. Unlike the CAD data, the reconstructed CT volume is a three-dimensional image with a gray level with pixel values between 0 and 1. In order to distinguish between the inside and the outside of the scanned object, simple thresholding is applied after denoising. The level of the simple thresholding can be set considering the substance that makes up the scanned object. An anisotropic diffusion model called Perona-Malik (Perona et al. 1994) and the shock filter (Osher and Rudin 1990) are applied to the reconstructed CT volume for noise reduction. The resulting binarized CT volume is denoted as V_{CT} .

Registration via Shape Prior Chan-Vese Model

Since the reconstructed CT volume contains various artifacts, V_{CT} is not exactly the same with V_{CAD} .

With a given shape prior information, the following energy functional of the Chan-Vese model (Chan and Vese 2001) is considered for an image $I: \Omega \rightarrow \mathbb{R}$,

$$E(\phi, c_1, c_2) = \int_{\Omega} (I - c_1)^2 H(\phi) dx + \int_{\Omega} (I - c_2)^2 (1 - H(\phi)) dx, \quad (5)$$

where H is the Heaviside function and ϕ is a level set function representation of the shape prior, whose zero level set is the boundary of the shape prior in the image I (Osher and Sethian 1988). Scalar values c_1 and c_2 become average intensities of I in the regions where ϕ is positive and negative, respectively. The level set-based approach (5) allows V_{CAD} to be registered to V_{CT} , the result being the closest in terms of volume.

Shape Prior SMAR Algorithm: Alignment and Registration

In the average fill-in step of SMAR, segmentation of the average fill-in region can be easily done by registering V_{CAD} into V_{CT} . Since the functional in (5) is non-convex, in order to avoid being trapped at local minima, V_{CAD} needs to be located as close as possible to V_{CT} while being resized to have the same size as V_{CT} before performing the minimization process.

For a three-dimensional binary object V , the moment tensor is defined by

$$T = \begin{bmatrix} T_{xx} & -T_{xy} & -T_{xz} \\ -T_{yx} & T_{yy} & -T_{yz} \\ -T_{zx} & -T_{zy} & T_{zz} \end{bmatrix},$$

where the moments T_{xx}, T_{yy}, T_{zz} and the products of moment T_{xy}, T_{xz}, T_{yz} are given by

$$T_{xx} = \int_V (x^2) dV, T_{yy} = \int_V (y^2) dV, T_{zz} = \int_V (z^2) dV,$$

and

$$T_{xy} = T_{yx} = - \int_V xy dV, T_{yz} = T_{zy} = - \int_V yz dV, T_{xz} = I_{zx} = - \int_V zx dV.$$

Since the moment tensor T is symmetric, by the principal axis theorem, the eigenvectors of T are the principal axes of V .

Let v_1, v_2, v_3 be unit eigenvectors of T . Then corresponding eigenvalues $\lambda_1, \lambda_2, \lambda_3$ satisfy the relation

$$T v_i = \lambda_i v_i, \quad i = 1, 2, 3.$$

For the binary volumes V_1 and V_2 , the ratio of object sizes can be obtained by using the relationship of the eigenvalues. If r denotes the ratio of sizes between V_1 and V_2 , then

$$\begin{cases} T_{xx}^{V_1} = \int_{V_1} (x^2) dV \\ T_{xx}^{V_2} = \int_{V_2} (rx)^2 r^3 dV \end{cases}$$

implies that $r^5 T_{xx}^{V_1} = T_{xx}^{V_2}$. Therefore, the scaling constant r becomes

$$r = \sqrt[5]{\frac{\lambda_{V_2}}{\lambda_{V_1}}}.$$

Using matrices of principal axes, Q_{CAD} and Q_{CT} , and scaling ratio r , the transformation matrix Q to align V_{CAD} to V_{CT} is expressed as

$$Q = r Q_{CT} Q_{CAD}^{-1} = r Q_{CT} Q_{CAD}^T.$$

Then, $V_{align} = Q V_{CAD}$ is closely aligned to V_{CT} .

Now, the functional in (5) for $I = V_{CT}$ is minimized with

$$\phi(x, y, z) = \phi_0 \begin{bmatrix} (x-a)(n_1^2(1-\cos\theta)+\cos\theta)+(y-b)(n_1n_2(1-\cos\theta)-n_3\sin\theta)+(z-c)(n_1n_3(1-\cos\theta)+n_2\sin\theta) \\ (x-a)(n_1n_2(1-\cos\theta)+n_3\sin\theta)+(y-b)(n_2^2(1-\cos\theta)+\cos\theta)+(z-c)(n_2n_3(1-\cos\theta)-n_1\sin\theta) \\ (x-a)(n_1n_3(1-\cos\theta)-n_2\sin\theta)+(y-b)(n_2n_3(1-\cos\theta)+n_1\sin\theta)+(z-c)(n_3^2(1-\cos\theta)+\cos\theta) \end{bmatrix}, \quad (6)$$

where ϕ_0 is a level set function representation of V_{align} . Here, (a, b, c) is the translation factor along x, y, z -axes, respectively, $\mathbf{n} = (n_1, n_2, n_3)$ a unit vector of rotation axis, and θ a rotation angle with respect to the rotation axis \mathbf{n} ; see Fig. 6.

A particle swarm optimization (PSO) technique is used for the minimization process (Eberhart and Kennedy 1995; Jaberipour et al. 2011). To find a, b, c, n_1, n_2, n_3 , and θ of (6) minimizing (5) by applying the modified PSO algorithm (Jaberipour et al. 2011), a strategy for efficient computation is adopted. As shown in Algorithm 1, the parameters are coupled into (a, b, c) , (n_1, n_2, n_3) , and θ depending on their meaning: translation, rotation axis, and rotation angle. First, fixing the center of mass of two volumes at the origin of the computation region, the initial seed $(a, b, c)^{(0)} = (0, 0, 0)$ is set. Then the principal axes of the moment matrices of the two volume data are aligned with the x, y, z -axes. At this time, the first principal

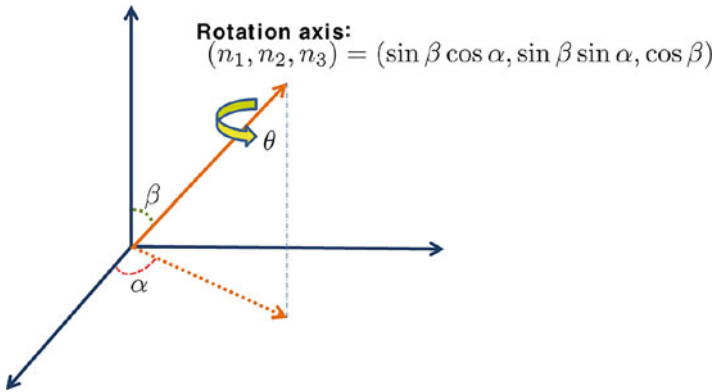


Fig. 6 Three-dimensional rotation. (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)

Algorithm 1 Finding a minimizer of E in (5)

An initial level set ϕ for aligned prior is given.

for $k = 1, 2, 3, \dots$ **do**

Update c_1, c_2 using $c_1 = \frac{\int_{\Omega} I(x)H(\phi)dx}{\int_{\Omega} H(\phi)dx}$ and $c_2 = \frac{\int_{\Omega} I(x)(1-H(\phi))dx}{\int_{\Omega} 1-H(\phi)dx}$.

For fixed $c_1, c_2, (a, b, c)$, and (n_1, n_2, n_3) , update θ .

For fixed $c_1, c_2, (n_1, n_2, n_3)$, and θ , update (a, b, c) .

For fixed c_1, c_2, θ , and (a, b, c) , update normalized (n_1, n_2, n_3) .

end

During the update process, reinitialization is applied first to avoid the numerical deterioration of the interface.

axis is aligned with z -axis and $(n_1, n_2, n_3)^{(0)} = (0, 0, 1)$ is set. Finally, the angle between the center slices of V_{CT} and V_{align} is computed and set as $\theta^{(0)}$. We generate particles in the proper intervals centered at $(a, b, c)^{(0)}$, $(n_1, n_2, n_3)^{(0)}$, and $\theta^{(0)}$. Which variable is updated first depends on how much the updated value affects other variables: updates in order of rotation angle, translation, and rotation axis.

The three-dimensional computation is highly time-consuming, and the most time-consuming part is the PSO process of finding parameters that minimize (5). As the number of particles increases, computation time is linearly increasing. Therefore, a two-resolution approach can be adopted to reduce the computation time of the registration process. For the down-sampled data, less particles can be used. The parameter obtained from the down-sampled data is used as an initial for the registration of the original sized data.

Shape Prior SMAR Algorithm: CT Volume Reconstruction

For sinogram surgery, the two-dimensional forward and backward projection operators, \mathcal{R} and \mathcal{R}^{-1} , are straightforwardly extended to three-dimensional cone beam case. This approach is compatible with the filtered back projection (FBP), and the other sophisticated reconstruction methods can be also applied for the image quality improvement. The segmented region is obtained from the registration result of Algorithm 1 and is used as the average fill-in region. The flowchart of the whole shape prior SMAR algorithm is shown in Fig. 7.

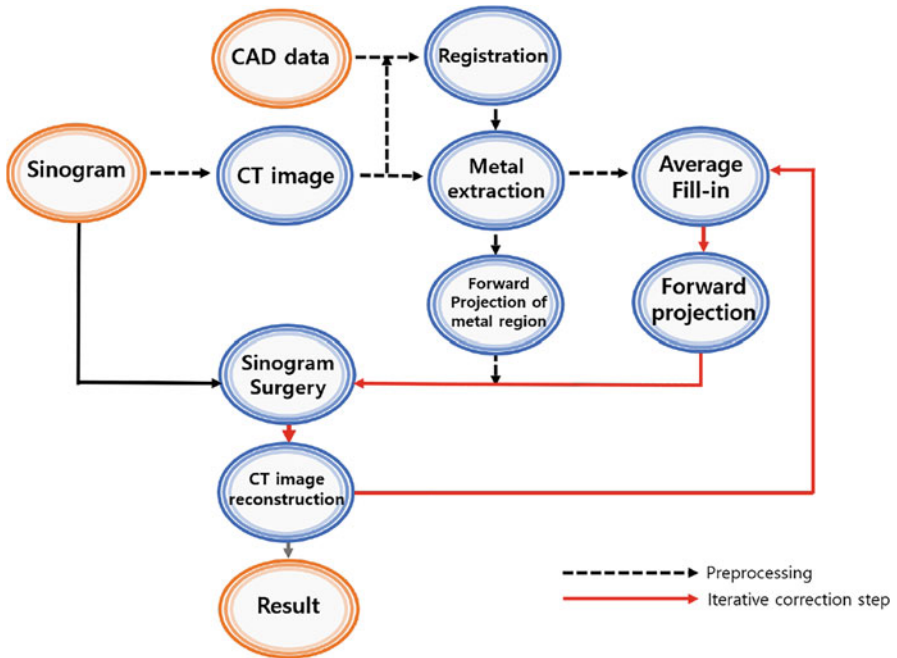


Fig. 7 The flowchart of the shape prior SMAR for 3D CBCT

Simulations and Results

Simulation Conditions

In the simulation study, to generate the polychromatic sinogram, the parallel beam were modeled with 512 channels per detector and 1800 views per half rotation. Seven discrete energy bins (10, 20, 30, 40, 60, 80, 100 keV) were defined (Table 1), and all X-ray coefficients were obtained from the National Institute of Standards and Technology (NIST) database (Hubbell and Seltzer 2004).

For a quantitative analysis, the metal effect-free CT images f^\diamond were used as references, and the performance of MAR algorithms are measured with three error measurements: the relative l_2 error, the relative l_∞ error, and peak signal-to-noise ratio (PSNR). PSNR is defined as

$$\text{PSNR} = 20 \log \frac{\text{peak value}}{\text{RMSE}},$$

where peak value is the range of window and RMSE is the root mean squared error.

A simple notation is used for the relative error between a CT image f and the reference CT image f^\diamond ,

$$\|f\|_{*,\diamond} := \frac{\|f - f^\diamond\|_*}{\|f^\diamond\|_*}, \quad * = 2, \infty,$$

where

$$\|f\|_2 := \sqrt{\sum_i |f_i|^2}, \quad \|f\|_\infty := \max_i |f_i|.$$

The iteration process of the SMAR algorithm was terminated when the relative difference of the sinogram data in the sinogram surgery region became less than $\text{Tol} = 10^{-4}$. In all reconstructed images, the window level with a width of 1000 centered at 0 ($C/W = 0/1000$ (HU)) is used.

NMAR vs. SMAR: Patient Image Simulations

In this section, the numerical results in Jeon and Lee (2018) are presented.

To compare NMAR and SMAR algorithms, patient images were tested (Fig. 8). Three cross-sectional images (pelvis, chest, and dental) were selected from a CT dataset acquired in a dosimetry study of ^{68}Ga -NOTA-RGD PET/CT (Kim et al. 2012). All study procedures were approved by the Institutional Review Board of Seoul National University Hospital, Seoul, Korea. Simulated metallic objects were inserted into the patient images while assuming that the metallic objects are titanium. The X-ray energy spectrum in Table 1 was used. Because it is difficult

Table 1 X-ray intensity and nominal Hounsfield units (HU) and linear attenuation coefficients for materials used in simulations

Energy [keV]	X-ray intensity	Air, dry [1/mm] (-1000 HU)	Adipose [1/mm] (-100 HU)	Water [1/mm] (0 HU)	Tissue [1/mm] (150 HU)	Bone [1/mm] (1000 HU)	Iron [1/mm] (1000 HU \leq)	Titanium [1/mm] (1000 HU \leq)
10	0.0000	6.169E-04	0.3037	0.5329	0.6455	4.989	134.26	49.815
20	1.604	9.372E-05	0.0528	0.08096	0.09876	0.7002	20.21	7.1325
30	26.93	4.263E-05	0.02847	0.03756	0.04548	0.2329	6.435	2.2374
40	49.12	2.994E-05	0.02227	0.02683	0.03226	0.1165	2.856	0.9963
60	42.78	2.506E-05	0.01835	0.02059	0.02458	0.05509	0.9483	0.3447
80	46.31	2.002E-05	0.01673	0.01837	0.02188	0.03901	0.4684	0.1823
100	14.00	1.857E-05	0.01569	0.01707	0.02032	0.03246	0.2925	0.1224



Fig. 8 Patient images (pelvis, chest, and dental). (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

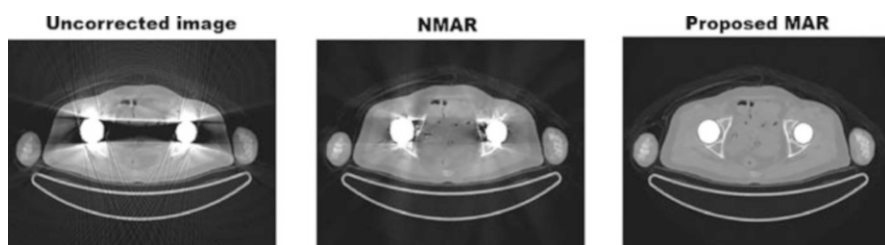


Fig. 9 Patient pelvis experiment. Uncorrected CT image (left) and results of NMAR (middle) and SMAR (right): $C/W = 0/1000$ (HU). (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

to assign energy level-varying X-ray attenuation coefficients for each tissue type in patient images, it is assumed that only the X-ray attenuation coefficient of a metallic object depends on the X-ray energy level.

Figure 9 shows the simulation result for the pelvis of a patient with metallic hips. In the uncorrected CT image, there are streak artifacts between the metallic hips, and they corrupt the anatomical structure. NMAR reduces most of the streak artifacts; however, the resulting image contains bright and dark artifacts which blur the anatomical structure. In comparison, SMAR reduces streak artifacts effectively without generating such bright and dark artifacts. As a result, a clean CT image is obtained and the textures are also preserved well.

As shown in Table 2, the initial relative l_∞ and l_2 errors, 5.0085 and 0.7044, are decreased by nearly half to 3.5728 and 0.2341, respectively, for NMAR. The SMAR algorithm drops the errors more significantly, with the resulting relative l_∞ and l_2 errors becoming 0.2293 and 0.0269, respectively, the values which are decreased by a factor of 20 from the initial levels.

Figure 10 presents the experimental results for the chest of a patient. Two metallic screws inserted into the spine generate streak artifacts, which severely damage the anatomical structure near the spine. While NMAR does reduce the major part of the streak artifacts, it generates additional artifacts near the metallic objects,

Table 2 The performance comparison between NMAR and SMAR for the patient image simulations. The number of iterations is denoted by n . (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

Phantom	Initial error		NMAR		SMAR		
	$\ f^{(0)}\ _{\infty, \diamond}$	$\ f^{(0)}\ _{2, \diamond}$	$\ \cdot\ _{\infty, \diamond}$	$\ \cdot\ _{2, \diamond}$	n	$\ f^{(n)}\ _{\infty, \diamond}$	$\ f^{(n)}\ _{2, \diamond}$
Pelvis	5.0085	0.7044	3.5728	0.2341	16	0.2293	0.0269
Chest	12.1957	0.9187	7.8182	0.3100	26	0.5878	0.0314
Dental	11.9255	1.7471	3.4632	0.4378	14	0.3734	0.0476

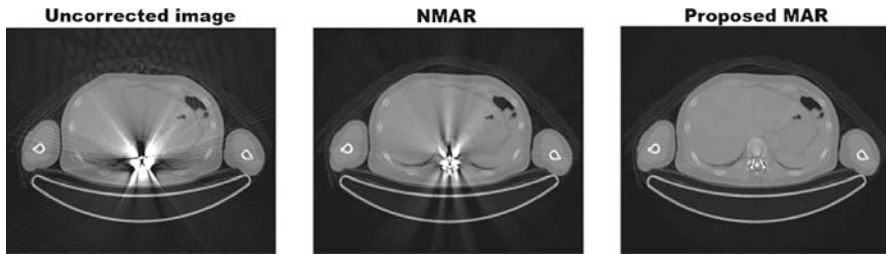


Fig. 10 Patient chest experiment. Uncorrected CT image (left) and results of NMAR (middle) and SMAR (right): $C/W = 0/1000$ (HU). (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

resulting in bright and dark patterns. These newly generated artifacts appear near the metallic objects and thus corrupt the anatomical structure. Moreover, in the NMAR result, the metallic objects are thicker than the original metallic objects, whereas SMAR improves the image quality without generating additional artifacts, so that the anatomical structures near the spine can be successfully distinguished. As shown in Table 2, the initial relative l_{∞} and l_2 errors, 12.1957 and 0.9187, are decreased in the NMAR result to 7.8182 and 0.3100, respectively. The resulting relative l_{∞} and l_2 errors of SMAR are 0.5878 and 0.0314, respectively, values which are lower by a factor of 20 from the initial values.

In the dental image simulations, streak artifacts appear to connect three metallic objects, as shown in Fig. 11. As shown in the zoomed images of the solid boxes, both NMAR and SMAR reduce the streak artifacts. However, NMAR produces the shadow effects even in the region near the teeth and shows undulated artifacts across the entire image domain. Even in a region far from the metallic objects, undulated artifacts also appear, as shown in the zoomed images, and they degrade the image quality. As shown in Table 2, the initial relative l_{∞} and l_2 errors, 11.9255 and 1.7471, are decreased for NMAR to 3.4632 and 0.4378, respectively. The resulting relative l_{∞} and l_2 errors for SMAR are 0.3734 and 0.0476, respectively, showing decreases by a factor of 30 from the initial levels.

In patient image simulations, unlike NMAR, SMAR does not generate undulated artifacts. SMAR produces clear images and performs noticeably better than NMAR.

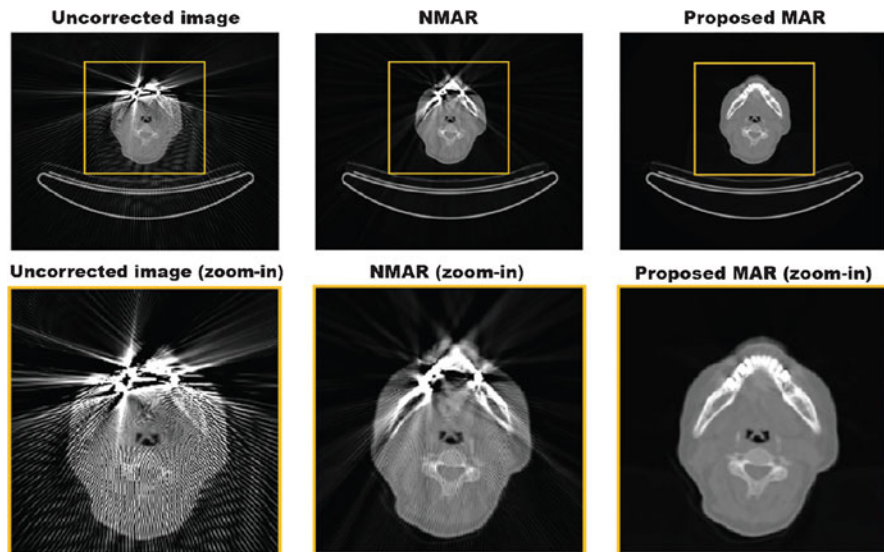


Fig. 11 Patient dental experiment. Uncorrected CT image (left) and results of NMAR (middle) and SMAR (right): $C/W=0/1000$ (HU). (Reprinted from Jeon and Lee (2018) with permission from IOS Press)

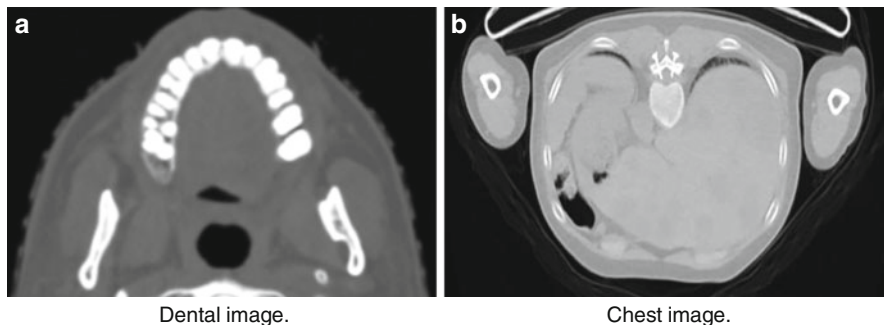


Fig. 12 Reference images for the test of CNN-MAR. (a) Dental image. (b) Chest image

SMAR vs. CNN-MAR

Data Acquisition

For the training of CNN in the CNN-MAR method, data patches are obtained from dental, head, and pelvis images collected from “The 2016 Low-Dose CT Grand Challenge” training dataset (AAPM 2016). To show the performance of the CNN-MAR, a dental image and a chest image in Fig. 12 were chosen from a CT dataset acquired in a dosimetry study of $^{68}\text{Ga-NOTA-RGD}$ PET/CT (Kim et al. 2012). It is expected that the CNN works well when the dental image is used for test, but it will not work when the chest image is used since it is not in the training set.

Results

First, the CNN-MAR method in Zhang and Yu (2018) used BHC and LI results as auxiliary images for the input data, but in this work, SMAR and LI results are also provided as auxiliary images. Since the SMAR algorithm produces better results than the BHC method, it is expected that CNN-MAR will produce a better output if the BHC image input is replaced by an SMAR image. Furthermore, since the SMAR algorithm gives better results than LI, CNN-MAR with only one auxiliary image from the SMAR algorithm is considered.

Figure 13 is the results of the dental case. Indeed, streak artifacts between the teeth are reduced in all methods. However, there are big differences in the red-colored squares. From the numerical results in Table 3, CNN-MAR with SMAR and LI is the best.

Figure 14 is the results of the chest case. Note that the CNN did not learn the chest image patches. From the figure, it can be seen that CNN-MAR with SMAR and LI reduces the streak artifacts well compared with LI. However, comparing with SMAR, breastbone structure is not clearly reconstructed. In Table 4, SMAR shows

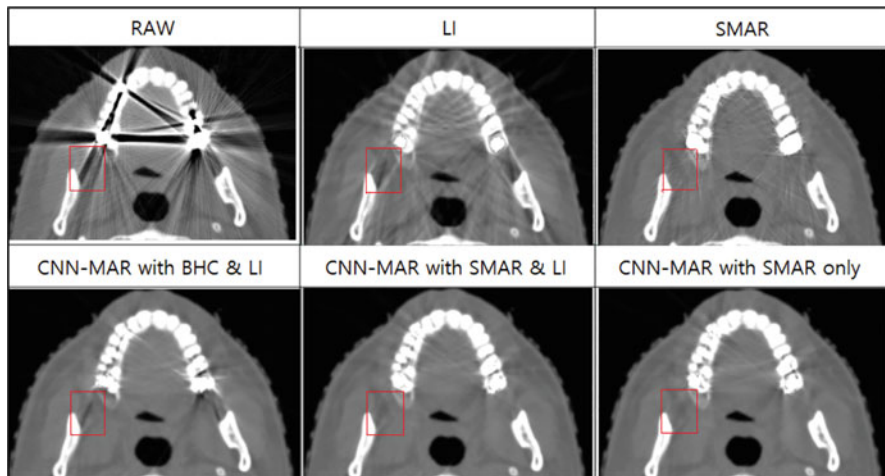


Fig. 13 Dental CT image results with various MAR methods

Table 3 Numerical results for the dental case

	Raw	CNN-MAR with BHC and LI	SMAR	CNN-MAR with SMAR and LI	CNN-MAR with SMAR only
PSNR	15.6867	30.6590	32.6508	32.9713	32.2920
Relative maximum error	6.1329	0.5401	0.7506	0.2812	0.3963
Relative L ² error	0.4666	0.0832	0.0653	0.0638	0.0690

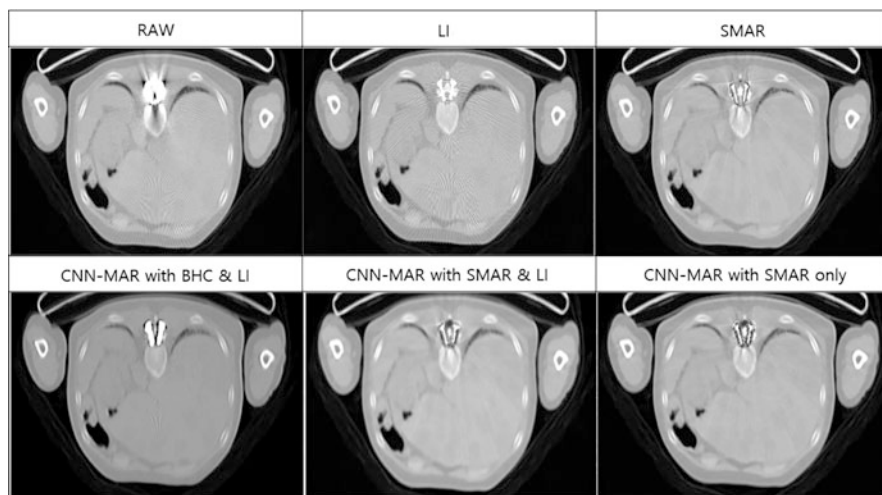


Fig. 14 Chest CT image results with various MAR methods

Table 4 Numerical results for the chest case

	Raw	CNN-MAR with BHC and LI	SMAR	CNN-MAR with SMAR and LI	CNN-MAR with SMAR only
PSNR	8.2912	23.8660	34.1109	29.1756	30.1433
Relative maximum error	12.7421	1.8401	0.4906	0.3192	0.2811
Relative l_2 error	1.0139	0.2812	0.0519	0.0916	0.0819

the best performance in terms of PSNR and the l_2 error. Furthermore, CNN-MAR with SMAR only shows the best performance in terms of the maximum error.

NMAR vs. SMAR for 3D CBCT

In this section, the numerical results in Jeon et al. (2021) are presented.

Phantoms and Hardware Specifications

Two real samples, Samples 1 and 2 in Figs. 15 and 16, respectively, are used for the simulations. Sample 1 consists of an acrylic body with 32 poles, and each pole is made of either Teflon or stainless steel. For the experiment, 4 stainless steel poles and 28 Teflon poles were used. Sample 2 consists of a cylindrical aluminum body with cylindrical holes of various sizes (six large, three middle, and three tiny cylindrical holes), and three lead poles are made to be inserted into the large cylindrical holes. Both samples are designed by CAD and each binary volume data is constructed from them.

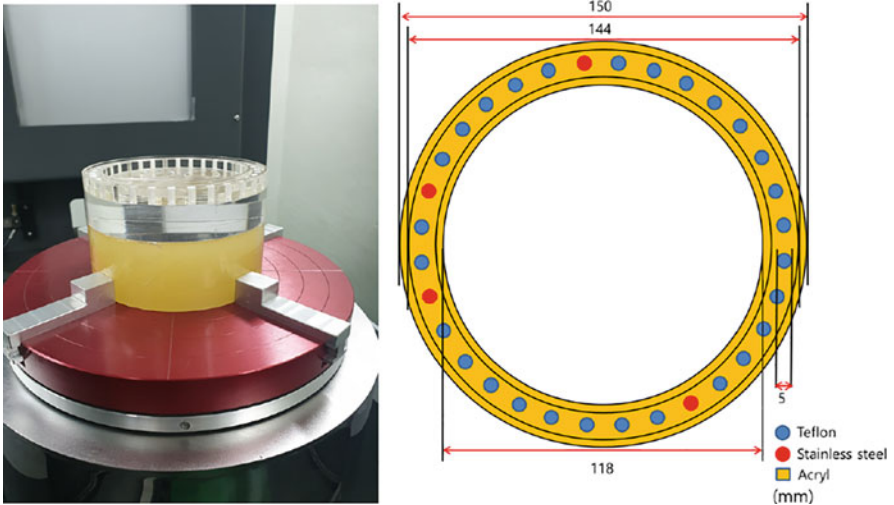


Fig. 15 Sample 1: Data acquisition setting (left) and upper view of the scanned object (right). (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)



Fig. 16 Sample 2: Data acquisition setting (left), upper view of sample body (middle), and lead (Pb) poles. (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)

The projection data of Sample 1 is acquired from an X-ray inspection system of *EB Tech Co., Ltd.*, and the scanning and reconstruction parameters are given in Table 5. The projection data of Sample 2 is acquired from an X-ray inspection system *Bright 240 450 Dual CTR* of *Dukin Co., Ltd.*; the specification of X-ray source is 450 kV and 700 W/1500 W, the focal spot is 0.4 mm/1.0 mm, the detector is a flat panel with size of 409.6 × 409.6 mm, and the resulting projection size per angle is 1024 × 1024.

Table 5 Parameters for Sample 1 data acquisition. (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)

	Values
Source-to-detector distance	2200 mm
Source-to-object distance	1500 mm
Tube voltage	80 kVp
Tube current	5 mA
Number of projection views	720
Scanning angle range	Full rotation (360°)
Detector pixel array size	1024 × 1024
Detector pitch	0.4 mm
Reconstructed volume size	512 × 512 × 512

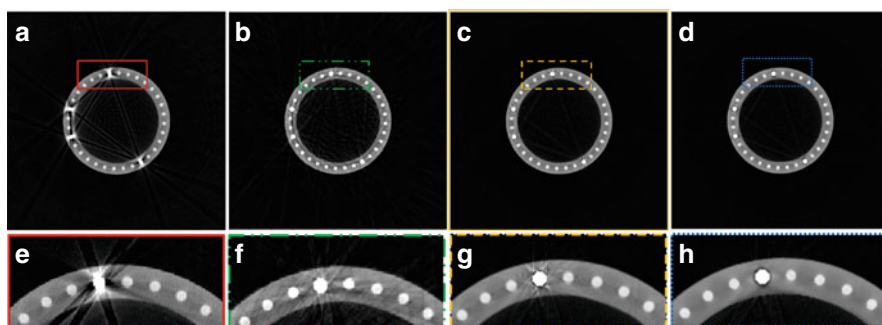


Fig. 17 MAR results for Sample 1: (a) uncorrected CT image, (b) NMAR, (c) SMAR, and (d) shape prior SMAR; (e), (f), (g), and (h) are zoomed-in images of (a), (b), (c), and (d), respectively. (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)

Test I: Performance Evaluation

The performance of the shape prior SMAR algorithm was evaluated and compared with NMAR. To demonstrate the benefits of the shape prior information, the SMAR algorithm was applied to two different situations: one is *with CAD* and the other is *without CAD*. In the *without CAD* case, the average fill-in regions are segmented with simple thresholding. Here, the real data for Sample 1 is used.

As shown in Fig. 17a and e, the inserted stainless poles generate severe metal artifacts. Although a considerable amount of artifacts has disappeared when NMAR is applied as shown in Fig. 17b and f, there are still streak shape of artifacts left. In the case of shape prior SMAR method (Fig. 17d), although slightly uneven parts are observed, streak shape of artifacts are almost eliminated, and the resulting image is very clean. Even though the shape prior information is not available, metal artifacts are reduced significantly as shown in Fig. 17c; however, the performance difference can be clearly seen in the zoomed-in images shown Fig. 17g and h.

Test II: Practical Application – Air Bubble Detection Simulation

Air bubble detection was simulated using Sample 2, where two lead poles are inserted into the aluminum body and about one-fourth of the pore size air bubble

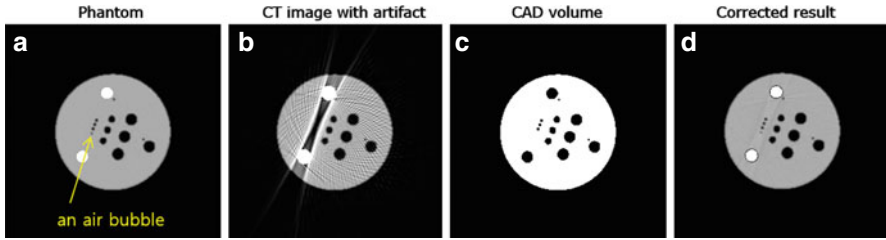


Fig. 18 Center slice views: (a) Sample 2 with an air bubble, (b) reconstructed CT image containing severe beam hardening artifacts, (c) V_{CAD} , and (d) corrected result. (Reprinted from Jeon et al. (2021) with permission from Taylor & Francis)

is included near the three tiny cylindrical pores as shown in Fig. 18a. However, as shown in Fig. 18c, the CAD data does not have the information about the air bubble. For implementation, three discrete bins (40, 60, 100 keV) are defined, and all X-ray attenuation coefficients are obtained from Table 1. For convenience, it is assumed that only the X-ray attenuation coefficient of the lead poles depends on the X-ray energy level. Using the registration results in the previous section, sinogram surgery is performed to reduce the metal artifacts due to two lead poles. As shown in Fig. 18c, an air bubble is not contained in the CAD data. Due to the severe artifacts, the bubble is hardly identified in Fig. 18b. The shape prior SMAR algorithm successfully reduces most of the streak artifacts, and it can also accurately detect the hidden air bubble as shown in Fig. 18d.

Conclusion

In this work, three recent approaches for metal artifact reduction in X-ray CT were investigated: NMAR, SMAR, and CNN-MAR.

NMAR has shown good performance for various types of metallic implants and thus been considered as one of the best currently available MAR algorithms. However, finding a good prior image is at the heart of this algorithm. Incorrect segmentation results can lead to residual artifacts. A more advanced segmentation algorithm will definitely improve the results compared to simple thresholding.

SMAR algorithm was applied to various patient images. As in other MAR approaches based on tissue classification, it is essential for the SMAR algorithm to find a good tissue classification. The average fill-in region is decided based on the tissue classification. The advantage of the SMAR algorithm stems from this point. By filling in the region surrounding metallic objects with the average values, a resulting image is obtained with less streak artifacts. Then this image is used as new input data for the next iteration. The SMAR algorithm tends to converge to a moderate value of the image intensity. Results can be improved when using a more sophisticated segmentation method rather than simple segmentation based on CT numbers.

From the aforementioned simulation results of CNN-MAR, it can be seen that CNN and tissue processing are two mutual beneficial steps. In the CNN step, useful information from pre-corrected auxiliary CT images is fused to avoid strong artifacts. However, mild artifacts typically remain. With the tissue processing, similar to other prior image-based MAR methods, it can remove such moderate artifacts and generates a prior image. Then the final result is produced by doing seamless surgery using the prior image.

In addition, for industrial cone beam CT, shape prior SMAR algorithm reduced metal artifacts using the shape prior information of the scanned object. For the segmentation task, a registration model was designed using level set approach. With CAD data, which is available in most cases in the manufacturing industry, the average fill-in region can be accurately segmented. Also, to overcome the non-convexity and nonlinearity of the energy functional for registration, an algorithm to find good initial parameters was proposed.

In the numerical section, the performance of the SMAR algorithm was compared with that of NMAR both qualitatively and quantitatively, and SMAR outperformed NMAR in the patient image simulation. Through numerical experiments, it was demonstrated that the SMAR algorithm reduces metal artifacts effectively without a loss of anatomical structures.

The CNN-MAR method used two pre-corrected auxiliary CT images to generate the output of CNN. Then, with the projection data of the prior image based on the tissue classification, the seamless surgery produced the final corrected projected data. The quality of the final corrected CT image is dependent upon the choice of the model-based reconstruction methods to generate the auxiliary images. CNN-MAR with SMAR and LI methods outperformed CNN-MAR with BHC and LI methods and with SMAR only. Note that CNN-MAR is not working well when it is applied for new types of images that are not in the training dataset; this result is expected in the general deep learning-based methods. The CNN-MAR method takes a long time for training; however, once trained, it produces outputs in a short time.

In the shape prior SMAR algorithm, through various experiments, the performance of the algorithm and the possibilities for the practical uses were investigated.

References

- AAPM: Low dose CT grand challenge. Resource document. American Association of Physicists in Medicine (2016). <http://www.aapm.org/GrandChallenge/LowDoseCT/>
- Abdoli, M., Ay, M.R., Ahmadian, A., Dierckx, R., Zaidi, H.: Reduction of dental filling metallic artifacts in CT-based attenuation correction of PET data using weighted virtual sinograms optimized by a genetic algorithm. *Med. Phys.* **37**(12), 6166–6177 (2010)
- Bal, M., Spies, L.: Metal artifact reduction in CT using tissue-class modeling and adaptive prefiltering. *Med. Phys.* **33**(8), 2852–2859 (2006)
- Bal, M., Celik, H., Subramanyan, K., Eck, K., Spies, L.: A radial adaptive filter for metal artifact reduction. *Proc. SPIE* **5747**, 2075–2082 (2005)
- Barrett, J.F., Keat, N.: Artifacts in CT: recognition and avoidance. *Radiographics* **24**(6), 1679–1691 (2004)

- Boas, F.E., Fleischmann, D.: CT artifacts: causes and reduction techniques. *Imaging Med.* **4**(2), 229–240 (2012)
- Chan, T., Vese, L.: Active contours without edges. *IEEE Trans. Image Process.* **10**, 266–277 (2001)
- De Man, B., Nuyts, J., Dupont, P., Marchal, G., Suetens, P.: An iterative maximum-likelihood polychromatic algorithm for CT. *IEEE Trans. Med. Imaging* **20**(10), 999–1008 (2001)
- Deans, S.R.: *The Radon Transform and Some of Its Applications*. Dover, New York (2007)
- Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: *MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43 (1995). <https://doi.org/10.1109/MHS.1995.494215>
- Ghani, M.U., Karl, W.C.: Fast enhanced CT metal artifact reduction using data domain deep learning. *IEEE Trans. Comput. Imaging* **6**, 181–193 (2020). <https://doi.org/10.1109/TCL.2019.2937221>
- Gjesteby, L., Yang, Q., Xi, Y., Shan, H., Claus, B., Jin, Y., De Man, B., Wang, G.: Deep learning methods for CT image-domain metal artifact reduction. *Proc. SPIE* **10391**, 103910W (2017). <https://doi.org/10.1117/12.2274427>
- Gu, J., Zhang, L., Yu, G., Xing, Y., Chen, Z.: X-ray CT metal artifacts reduction through curvature based sinogram inpainting. *J. X-Ray Sci. Technol.* **14**(2), 73–82 (2006)
- Helgason, S.: The Radon transform on Euclidean spaces, compact two point homogeneous spaces and Grassmann manifolds. *Acta Math.* **113**, 153–180 (1965)
- Huang, X., Wang, J., Tang, F., Zhong, T., Zhang, Y.: Metal artifact reduction on cervical CT images by deep residual learning. *BioMed. Eng. OnLine* **17**, 175 (2018). <https://doi.org/10.1186/s12938-018-0609-y>
- Hubbell, J.H., Seltzer, S.M.: X-ray mass attenuation coefficients. Resource document. National Institute of Standards and Technology (2004). <https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients/>
- Jaberipour, M., Khorram, E., Karimi, B.: Particle swarm algorithm for solving systems of nonlinear equations. *Comput. Math. Appl.* **62**(2), 566–576 (2011). <https://doi.org/10.1016/j.camwa.2011.05.031>
- Jeon, S., Lee, C.-O.: A CT metal artifact reduction algorithm based on sinogram surgery. *J. X-Ray Sci. Technol.* **26**, 413–434 (2018)
- Jeon, S., Kim, S., Lee, C.-O.: Shape prior metal artefact reduction algorithm for industrial 3D cone beam CT. *Nondestruct. Test. Eval.* **36**(2), 176–194 (2021). <https://doi.org/10.1080/10589759.2019.1709457>
- Kachelrieß, M., Watzke, O., Kalender, W.A.: Generalized multi-dimensional adaptive filtering (MAF) for conventional and spiral single-slice, multi-slice, and cone-beam CT. *Med. Phys.* **28**(4), 475–490 (2001)
- Kalender, W.A., Hebel, R., Ebersberger, J.: Reduction of CT artifacts caused by metallic implants. *Radiology* **164**(2), 576–577 (1987)
- Kano, T., Koseki, M.: A new metal artifact reduction algorithm based on a deteriorated CT image. *J. X-Ray Sci. Technol.* **24**(6), 901–912 (2016)
- Kim, Y., Yoon, S., Yi, J.: Effective sinogram-inpainting for metal artifacts reduction in X-ray CT images. In: *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, pp. 597–600 (2010)
- Kim, J.H., Lee, J.S., Kang, K.W., Lee, H.-Y., Han, S.-W., Kim, T.-Y., Lee, Y.-S., Jeong, J.M., Lee, D.S.: Whole-body distribution and radiation dosimetry of ^{68}Ga -NOTA-RGD, a positron emission tomography agent for angiogenesis imaging. *Cancer Biother. Radiopharm.* **27**, 65–71 (2012)
- Klotz, E., Kalender, W., Sokiranski, R., Felsenberg, D.: Algorithm for the reduction of CT artifacts caused by metallic implants. *Proc. SPIE* **1234**, 642–650 (1990)
- Koehler, T., Brendel, B., Brown, K.: A new method for metal artifact reduction. In: *The Second International Conference on Image Formation in X-Ray Computed Tomography*, Salt Lake City (2012)
- Lemmens, C., Faul, D., Nuyts, J.: Suppression of metal artifacts in CT using a reconstruction procedure that combines MAP and projection completion. *IEEE Trans. Med. Imaging* **28**(2), 250–260 (2009)

- Mahnken, A.H., Raupach, R., Wildberger, J.E., Jung, B., Heussen, N., Flohr, T.G., Günther, R.W., Schaller, S.: A new algorithm for metal artifact reduction in computed tomography: in vitro and in vivo evaluation after total hip replacement. *Investig. Radiol.* **38**(12), 769–775 (2003)
- Meyer, E., Raupach, R., Lell, M., Schmidt, B., Kachelrieß, M.: Normalized metal artifact reduction (NMAR) in computed tomography. *Med. Phys.* **37**(10), 5482–5493 (2010)
- Müller, J., Buzug, T.M.: Spurious structures created by interpolation-based CT metal artifact reduction. *Proc. SPIE* **7258**, 72581Y (2009)
- Osher, S., Rudin, L.I.: Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.* **27**, 919–940 (1990)
- Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988)
- Park, H.S., Hwang, D., Seo, J.K.: Metal artifact reduction for polychromatic X-ray CT based on a beam-hardening corrector. *IEEE Trans. Med. Imaging* **35**, 480–487 (2016)
- Perona, P., Shiota, T., Malik, J.: Anisotropic diffusion. In: ter Haar Romeny, B.M. (ed.) *Geometry-Driven Diffusion in Computer Vision*, pp. 73–92. Springer, Dordrecht (1994)
- Philips Healthcare: Metal artifact reduction for orthopedic implants (O-MAR), White Paper, Philips CT Clinical Science, Andover (2012)
- Prell, D., Kyriakou, Y., Beister, M., Kalender, W.A.: A novel forward projection-based metal artifact reduction method for at-detector computed tomography. *Phys. Med. Biol.* **54**, 6575–6591 (2009)
- Timmer, J.: Metal artifact correction in computed tomography. US Patent, 7,340,027 (2008)
- Verburg, J.M., Seco, J.: CT metal artifact reduction method correcting for beam hardening and missing projections. *Phys. Med. Biol.* **57**(9), 2803–2818 (2012)
- Wang, G., Snyder, D.L., O’Sullivan, J.A., Vannier, M.W.: Iterative deblurring for CT metal artifact reduction. *IEEE Trans. Med. Imaging* **15**(5), 657–664 (1996)
- Watzke, O., Kalender, W.A.: A pragmatic approach to metal artifact reduction in CT: merging of metal artifact reduced images. *Eur. J. Radiol.* **14**(5), 849–856 (2004)
- Wei, J., Chen, L., Sandison, G.A., Liang, Y., Xu, L.X.: X-ray CT high-density artifact suppression in the presence of bones. *Phys. Med. Biol.* **49**(24), 5407–5418 (2004)
- Zhang, Y., Yu, H.: Convolutional neural network based metal artifact reduction in X-ray computed tomography. *IEEE Trans. Med. Imaging* **37**, 1370–1381 (2018)
- Zhang, Y., Yan, H., Jia, X., Yang, J., Jiang, S.B., Mou, X.: A hybrid metal artifact reduction algorithm for X-ray CT. *Med. Phys.* **40**, 041910 (2013)
- Zhang, K., Han, Q., Xu, X., Jiang, H., Ma, L., Zhang, Y., Yang, K., Chen, B., Wang, J.: Metal artifact reduction of orthopedics metal artifact reduction algorithm in total hip and knee arthroplasty. *Medicine (Baltimore)* **99**(11), e19268 (2020)
- Zhao, S., Bae, K.T., Whiting, B., Wang, G.: A wavelet method for metal artifact reduction with multiple metallic objects in the field of view. *J. X-Ray Sci. Technol.* **10**, 67–76 (2002)



Domain Decomposition for Non-smooth (in Particular TV) Minimization

11

Andreas Langer

Contents

Introduction	380
Basic Idea of Domain Decomposition	380
Difficulty for Non-smooth and Non-separable Optimization Problems	385
Domain Decomposition for Smoothed Total Variation	390
Direct Splitting Approach	390
Decomposition Based on the Euler-Lagrange Equation	391
Decomposition for Predual Total Variation	391
Overlapping Domain Decomposition	392
Non-overlapping Domain Decomposition	397
Decomposition for Primal Total Variation	406
Basic Domain Decomposition Approach	406
Domain Decomposition Approach Based on the (Pre)Dual	412
Conclusion	421
References	422

Abstract

Domain decomposition is one of the most efficient techniques to derive efficient methods for large-scale problems. In this chapter such decomposition methods for the minimization of the total variation are discussed. We differ between approaches which directly tackle the (primal) total variation minimization and approaches which deal with their predual formulation. Thereby we mainly concentrate on the presentation of domain decomposition methods which guarantee to converge to a solution of the global problem.

A. Langer (✉)
Centre for Mathematical Sciences, Lund University, Lund, Sweden
e-mail: andreas.langer@math.lth.se

Keywords

Domain decomposition · Schwarz method · Non-smooth optimisation · Total variation

Introduction

Due to the improvement of hardware, the dimensionality of measurements and in particular images is continuously increasing, resulting into large-scale data sets, which want to be processed further. For image processing, e.g., image restoration (denoising, deblurring, inpainting, etc.) and image analysis (segmentation, optical flow calculation, etc.), in the last decades non-smooth minimization problems such as total variation minimization became increasingly important. While being favorable due to the improved enhancement of images compared to smooth imaging approaches, non-smooth minimization problems typically scale badly with the dimension of the data. Hence, existing state-of-the-art standard methods for solving total variation minimization, as described in Burger et al. (2016) and Chambolle et al. (2010), perform well for small- and medium-scale problems. However, they are not able to perform in realistic CPU-time large imaging problems, such as 3D or even 4D imaging (spatial plus temporal dimensions) from functional magnetic resonance in nuclear medical imaging, astronomical imaging, or global terrestrial seismic imaging. Let us mention that with a clever implementation of these standard methods on a parallel architecture such as the graphics processing unit (GPU), one can accelerate them tremendously (Pock et al. 2008).

Here we are interested to address methods to large-scale total variation problems, which allow us to reduce the problem to a finite sequence of subproblems of a more manageable size by splitting the spatial domain into several smaller subdomains. Such methods are known under the name of *domain decomposition*.

Basic Idea of Domain Decomposition

Domain decomposition is a divide-and-conquer technique for solving partial differential equations by iteratively solving on each subdomain an appropriate defined subproblem. It has been shown multiple times (Quarteroni and Valli 1999; Toselli and Widlund 2006) that such methods are one of the most successful methods to construct efficient solvers for large-scale problems. The main reason for this is that they allow to reduce the dimension with the possibility for parallelization. In particular, domain decomposition is one of the most significant ways for devising parallel approaches that can benefit strongly from multiprocessor computers. Parallel approaches are mandatory when one has to solve large-scale numerical simulations, as they appear in a wide range of applications in physics and engineering. We summarize the main advantages of domain decomposition approaches, which include (i) dimension reduction; (ii) enhancement of parallelism;

Fig. 1 Overlapping decomposition into two domains

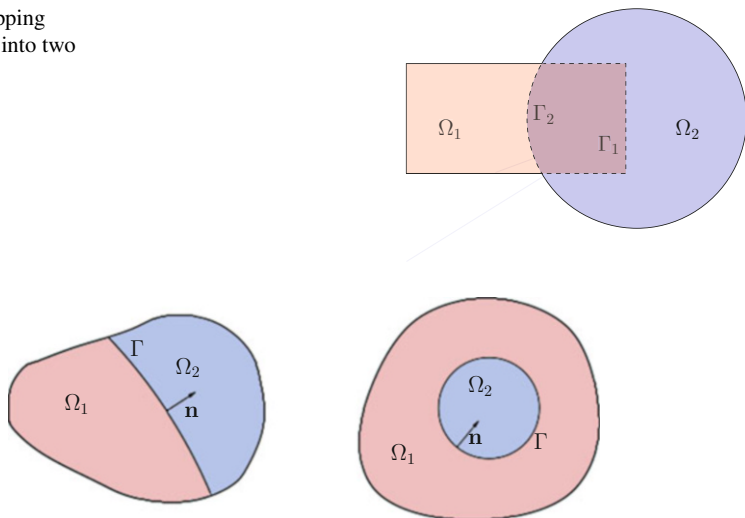


Fig. 2 Non-overlapping decomposition into two domains

(iii) localized treatment of complex and irregular geometries, singularities, and anomalous regions; (iv) and sometimes reduction of the computational complexity of the underlying solution method.

The first known domain decomposition has been proposed by Schwarz in (1869). In particular, he developed an overlapping domain decomposition method in order to show the existence of harmonic functions on irregular regions that are the union of overlapping subregions (Quarteroni and Valli 1999, p. 26). Since this pioneering work and due to the invention of computers and the need of fast computation, domain decomposition became one of the most successful numerical techniques.

When we talk about domain decomposition methods, we distinguish between an overlapping (see Fig. 1) and a non-overlapping (see Fig. 2) separation of the physical domain into two or more subdomains, as well as between successive and parallel computation of the subdomain problems.

Let us discuss the basic idea of domain decomposition methods for the *Poisson problem*, i.e., second-order self-adjoint elliptic problem,

$$\mathcal{L}u \equiv -\Delta u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \tag{1}$$

for a decomposition of the spatial domain Ω into two subdomains. Here u is the unknown function; Δ denotes the Laplace operator; Ω is a two-dimensional domain, i.e., $\Omega \subset \mathbb{R}^2$, with Lipschitz boundary $\partial\Omega$; and f is a given function.

Non-overlapping Domain Decomposition

Let us start by splitting the spatial domain Ω into two non-overlapping subdomains Ω_1 and Ω_2 such that $\overline{\Omega} = \overline{\Omega_1} \cup \overline{\Omega_2}$ and $\Omega_1 \cap \Omega_2 = \emptyset$; cf. Fig. 2. We define the

interface between these two regions by $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. In addition, we assume that the boundaries of the subdomains are Lipschitz continuous. Then problem (1) can be formulated as

$$\begin{cases} \mathcal{L}u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1 = u_2 & \text{on } \Gamma \\ \frac{\partial u_1}{\partial \mathbf{n}} = \frac{\partial u_2}{\partial \mathbf{n}} & \text{on } \Gamma \\ \mathcal{L}u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega \end{cases}, \tag{2}$$

where each \mathbf{n} is the outward pointed normal on Γ from Ω_1 . Here we see that due to the partition of Ω , the original problem (1) is replaced by two subproblems on each subdomain by imposing both Neumann and Dirichlet conditions on Γ . These conditions transmit information from one domain patch to the other and therefore they are called *transmission conditions*. The equivalence between the Poisson problem (1) and the multi-domain problem (2) is in general not obvious, but can be shown under suitable regularity assumptions on f , typically $f \in L^2(\Omega)$, by considering the associated variational formulation; see, for example, Quarteroni and Valli (1999).

The successive Dirichlet-Neumann method We will now focus on solving the multi-domain problem (2) by an iterative method. Such methods typically introduce a sequence of subproblems on Ω_1 and Ω_2 for which boundary conditions at the internal boundary are provided, which play the role of the transmission conditions.

For a given λ^0 , solve for each $k \geq 0$ with respect to u_1^{k+1} and u_2^{k+1}

$$\begin{cases} \mathcal{L}u_1^{k+1} = f & \text{in } \Omega_1 \\ u_1^{k+1} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma \\ u_1^{k+1} = \lambda^k & \text{on } \Gamma \end{cases} \text{ and } \begin{cases} \mathcal{L}u_2^{k+1} = f & \text{in } \Omega_2 \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma \\ \frac{\partial u_2^{k+1}}{\partial \mathbf{n}} = \frac{\partial u_1^{k+1}}{\partial \mathbf{n}} & \text{on } \Gamma \end{cases} \tag{3}$$

with

$$\lambda^{k+1} := \hat{\alpha}u_{2|\Gamma}^{k+1} + (1 - \hat{\alpha})\lambda^k,$$

where $\hat{\alpha} > 0$ is an acceleration or relaxation parameter and $u_{|\Gamma}$ denotes the restriction of the function u to Γ . Note that the boundary conditions on the interface Γ are different for each subdomain problem.

We remark that this method does not necessarily converge, unless assumptions on the parameter $\hat{\alpha}$ or on Ω_1 and Ω_2 are made. However, if it is converging, then the rate of convergence is independent of the mesh size; see Marini and Quarteroni

(1989) for a convergence proof based on a functional analysis argument for partial differential equations.

Parallelism The Dirichlet-Neumann method (3) is generating at each step two boundary value problems, the first in Ω_1 and the second in Ω_2 , to be solved successively. A simple modification frees these two subproblems from each other, which makes it more interesting in view of a parallel implementation. More precisely, when solving the boundary value problem in Ω_2 at the iteration step $k+1$, it is indeed enough to use u_1^k instead of u_1^{k+1} . That is, in (3) we simply replace the Neumann conditions on Γ by the new ones $\frac{\partial u_2^{k+1}}{\partial \mathbf{n}} = \frac{\partial u_1^k}{\partial \mathbf{n}}$.

Variational formulation For $i = 1, 2$ set $(w, v)_{\Omega_i} := \int_{\Omega_i} wv$, $a_i(w, v) := (\mathcal{L}w, v)_{\Omega_i}$, $W_i := \{w_i \in H^1(\Omega_i) : w_i|_{\partial\Omega \cap \partial\Omega_i} = 0\}$, and $W_\Gamma := \{\eta \in H^{1/2}(\Gamma) : \eta = w|_\Gamma \text{ for a suitable } w \in H_0^1(\Omega)\}$, where $H^{1/2}(\Gamma)$ is the trace space of $H^1(\Omega)$ on Γ . Then the variational formulation of (3) reads as follows:

$$\begin{aligned} \text{find } u_1^{k+1} \in W_1 : & \quad a_1(u_1^{k+1}, v_1) = (f, v_1)_{\Omega_1} & \forall v_1 \in H_0^1(\Omega_1) \\ & \quad u_1^{k+1} = \lambda^k & \text{on } \Gamma \\ \text{find } u_2^{k+1} \in W_2 : & \quad a_2(u_2^{k+1}, v_2) = (f, v_2)_{\Omega_2} & \forall v_2 \in H_0^1(\Omega_2) \\ & \quad a_2(u_2^{k+1}, R_2\mu) = (f, R_2\mu)_{\Omega_2} + (f, R_1\mu)_{\Omega_1} - a_2(u_1^{k+1}, R_1\mu) \quad \forall \mu \in W_\Gamma \end{aligned}$$

where $R_i : W_\Gamma \rightarrow W_i$, $i = 1, 2$, is some extension operator.

In a similar but different way, domain decomposition methods for an overlapping splitting of the spatial domain are derived.

Overlapping Domain Decomposition

The so-called *multiplicative* and *additive* Schwarz methods, whose terminology refers to successive and parallel overlapping domain decomposition methods, respectively, are shortly discussed next. Therefore, let us split the spatial domain Ω into two overlapping subdomains Ω_1 and Ω_2 such that $\Omega_1 \cap \Omega_2 \neq \emptyset$ and $\Omega = \Omega_1 \cup \Omega_2$; cf. Fig. 1. Further we denote $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ and $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$ the interior boundaries of the subdomains.

Multiplicative Schwarz method The multiplicative Schwarz method starts with an initial value u^0 defined in Ω and vanishing on $\partial\Omega$ and computes a sequence of approximate solutions u^1, u^2, \dots by solving

$$\left\{ \begin{array}{ll} \mathcal{L}u_1^{k+1} = f & \text{in } \Omega_1 \\ u_1^{k+1} = u|_{\Gamma_1} & \text{on } \Gamma_1 \\ u_1^{k+1} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{ll} \mathcal{L}u_2^{k+1} = f & \text{in } \Omega_2 \\ u_2^{k+1} = u|_{\Gamma_2} & \text{on } \Gamma_2 \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2 \end{array} \right. , \quad (4)$$

with respect to u_1^{k+1} and u_2^{k+1} . That is, the subproblems are solved successively. The next approximate u^{k+1} is then defined by

$$u^{k+1}(x) = \begin{cases} u_2^{k+1}(x) & \text{if } x \in \Omega_2, \\ u_1^{k+1}(x) & \text{if } x \in \Omega \setminus \Omega_2. \end{cases}$$

It can be shown that the multiplicative Schwarz method (4) converges to a solution of problem (1); see Lions (1971, 1988) and for a variational based proof consult (Quarteroni and Valli 1999).

Variational formulation Set $(w, v) := \int_{\Omega} wv$, $a(w, v) := (\mathcal{L}w, v)$, and $W_i^0 := \{v \in H_0^1(\Omega) : v = 0 \text{ in } \Omega \setminus \overline{\Omega}_i\}$, $i = 1, 2$, as closed subspaces of $H_0^1(\Omega)$ by extending their elements on Ω by 0. Moreover, we define the energy

$$\mathcal{J}(w, u) := \frac{1}{2}a(w, w) - (f, w) + a(u, w). \tag{5}$$

Let us rewrite (4) in the following form:

$$\begin{cases} \mathcal{L}(u^{k+1/2} - u^k) = f - \mathcal{L}u^k \text{ in } \Omega_1 \\ u^{k+1/2} - u^k \in W_1^0 \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}(u^{k+1} - u^{k+1/2}) = f - \mathcal{L}u^{k+1/2} \text{ in } \Omega_2 \\ u^{k+1} - u^{k+1/2} \in W_2^0. \end{cases}$$

The variational formulation of method (4) reads as follows: initialize $u^0 \in H_0^1(\Omega)$ and for $k \geq 0$ solve

$$\begin{cases} w_1^k \in W_1^0 : a(w_1^k, v_1) = (f, v_1) - a(u^k, v_1) \quad \text{for all } v_1 \in W_1^0 \\ u^{k+1/2} = u^k + w_1^k \\ w_2^k \in W_2^0 : a(w_2^k, v_2) = (f, v_2) - a(u^{k+1/2}, v_2) \quad \text{for all } v_2 \in W_2^0 \\ u^{k+1} = u^{k+1/2} + w_2^k \end{cases} \tag{6}$$

or equivalently

$$\begin{cases} w_1^k = \arg \min_{w_1 \in W_1^0} \mathcal{J}(w_1, u^k) \\ u^{k+1/2} = u^k + w_1^k \\ w_2^k = \arg \min_{w_2 \in W_2^0} \mathcal{J}(w_2, u^{k+1/2}) \\ u^{k+1} = u^{k+1/2} + w_2^k. \end{cases} \tag{7}$$

Additive Schwarz method If we make the two steps in (4) independent from each other, which allows for parallelization, then we obtain the additive alternating Schwarz method, which computes the sequence of approximations by solving

$$\begin{cases} \mathcal{L}u_1^{k+1} = f & \text{in } \Omega_1 \\ u_1^{k+1} = u_{|\Gamma_1}^k & \text{on } \Gamma_1 \\ u_1^{k+1} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1 \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}u_2^{k+1} = f & \text{in } \Omega_2 \\ u_2^{k+1} = u_{|\Gamma_2}^k & \text{on } \Gamma_2 \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2 \end{cases}. \quad (8)$$

The next update u^{k+1} is then defined by

$$u^{k+1}(x) = \begin{cases} u_1^{k+1}(x) & x \in \Omega \setminus \Omega_2 \\ u_1^{k+1}(x) + u_2^{k+1}(x) - u^k(x) & x \in \Omega_1 \cap \Omega_2. \\ u_2^{k+1}(x) & x \in \Omega \setminus \Omega_1 \end{cases} \quad (9)$$

Variational formulation The variational formulation of method (8) reads as

$$\begin{cases} w_1^k \in W_1^0 : a(w_1^k, v_1) = (f, v_1) - a(u^k, v_1) & \text{for all } v_1 \in W_1^0 \\ w_2^k \in W_2^0 : a(w_2^k, v_2) = (f, v_2) - a(u^k, v_2) & \text{for all } v_2 \in W_2^0 \\ u^{k+1} = u^k + w_1^k + w_2^k \end{cases}$$

or

$$\begin{cases} w_1^k = \arg \min_{w_1 \in W_1^0} \mathcal{J}(w_1, u^k) \\ w_2^k = \arg \min_{w_2 \in W_2^0} \mathcal{J}(w_2, u^k), \\ u^{k+1} = u^k + w_1^k + w_2^k \end{cases} \quad (10)$$

where \mathcal{J} is defined as in (5). By relation (9) we verify that the original formulation (8) is equivalent to the variational formulation.

Note that in the overlapping domain decomposition methods presented above, the subdomain problems are of the same type in each subdomain, while for the non-overlapping methods the subdomain problems differ due two interface conditions, which are distributed among the subdomain problems.

For a broader discussion on domain decomposition approaches for partial differential equations, we refer to Chan and Mathew (1994), Dolean et al. (2015), Mathew (2008), Quarteroni and Valli (1999), Toselli and Widlund (2006), and Smith et al. (2004).

Difficulty for Non-smooth and Non-separable Optimization Problems

Three main issues are of high interest when analyzing domain decomposition methods: (i) convergence, (ii) rate of convergence, and (iii) the independence of the rate of convergence on the mesh size, which can be interpreted as a preconditioning strategy. When talking about convergence, one usually means convergence to a

solution of the global problem. However, we will also learn to know domain decomposition methods that do converge but not necessarily to a solution of the global problem. Hence, in the sequel when we talk about convergence, we distinguish between convergence to some point, which may not be a solution of the global problem, and convergence to a solution of the global problem. For smooth energies, the convergence to a solution of the global problem and the other two concerns are at large well established. We remark, that for non-smooth problems, decomposition algorithms may still work fine as long as the energy splits additively with respect to the domain decomposition. For such problems convergence to a solution of the original problem and sometimes even the rate of convergence are ensured; see, for example, Fornasier (2007), Tseng (2001), Tseng and Yun (2009), and Wright (2015). In (2009) Vonesch and Unser could provide preconditioning effects of a subspace correction algorithm for minimizing a non-smooth energy when applied to deblurring problems. Let us mention that there is a tremendous amount of literature devoted to splitting methods for non-smooth but separable problems in the context of coordinate descent methods (Wright 2015). We are not revising these methods, but concentrate on non-smooth and non-separable problems, where the situation to construct splitting methods that converge to the correct solution seems more complicated as the following counterexample by Warga (1963) indicates.

Example 1. Let $V := [0, 1]^2$, $V_1 := \{(c, 0) : c \in [0, 1]\}$, $V_2 := \{(0, c) : c \in [0, 1]\}$ and $\varphi : V \rightarrow \mathbb{R}$ given by $\varphi(x) = |x_1 - x_2| - \min\{x_1, x_2\}$, where $x = (x_1, x_2)$. We observe that φ is convex but non-smooth and non-additive with respect to the splitting, i.e., $\varphi(x) \neq \varphi((x_1, 0)) + \varphi((0, x_2))$. We have that $0 \in \arg \min_{x \in V_i} \varphi(x)$ for $i \in \{1, 2\}$ and thus $x_2^k = x_1^k = 0$ for all $k \geq 0$. On the contrary $(1, 1) \in \arg \min_{x \in V} \varphi(x)$.

While this example is more of an academic interest, non-smooth and non-separable problems often arise in image processing, where one is interested to obtain a non-smooth solution such that discontinuities (edges) are well represented. This may lead to the minimization of a functional that consists of a total variation term. Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, be an open-bounded set with Lipschitz boundary $\partial\Omega$. For $u \in L^1(\Omega)$ we denote by

$$\int_{\Omega} |Du| := \sup \left\{ \int_{\Omega} u \operatorname{div} \mathbf{p} \, dx : \mathbf{p} \in C_0^1(\Omega, \mathbb{R}^d), |\mathbf{p}|_{\ell^2} \leq 1 \text{ almost everywhere (a.e.) in } \Omega \right\} \quad (11)$$

the total variation of u in Ω (Ambrosio et al. 2000; Giusti 1984), where $C_0^1(\Omega, \mathbb{R}^d)$ is the space of continuously differentiable vector-valued functions with compact support in Ω and $|\cdot|_{\ell^2}$ denotes the standard Euclidean norm. Here and in the rest of this chapter, bold letters indicate vector-valued functions. If $u \in W^{1,1}(\Omega)$, the

Sobolev space of L^1 functions with L^1 distributional derivative, then $\int_{\Omega} |Du| = \int_{\Omega} |\nabla u|_{\ell^2} dx$. Note that different vector norms may be used in the definition of the total variation. More precisely one may use $|\cdot|_{\ell^r}$ with $1 < r \leq \infty$. For example, the case $r = \infty$ is considered in Hintermüller and Kunisch (2004).

It is well established that the total variation preserves edges and discontinuities in images (Chambolle et al. 2010; Chan and Shen 2005), which is one of the reasons why it has been introduced to image processing as a regularization technique (Rudin et al. 1992). In this approach one typically minimizes an energy consisting of a data-fidelity term \mathcal{D} , which enforces the consistency between the observed image and the solution, a total variation term, as a regularizer, and a positive parameter λ weighting the importance of these two terms. That is, one solves

$$\min_u \mathcal{D}(u) + \lambda \int_{\Omega} |Du|.$$

The choice of the data term usually depends on the type of noise contamination. For example, in the case of Gaussian noise, a quadratic L^2 data fidelity term is used, while for impulsive noise an L^1 term is suggested (Alliney 1997) and seems more successful than an L^2 term (Nikolova 2002, 2004). Other and different fidelity terms have been considered in connection with other types of noise models as Poisson noise (Le et al. 2007), multiplicative noise (Aubert and Aujol 2008), and Rician noise (Getreuer et al. 2011). For images which are simultaneously contaminated by Gaussian and impulse noise (Cai et al. 2008), a combined L^1 - L^2 data fidelity term has been suggested and demonstrated to work satisfactorily (Calatroni et al. 2017; Hintermüller and Langer 2013; Langer 2017b, 2019). We will restrict ourselves to Gaussian noise removal, i.e., L^2 data fidelity, as it will cover the fundamental domain decomposition approaches for total variation minimization proposed so far. That is, we consider the so-called L^2 -TV model

$$\min_{u \in BV(\Omega)} \left\{ J(u) := \frac{1}{2} \|Tu - g\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} |Du| \right\}, \quad (12)$$

where $BV(\Omega) = \{u \in L^1(\Omega) : \int_{\Omega} |Du| < \infty\}$ is the space of bounded variation functions (Ambrosio et al. 2000), $g \in L^2(\Omega)$ is the observation, and $T \in \mathcal{L}(L^2(\Omega))$ is a bounded linear operator modeling the image formation device. Typical examples for T are (i) convolution operators, which describe blur in an image; (ii) the identity operator I , if an image is only corrupted by noise; (iii) the characteristic function of a subdomain marking missing parts, i.e., the inpainting domain; or (iv) the Fourier transform, if the observed data are given as corresponding frequencies. Since $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, the embedding $BV(\Omega) \hookrightarrow L^2(\Omega)$ is continuous (Attouch et al. 2014, Theorem 10.1.3), and hence problem (12) is equivalent to $\min_{u \in L^2(\Omega)} J(u)$. In order to ensure the existence of a minimizer of J , we assume that J is coercive in $BV(\Omega)$, i.e., for every sequence $(u^n)_{n \in \mathbb{N}} \subset BV(\Omega)$ with $\|u^n\|_{BV(\Omega)} \rightarrow \infty$, we have $J(u^n) \rightarrow \infty$ or equivalently $\{u \in BV(\Omega) : J(u) \leq c\}$ is bounded in $BV(\Omega)$ for all

constants $c > 0$. This condition holds if T does not annihilate constant functions, i.e., $1 \notin \ker(T)$ (Acar and Vogel 1994).

In the context of total variation minimization, the crucial difficulty in deriving suitable domain decomposition methods lies in the correct treatment of the interfaces of the domain decomposition patches, i.e., the preservation of crossing discontinuities and the correct matching where the solution is continuous. This difficulty is reflected by various effects of the total variation: (i) it is non-smooth, (ii) it preserves discontinuities and edges in images, and (iii) it is non-additive (non-separable) with respect to a non-overlapping domain decomposition, since the total variation of a function on the whole domain equals the sum of the total variation on the subdomains plus the size of the possible jumps at the interface. That is, let Ω_1 and Ω_2 be a disjoint (non-overlapping) decomposition of Ω , then the total variation has the following splitting property (cf. Ambrosio et al. (2000, Theorem 3.84, p. 177)):

$$\int_{\Omega} |D(u|_{\Omega_1} + u|_{\Omega_2})| = \int_{\Omega_1} |D(u|_{\Omega_1})| + \int_{\Omega_2} |D(u|_{\Omega_2})| + \int_{\partial\Omega_1 \cap \partial\Omega_2} |u|_{\Omega_1}^+ - u|_{\Omega_2}^-| \, d\mathcal{H}^{d-1}(x), \tag{13}$$

where \mathcal{H}^d denotes the Hausdorff measure of dimension d . The symbols u^+ and u^- denote the ‘‘interior’’ and ‘‘exterior’’ trace of u on $\partial\Omega_1 \cap \partial\Omega_2$, respectively.

For the L^2 -TV model counterexamples of decomposition methods do exist, indicating failure of such splitting techniques. For example, in Fornasier et al. (2012) for a wavelet space decomposition method, a condition is derived which allows to establish global optimality of a limit point obtained by the decomposition method. Unfortunately, despite the good practical behavior of the method, this condition cannot be ensured to hold in general as shown by an example in Fornasier et al. (2012). Thus, the aforementioned condition may only be used in order to check a posteriori whether the algorithm indeed found a solution or failed to do so. A further counterexample for the L^2 -TV model is presented in Lee and Nam (2017) for a decomposition of the spatial domain into two overlapping or non-overlapping domains; cf. Example 3 below.

We emphasize that for well-known approaches as those in Carstensen (1997), Chan and Mathew (1994), Tai and Tseng (2002), and Tai and Xu (2002), it is not clear yet whether they indeed converge to a global minimizer for non-smooth and non-additive problems, as any convergence theory in this direction is missing.

Instead of considering problem (12), one may tackle one of their dual or predual problems. In fact, if $T = I$, a predual of (12) is given by

$$\begin{aligned} & \min \frac{1}{2} \|\operatorname{div} \mathbf{p} + g\|_{L^2(\Omega)}^2 \quad \text{over } \mathbf{p} \in H_0(\operatorname{div}, \Omega) \\ & \text{subject to (s.t.) } |\mathbf{p}(x)|_{\ell^2} \leq \lambda \text{ for almost all (f.a.a.) } x \in \Omega, \end{aligned} \tag{14}$$

(see Hintermüller and Kunisch (2004) and Hintermüller and Rautenberg (2015)) where $H_0(\operatorname{div}, \Omega) := \{\mathbf{v} \in L^2(\Omega)^d : \operatorname{div} \mathbf{v} \in L^2(\Omega), \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$ with \mathbf{n} being the outward unit normal on $\partial\Omega$. Instead of (14), one may write equivalently

$$\min_{\mathbf{p} \in H_0(\operatorname{div}, \Omega)} \{F(\mathbf{p}) := \frac{1}{2} \|\operatorname{div} \mathbf{p} + g\|_{L^2(\Omega)}^2 + \chi_K(\mathbf{p})\}, \tag{15}$$

or

$$\min_{\mathbf{p} \in H_0(\operatorname{div}, \Omega)} \{\mathfrak{F}(p) := \frac{1}{2} \|\operatorname{div} \mathbf{p} + g\|_{L^2(\Omega)}^2 dx + \mathfrak{J}_\lambda(\mathbf{p})\}$$

where $K := \{\mathbf{p} \in H_0(\operatorname{div}, \Omega) : |\mathbf{p}(x)|_{\ell^2} \leq \lambda \text{ f.a.a. } x \in \Omega\}$, χ_K being the characteristic function of the set K , i.e.,

$$\chi_K(\mathbf{p}) := \begin{cases} 0 & \text{if } \mathbf{p} \in K \\ \infty & \text{if } \mathbf{p} \notin K, \end{cases}$$

and

$$\mathfrak{J}_\lambda(\mathbf{p}) := \begin{cases} 0 & \text{if } |\mathbf{p}(x)|_{\ell^2} \leq \lambda \text{ f.a.a. } x \in \operatorname{Dom}(\mathbf{p}) \\ \infty & \text{otherwise.} \end{cases}$$

with $\operatorname{Dom}(\mathbf{p}) := \{x \in \Omega : \mathbf{p}(x) < \infty\}$ denoting the domain of \mathbf{p} . We note that if $T = I$, then (12) is strictly convex and hence possesses a unique minimizer, while its predual problem (14) may not have a unique solution, as it is “only” convex but not strictly convex. In the case of $T = I$, the solution u^* of (12) and a solution \mathbf{p}^* of (14) are related by

$$u^* = \operatorname{div} \mathbf{p}^* + g, \tag{16}$$

(see Hintermüller and Kunisch (2004)). Note that (14) is separable with respect to a disjoint decomposition of the spatial domain Ω . Let Ω be decomposed into M disjoint subdomains $(\Omega_j)_{j=1}^M$, then for $\mathbf{p} \in H_0(\operatorname{div}, \Omega)$ we have

$$\int_{\Omega} |\operatorname{div} \mathbf{p} + g|^2 dx + \mathfrak{J}_\lambda(\mathbf{p}) = \sum_{j=1}^M \int_{\Omega_j} |\operatorname{div}(\mathbf{p}|_{\Omega_j}) + g|^2 dx + \mathfrak{J}_\lambda(\mathbf{p}|_{\Omega_j}). \tag{17}$$

Nevertheless, domain decomposition approaches as in Tai (2003) and Tai and Xu (2002), which may be utilized for obstacle problems, cannot be directly applied to (14), as the convergence theory used in Tai (2003) and Tai and Xu (2002) essentially relies on the strong convexity of the objective. For a class of non-smooth and non-separable minimization problems in Tseng and Yun (2009), a convergence theory

for coordinate gradient descent methods is established. In that paper, convergence to a minimizer of the global problem could be proven only under the assumption of strict convexity of the objective, which does not hold for (14), and hence the convergence analysis in Tseng and Yun (2009) is not directly applicable to it.

Further basic terminology For a Banach space \mathcal{V} we denote by \mathcal{V}' its topological dual and $\langle \cdot, \cdot \rangle_{\mathcal{V}' \times \mathcal{V}}$ describes the bilinear canonical pairing over $\mathcal{V}' \times \mathcal{V}$. The norm of a Banach space \mathcal{V} is written as $\| \cdot \|_{\mathcal{V}}$. By (\cdot, \cdot) we denote the standard inner product in $L^2(\Omega)$.

For a convex functional $\mathcal{F} : \mathcal{V} \rightarrow \overline{\mathbb{R}}$, we define the *subdifferential* of \mathcal{F} at $v \in \mathcal{V}$ as the set valued function

$$\partial \mathcal{F}(v) := \begin{cases} \emptyset & \text{if } \mathcal{F}(v) = \infty, \\ \{v^* \in \mathcal{V}' : \langle v^*, u - v \rangle_{\mathcal{V}' \times \mathcal{V}} + \mathcal{F}(v) \leq \mathcal{F}(u) \quad \forall u \in \mathcal{V}\} & \text{otherwise.} \end{cases}$$

It is clear from this definition, that $0 \in \partial \mathcal{F}(v)$ if and only if v is a minimizer of \mathcal{F} . Let \mathcal{V}, \mathcal{W} be two Banach spaces, then for any operator $\Lambda : \mathcal{V} \rightarrow \mathcal{W}$ we define by $\Lambda^* : \mathcal{W}' \rightarrow \mathcal{V}'$ its *adjoint*.

For ease of notation, in the sequel for any sequence $(v^n)_{n \in \mathbb{N}}$, we write $(v^n)_n$ instead.

Domain Decomposition for Smoothed Total Variation

If one seeks a minimizer of (12) in the Sobolev space $W^{1,1}(\Omega)$, then (12) becomes

$$\min_{u \in W^{1,1}(\Omega)} \{J(u) = \frac{1}{2} \|Tu - g\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} |\nabla u| \, dx\}. \tag{18}$$

We note that the total variation of $u \in W^{1,1}(\Omega)$ is additive with respect to a disjoint decomposition of Ω , i.e., the interface term in (13) vanishes.

Direct Splitting Approach

By means of space decomposition, split $W^{1,1}(\Omega)$ into M subspaces V_1, \dots, V_M such that $W^{1,1}(\Omega) = \sum_{i=1}^M V_i$. Then following Chen and Tai (2007), Tai (2003), Tai and Tseng (2002), and Tai and Xu (2002) initialize $u^0 \in W^{1,1}(\Omega)$ and solve (18) successively by iterating for $n = 1, 2, \dots$

$$v_i^n \in \arg \min_{v_i \in V_i} J(u^{n+(i-1)/M} + v_i), \quad u^{n+i/M} = u^{n+(i-1)/M} + v_i^n, \quad i=1, \dots, M.$$

Due to the optimality of v_i^n we get that $(J(u^n))_n$ is monotonically decreasing and hence $(u^n)_n \subset W^{1,1}(\Omega)$ is bounded, since J is coercive, i.e., $(u^n)_n \subset \{u \in$

$W^{1,1}(\Omega): J(u) \leq J(u^0)$. Note that $W^{1,1}(\Omega)$ is non-reflexive and (18) is convex but neither strongly nor strictly convex and still non-smooth, due to the presence of the L^1 term. Hence, the convergence theory of Tai (2003), Tai and Xu (2002), Tai and Tseng (2002), and Tseng and Yun (2009) does not cover this splitting algorithm. A similar decomposition method is considered in Chen and Tai (2007) but without any rigorous theoretical convergence analysis.

Decomposition Based on the Euler-Lagrange Equation

Assuming homogeneous Neumann boundary conditions, i.e., $\nabla u \cdot \mathbf{n} = 0$ on $\partial\Omega$ the Euler-Lagrange equation for (18) is

$$T^*(Tu - g) - \lambda \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) = 0.$$

Due to the presence of the term $\frac{1}{|\nabla u|}$, this equation is not well defined at points $\nabla u = 0$. To overcome this shortcoming, we introduce an additional small parameter $\epsilon > 0$ to slightly perturb the total variation semi-norm, such that (12) becomes

$$\min_{u \in W^{1,1}(\Omega)} \frac{1}{2} \|Tu - g\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} \sqrt{|\nabla u|^2 + \epsilon} \, dx. \quad (19)$$

The corresponding Euler-Lagrange equation is then

$$T^*(Tu - g) - \lambda \operatorname{div} \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \epsilon}} \right) = 0. \quad (20)$$

Note that the functional in (19) is now strictly convex, Gâteaux differentiable, and separable. Hence, domain decomposition methods which converge to a solution of the global problem may be constructed following Tseng and Yun (2009). Domain decomposition methods for (19) and (20) have been considered, for example, in Chen and Tai (2007) and Xu et al. (2010, 2014).

While these smoothed problems possess the advantage that domain decomposition methods with desired convergence properties could be possibly designed, they do not generate solutions that preserve discontinuities and edges.

Decomposition for Primal Total Variation

In order to avoid the difficulties due to the minimization of a non-smooth and non-additive energy over a non-reflexive Banach space in (12), the primal problem (14) of (12) may be tackled instead. In particular the smooth objective and the box constraint in (14) seem more amenable to domain decomposition than the structure

of (12). In fact, in Chang et al. (2015) overlapping and in Hintermüller and Langer (2015), Lee et al. (2019b), and Lee and Park (2019a,b) non-overlapping domain decomposition methods for (14) are proposed. Let us review the main ideas and results for these approaches.

Overlapping Domain Decomposition

Let Ω be partitioned into $M \in \mathbb{N}$ overlapping subdomains such that $\Omega = \bigcup_{j=1}^M \Omega_j$ and for any $j \in \{1, \dots, M\}$ there is at least one $i \in \{1, \dots, M\} \setminus \{j\}$ with $\Omega_i \cap \Omega_j \neq \emptyset$. Associated with this decomposition, we define subspaces $V_j := \{\mathbf{p} \in H_0(\text{div}, \Omega) : \text{supp}(\mathbf{p}) \subset \overline{\Omega}_j\}$, $j = 1, \dots, M$. Based on this splitting, the fundamental idea of domain decomposition is to solve (14) or (15) by iteratively minimizing F on the subspaces V_j . Unfortunately $K \neq \sum_{j=1}^M K \cap V_j$ and hence there exist $\mathbf{p}_j \in K \cap V_j$, $j = 1, \dots, M$, such that $\sum_{j=1}^M \mathbf{p}_j \notin K$, due to the overlapping region. This means that in general the subspaces V_j are too large. The introduction of a partition of unity, denoted by $(\theta_j)_{j=1}^M$, with the properties

$$\theta_j \in W^{1,\infty}(\Omega) \quad \text{for } j = 1, \dots, M, \tag{21}$$

$$\sum_{j=1}^M \theta_j = 1, \tag{22}$$

$$\text{supp}(\theta_j) \subset \overline{\Omega}_j \quad \text{for } j = 1, \dots, M, \tag{23}$$

allows us to define $K_j := \{\mathbf{p} \in H_0(\text{div}, \Omega) : |\mathbf{p}(x)|_{\ell^2} \leq |\theta_j(x)|\lambda \text{ f.a.a. } x \in \Omega\}$ for $j = 1, \dots, M$. Properties (22) and (23) ensure that $(\theta_j \mathbf{p})_{j=1}^M$ is a partition of $\mathbf{p} \in H_0(\text{div}, \Omega)$ associated with the domain decomposition such that $\mathbf{p} = \sum_{j=1}^M \theta_j \mathbf{p}$ and $\text{supp}(\theta_j \mathbf{p}) \subset \overline{\Omega}_j$ for $j = 1, \dots, M$. By (21) it is guaranteed that $\theta_j \mathbf{p} \in H_0(\text{div}, \Omega)$ provided that $\mathbf{p} \in H_0(\text{div}, \Omega)$ for $j = 1, \dots, M$. This is easily seen by an application of the (generalized) Hölder inequality:

$$\|\theta_j \mathbf{p}\|_{L^2(\Omega)} \leq \|\theta_j\|_{L^\infty(\Omega)} \|\mathbf{p}\|_{L^2(\Omega)},$$

$$\|\text{div}(\theta_j \mathbf{p})\|_{L^2(\Omega)} \leq \|\nabla \theta_j\|_{L^\infty(\Omega)} \|\mathbf{p}\|_{L^2(\Omega)} + \|\theta_j\|_{L^\infty(\Omega)} \|\text{div} \mathbf{p}\|_{L^2(\Omega)}.$$

Moreover, one immediately sees that $K = \sum_{j=1}^M K_j$. In the case of a successive algorithm, this means (cf. Chang et al. (2015, Algorithm II)):

Note that, since $|\theta_j(\cdot)|\lambda : \overline{\Omega} \rightarrow \mathbb{R}_0^+$ is a bounded function, the existence of a minimizer of the subdomain problems in Algorithm 1 is ensured (Hintermüller and Rautenberg 2017, Proposition 3.2 (b)).

We are actually quite free in how to choose the partition of unity as long as the conditions (21), (22), and (23) hold. For example, one may additionally assume that $\theta_j \geq 0$ almost everywhere in Ω for $j = 1, \dots, M$, as in Chang et al. (2015). One can

Algorithm 1 Basic successive overlapping algorithm for (14)

```

Pick an initial  $\mathbf{p}^0 \in K$ 
for  $n = 0, 1, \dots$  do
  for  $j = 1, \dots, M$  do
     $\mathbf{p}_j^{n+1} \in \arg \min_{\mathbf{p}_j \in K_j} \mathfrak{F}(\mathbf{p}_j + \sum_{i < j} \mathbf{p}_i^{n+1} + \sum_{i > j} \theta_i \mathbf{p}^n)$ 
  end for
   $\mathbf{p}^{n+1} := \sum_{j=1}^M \mathbf{p}_j^{n+1}$ 
end for

```

also view it from the other way round, namely, that the partition of unity provides the overlapping splitting of the spatial domain. From a practical point of view, this has the advantage, that a partition of unity can always be easily constructed and hence an overlapping decomposition of the domain. In the case of a rectangle, which is a usual shape of an image, a simple example for a partition of unity for a splitting into three subdomains is shown in Fig. 3.

The first convergent overlapping domain decomposition method for the minimization of (14) is presented in Chang et al. (2015), here presented in Algorithm 2. There the partition of unity $(\theta_j)_j$ is chosen such that (21), (22), and (23), $\theta_j \geq 0$ and

$$\|\nabla \theta_j\|_{L^\infty(\Omega)} \leq \frac{C_\theta}{\delta}, \quad (24)$$

where $C_\theta > 0$ and $\delta > 0$ denotes the overlapping size, hold. The estimate (24) seems reasonable, as for small overlapping sizes we would expect a larger gradient and it may allow to get a feeling on how the convergence of the algorithm depends on the overlapping size; see Theorem 1.

Algorithm 2 Relaxed successive overlapping algorithm for (14)

```

Pick an initial  $\mathbf{p}^0 \in K$ .
Select a relaxation parameter  $\hat{\alpha} \in (0, 1]$ .
for  $n = 0, 1, \dots$  do
  for  $j = 1, \dots, M$  do
     $\hat{\mathbf{p}}_j^{n+1} \in \arg \min_{\mathbf{p}_j \in K_j} \mathfrak{F}(\mathbf{p}_j + \sum_{i < j} \mathbf{p}_i^{n+1} + \sum_{i > j} \theta_i \mathbf{p}^n)$ 
     $\mathbf{p}_j^{n+1} := (1 - \hat{\alpha})\theta_j \mathbf{p}^n + \hat{\alpha} \hat{\mathbf{p}}_j^{n+1}$ 
  end for
   $\mathbf{p}^{n+1} := (1 - \hat{\alpha})\mathbf{p}^n + \hat{\alpha} \sum_{j=1}^M \hat{\mathbf{p}}_j^{n+1}$ 
end for

```

In comparison to Algorithm 1, in Algorithm 2 a relaxation and associated parameter $\hat{\alpha} \in (0, 1]$ is introduced. This relaxation parameter weights the influence of the current (subspace) minimizer and a previous approximation. However, note that for $\hat{\alpha} = 1$ Algorithm 2 becomes Algorithm 1. Let $(\mathbf{p}_j^n)_n$ and $(\mathbf{p}^n)_n$ be

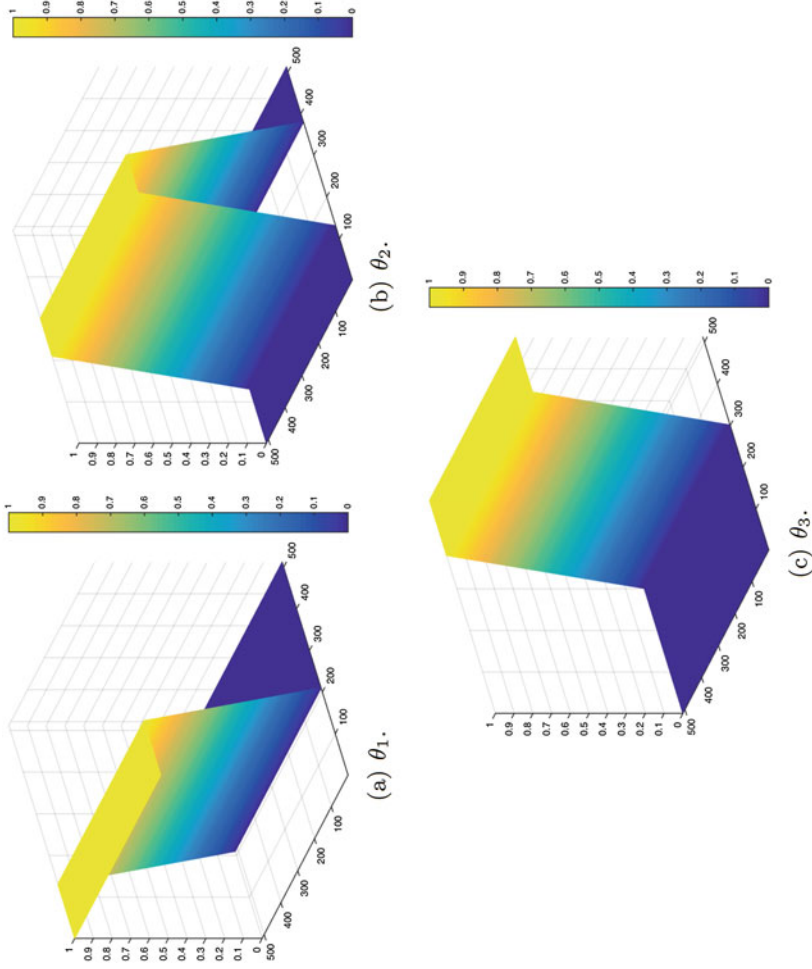


Fig. 3 Partition of unity for a decomposition into three overlapping subdomains. (a) θ_1 (b) θ_2 . (c) θ_3

generated by Algorithm 2, then by a straightforward calculation one easily checks that $\mathbf{p}_j^{n+1} \in K_j$ for all $j \in \{1, \dots, M\}$ and $\mathbf{p}^n \in K$.

By a simple modification, i.e., by replacing \mathbf{p}_i^{n+1} by $\theta_i \mathbf{p}^n$ in the minimization problem in Algorithm 2, the subdomain problems are made independent from each other, which makes it more interesting in view of a parallel implementation. Its parallel version is presented in Algorithm 3.

Algorithm 3 Parallel overlapping algorithm for (14)

```

Pick an initial  $\mathbf{p}^0 \in K$ .
Select a relaxation parameter  $\hat{\alpha} \in (0, \frac{1}{M}]$ .
for  $n = 0, 1, \dots$  do
  for  $j = 1, \dots, M$  do
     $\hat{\mathbf{p}}_j^{n+1} \in \arg \min_{\mathbf{p}_j \in K_j} \mathfrak{F}(\mathbf{p}_j + \sum_{i \neq j} \theta_i \mathbf{p}^n)$ 
  end for
   $\mathbf{p}^{n+1} := (1 - \hat{\alpha})\mathbf{p}^n + \hat{\alpha} \sum_{j=1}^M \hat{\mathbf{p}}_j^{n+1}$ 
end for

```

Remark that the relaxation parameter $\hat{\alpha}$ is now only in the interval $(0, \frac{1}{M}]$, whose range is theoretically justified, in particular to guarantee the monotonic decay of $(F(\mathbf{p}^n))_n$; see Chang et al. (2015) for more details.

The convergence of Algorithms 2 and 3 to a solution of the global problem (15) with rate $\mathcal{O}(\frac{1}{n})$ is guaranteed. We recall this main result by referring to Chang et al. (2015) for its proof.

Theorem 1. *Let \mathbf{p}^* be a minimizer of (14) and let $(\mathbf{p}^n)_n$ be a sequence generated by Algorithm 2 or Algorithm 3. Due to (16) we set $u^n := g + \operatorname{div} \mathbf{p}^n$ for all $n \in \mathbb{N}$ and $u^* := g + \operatorname{div} \mathbf{p}^*$. Then for all $n \in \mathbb{N}$, we have*

$$\|u^n - u^*\|_{L^2(\Omega)}^2 \leq F(\mathbf{p}^n) - F(\mathbf{p}^*) \leq \frac{C^2}{n}$$

with

$$C := \sqrt{\zeta^0} \left(\frac{2}{\hat{\alpha}} (2M + 1)^2 + 8\sqrt{2}C_\theta \lambda |\Omega|^{\frac{1}{2}} (\zeta^0)^{-\frac{1}{2}} \frac{M\sqrt{M}}{\delta\sqrt{\hat{\alpha}}} + \sqrt{2} - 1 \right)$$

where $\zeta^0 := |F(\mathbf{p}^0) - F(\mathbf{p}^*)|$.

We observe that the constant C in Theorem 1 depends on the tunable parameters $\hat{\alpha}$, δ , and M . Some comments according to these parameters are in order. Observe that if the number of subdomains M grows, C grows as well. In order to overcome this behavior, we may use a so-called coloring technique; see, e.g., Toselli and Widlund (2006). That is, Ω is partitioned into M_c classes of overlapping subdomains, where each class has a different color and each class is the union of disjoint subdomains

with the same color. We note that in general the disjoint domains with the same color cannot be solved in parallel without introducing additional new constraints, as the following example, borrowed from Warga (1963), shows.

Example 2. Let $V := [0, 1] \times \{0\} \times [0, 1]$, $V_1 := \{(c, 0, 0) \mid c \in [0, 1]\}$, $V_3 := \{(0, 0, c) \mid c \in [0, 1]\}$, and $\varphi : V \rightarrow \mathbb{R}$ given by $\varphi(x) = |x_1 - x_3| - \min\{x_1, x_3\}$, where $x = (x_1, x_2, x_3)$. We have that $\mathbf{0} = \arg \min_{x_i \in V_i} \varphi(x)$ for $i \in \{1, 3\}$, while $(1, 0, 1) = \arg \min_{x \in V} \varphi(x)$.

However, if the problem is additively separable with respect to the considered disjoint decomposition, then it can be solved independently and in parallel with the disjoint domains. Since this property holds for the considered subdomain problems in Algorithms 2 and 3 with respect to the disjoint domains with the same color, we can replace M by M_c in Algorithms 2 and 3. Further let $N_0 \in \mathbb{N}$ be the maximum number of classes where a point $x \in \Omega$ can belong. A typical decomposition of a rectangular domain into a total of 16 subdomains colored by four colors is illustrated in Fig. 4a. In this example $M_c = 4 = N_0$. A splitting into overlapping stripes, as in Fig. 4b, would even reduce M_c and N_0 to 2. Then the constant C in Theorem 1 can be decreased to

$$C = \sqrt{\xi^0} \left(\frac{2}{\hat{\alpha}} (2M_c + 1)^2 + 8\sqrt{2}C_\theta \lambda |\Omega|^{\frac{1}{2}} (\xi^0)^{-\frac{1}{2}} \frac{M_c \sqrt{N_0}}{\delta \sqrt{\hat{\alpha}}} + \sqrt{2} - 1 \right),$$

where M_c and N_0 may be small, e.g., 2 or 4 (see above), even if the total number of subdomains grows. A complementary behavior is observed for the parameters $\hat{\alpha}$ and δ . That is, the smaller these parameters, the larger the constant C . Consequently one may choose $\hat{\alpha} = 1$ in Algorithm 2 and $\hat{\alpha} = \frac{1}{M}$ (or respectively $\hat{\alpha} = \frac{1}{M_c}$ when using a coloring technique) in Algorithm 3, which will lead to a faster convergence,

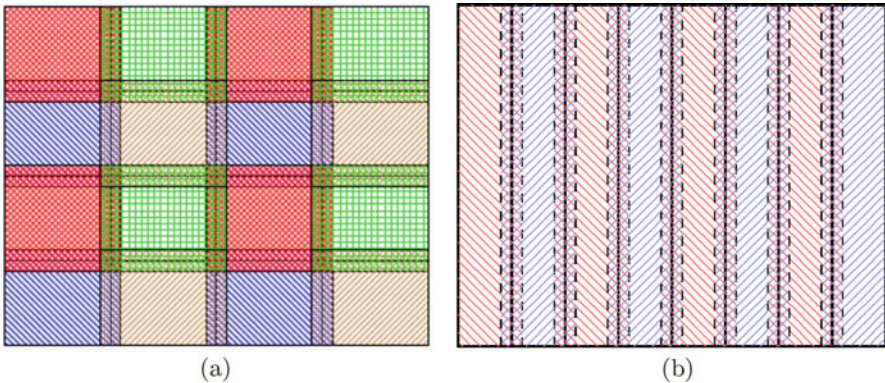


Fig. 4 Domain decomposition by coloring technique in M_c classes (colors) (a) $M_c = 4$. (b) $M_c = 2$

as also numerically observed in Chang et al. (2015). Note that $C \rightarrow \infty$ for $\delta \rightarrow 0$, which would be the non-overlapping case. Moreover, also the partition of unity cannot be carried over to the non-overlapping case, due to (21), as a non-overlapping decomposition would require a discontinuity at the interfaces of the patches. Hence, the here presented convergence results from Chang et al. (2015) do not imply the convergence of a non-overlapping domain decomposition method.

Subdomain problems Let $j \in \{1, \dots, M\}$. Note that $\mathbf{p} \in K_j$ implies $\mathbf{p} \in H_0(\text{div}, \Omega)$. Hence, on a first sight each subproblem seems to be optimized on whole Ω , which would not be in the vein of domain decomposition. However, thanks to the partition of unity functions $(\theta_j)_{j=1}^M$ any $\mathbf{p} \in K_j$ has compact support in Ω_j solely. Consequently one only needs to compute a minimizer of the subproblems of Algorithms 2 and 3 in Ω_j . In order to compute a solution of the subdomain problems in practice, one may use, for example, the iterative scheme presented in Chambolle (2004) adapted to locally adaptive parameters $\theta_j \lambda$. Here for simplicity we assume that $\theta_j \geq 0$. In this situation the algorithm of Chambolle (2004) computes an approximation of $\hat{\mathbf{p}}_j^{n+1}$ in Algorithms 2 and 3 for $j = 1, \dots, M$ by iterating

$$\hat{\mathbf{p}}_j^{n,0} \in K_j \text{ (e.g., } \hat{\mathbf{p}}_j^{n,0} = \hat{\mathbf{p}}_j^n), \quad \hat{\mathbf{p}}_j^{n,\ell+1} = \frac{\theta_j \lambda \hat{\mathbf{p}}_j^{n,\ell} + \theta_j \lambda \tau \nabla(\text{div } \hat{\mathbf{p}}_j^{n,\ell} - g_j)}{\theta_j \lambda + \tau |\nabla(\text{div } \hat{\mathbf{p}}_j^{n,\ell} - g_j)|_{\ell^2}} \quad \text{for } \ell \geq 0, \tag{25}$$

where $g_j := g - \text{div} \left(\sum_{i < j} \mathbf{p}_i^{n+1} + \sum_{i > j} \theta_i \mathbf{p}^n \right)$ for Algorithm 2 and $g_j := g - \text{div} \left(\sum_{i \neq j} \theta_i \mathbf{p}^n \right)$ for Algorithm 3. For $0 < \tau \leq \frac{1}{8}$ one shows analogous to the proof of Chambolle (2004, Theorem 3.1) that the iterates $(\hat{\mathbf{p}}_j^{n,\ell})_\ell$ converge to a respective minimizer $\hat{\mathbf{p}}_j^{n+1}$ of the subdomain problems as $\ell \rightarrow \infty$. Due to the presence of θ_j in the nominator in (25), the update of $\hat{\mathbf{p}}_j^{n,\ell}$ is only performed in Ω_j and hence the subdomain problems are indeed restricted to the respective subdomains.

Non-overlapping Domain Decomposition

As already mentioned above, the convergence analysis carried out for the overlapping domain decomposition algorithms in Chang et al. (2015) leading to Theorem 1 cannot be directly applied to a non-overlapping splitting. In particular, till now it is still an open problem to construct a non-overlapping domain decomposition method for (14) in an infinite dimensional setting which is guaranteed to converge to a minimizer of the original global problem. However, for a finite difference and finite element discretization of (14), splitting methods which converge to the desired optimum are introduced in Hintermüller and Langer (2015), Lee et al. (2019b), and Lee and Park (2019a,b). The first method in this series has been proposed in Hintermüller and Langer (2015) for a finite difference discretization of (14), where instead of $\|\mathbf{p}\|_{\ell^2} \leq \lambda$ the constraint $\|\mathbf{p}\|_{\ell^\infty} \leq \lambda$ is originally used. Nevertheless,

the algorithms in Hintermüller and Langer (2015) and its convergence results can be easily transformed into our setting, i.e., $\|\mathbf{p}\|_{\ell^2} \leq \lambda$, in which we will review them.

Finite Difference Setting

As our main application is image processing, in our discrete setting the spatial domain Ω^h is a mesh in \mathbb{R}^2 of size $N_1 \times N_2$, where $N_1, N_2 \in \mathbb{N}$ with mesh size $x_{i,j} - x_{i+1,j} = 1 = x_{i,j} - x_{i,j+1}$ for $x_{i,j} \in \Omega^h$, i.e., $\Omega^h = \{x_{i,j}\}_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2}}$. The respective function spaces are $X := \{u^h : \Omega^h \rightarrow \mathbb{R}\}$ and $Y = X^2$. For $u^h \in X$ and $\mathbf{p}^h = (p^{h,1}, p^{h,2}) \in Y$ we use the norms $\|u^h\|_X^2 := \|u^h\|_{\ell^2(\Omega^h)}^2 = \sum_{x \in \Omega^h} |u(x)|^2$ and $\|\mathbf{p}^h\|_Y^2 := \|p^{h,1}\|_X^2 + \|p^{h,2}\|_X^2$. On Ω^h the discrete gradient $\nabla_\Omega^h : X \rightarrow Y$ and the discrete $\text{div}_\Omega^h : Y \rightarrow X$ are defined in a standard way by forward and backward differences such that $\text{div}_\Omega^h = -(\nabla_\Omega^h)^*$; see, for example, Hintermüller and Langer (2015). Using this notation the discrete version of (15) is then written as

$$\min_{\mathbf{p}^h \in Y} F^h(\mathbf{p}^h) \tag{26}$$

where $F^h(\mathbf{p}^h) := \|\text{div}_\Omega^h \mathbf{p}^h + g^h\|_X^2 + \chi_{K^h}(\mathbf{p}^h)$ with $K^h := \{\mathbf{p}^h \in Y : |\mathbf{p}^h(x)|_{\ell^2} \leq \lambda \ \forall x \in \Omega^h\}$. Further let Ω^h be decomposed into $M \in \mathbb{N}$ overlapping or non-overlapping subdomains Ω_j^h such that $\Omega^h = \bigcup_{j=1}^M \Omega_j^h$. Associated with the subdomains we define $X_j := \{u^h : \Omega_j^h \rightarrow \mathbb{R}\}$ and $Y_j = X_j \times X_j$ together with the norms $\|u^h\|_{X_j}^2 := \sum_{x \in \Omega_j^h} |u^h(x)|^2$, $\|\mathbf{p}^h\|_{Y_j}^2 := \|p^{h,1}\|_{X_j}^2 + \|p^{h,2}\|_{X_j}^2$ for $u^h \in X_j$ and $\mathbf{p}^h \in Y_j, j = 1, \dots, M$.

Approach via Finite Differences

Let Ω^h be decomposed into $M \in \mathbb{N}$ disjoint subdomains Ω_j^h such that $\Omega^h = \bigcup_{j=1}^M \Omega_j^h$ and $\Omega_j^h = \Omega^h \setminus (\bigcup_{i \neq j} \Omega_i^h)$ for $j = 1, \dots, M$. Associated with this splitting, we set

$$\theta_j^h(x) := \begin{cases} 1 & \text{if } x \in \Omega_j^h \\ 0 & \text{if } x \in \Omega^h \setminus \Omega_j^h, \end{cases} \quad \text{for } j = 1, \dots, M,$$

denoting a discrete partition of unity. In particular $\sum_{j=1}^M \theta_j^h(x) = 1$ for all $x \in \Omega^h$ and $\text{supp}(\theta_j^h) = \Omega_j^h$ for $j = 1, \dots, M$. For $\mathbf{p}^h \in Y$ we note that $\theta_j^h \mathbf{p}^h$ is an orthogonal projection of \mathbf{p}^h onto $Y_j, j = 1, \dots, M$, and $\mathbf{p}^h = \sum_{j=1}^M \theta_j^h \mathbf{p}^h$. Associated with the subdomains we define $K_j^h := \{\mathbf{p}^h \in Y : |\mathbf{p}^h(x)|_{\ell^2} \leq \lambda \theta_j(x) \ \forall x \in \Omega^h\}$ for $j = 1, \dots, M$. With this splitting one may solve (26) by a successive domain decomposition algorithm (see Algorithm 4) or a parallel domain decomposition algorithm (see Algorithm 5; cf. Hintermüller and Langer (2015)).

Note that due to the disjoint decomposition of Ω^h we have that the sequences $(\mathbf{p}^{h,n})_n$ generated by Algorithms 4 and 5 are in K^h , as the constraint in K^h is

Algorithm 4 Successive non-overlapping algorithm for (14)

Pick an initial $\mathbf{p}^{h,0} \in K^h$.
for $n = 0, 1, \dots$ **do**
 for $j = 1, \dots, M$ **do**
 $\mathbf{p}_j^{h,n+1} \in \arg \min_{\mathbf{p}_j^h \in Y} \frac{1}{2} \|\operatorname{div}_\Omega^h(\mathbf{p}_j^h + \sum_{i<j} \mathbf{p}_i^{h,n+1} + \sum_{i>j} \mathbf{p}_i^{h,n}) + g^h\|_X^2 + \mathfrak{J}_{\lambda\theta_j^h}(\mathbf{p}_j^h)$
 end for
 $\mathbf{p}^{h,n+1} := \sum_{j=1}^M \mathbf{p}_j^{h,n+1}$
end for

Algorithm 5 Parallel non-overlapping algorithm for (14)

Pick an initial $\mathbf{p}^{h,0} \in K^h$.
for $n = 0, 1, \dots$ **do**
 for $j = 1, \dots, M$ **do**
 $\mathbf{p}_j^{h,n+1} \in \arg \min_{\mathbf{p}_j^h \in Y} \frac{1}{2} \|\operatorname{div}_\Omega^h(\mathbf{p}_j^h + \sum_{i \neq j} \theta_i^h \mathbf{p}^{h,n}) + g^h\|_X^2 + \mathfrak{J}_{\lambda\theta_j^h}(\mathbf{p}_j^h)$
 end for
 $\mathbf{p}^{h,n+1} := (1 - \frac{1}{M})\mathbf{p}^{h,n} + \frac{1}{M} \sum_{j=1}^M \mathbf{p}_j^{h,n+1}$
end for

pointwise and $|\mathbf{p}_j^{h,n} + \sum_{i<j} \mathbf{p}_i^{h,n} + \sum_{i>j} \mathbf{p}_i^{h,n-1}|_{\ell^2} \leq \lambda$ (Algorithm 4) as well as $|\mathbf{p}_j^{h,n} + \sum_{i \neq j} \theta_i^h \mathbf{p}^{h,n-1}|_{\ell^2} \leq \lambda$ (Algorithm 5). Similar as in Hintermüller and Langer (2015) one shows the following convergence results.

Theorem 2. *Let $(\mathbf{p}^{h,n})_n$ be a sequence generated by Algorithm 4 or Algorithm 5. Then we have that*

- (i) $(F^h(\mathbf{p}^{h,n}))_n$ is decreasing and converges.
- (ii) The sequence $(\mathbf{p}^{h,n})_n$ is bounded in Y and has an accumulation point which is a solution of (26).

Additionally for the parallel non-overlapping domain decomposition method (Algorithm 5), a convergence order of $\mathcal{O}(\frac{1}{n})$ is ensured (Lee and Park 2019a).

Theorem 3. *Let $(\mathbf{p}^{h,n})_n$ be a sequence generated by Algorithm 5 and $\mathbf{p}^{h,*}$ a solution of (26), then for all $n \in \mathbb{N}$ we have*

$$F^h(\mathbf{p}^{h,n}) - F^h(\mathbf{p}^{h,*}) \leq \frac{C}{n},$$

where

$$C := M \left(\sum_{j=1}^M \frac{1}{2} \|\operatorname{div}_\Omega^h \theta_j^h(\mathbf{p}^{h,*} - \mathbf{p}^{h,0})\|_X^2 \right) + (M-1) \left(F^h(\mathbf{p}^{h,0}) - F^h(\mathbf{p}^{h,*}) \right). \quad (27)$$

As in the overlapping case, the constant in Theorem 3 can be reduced, if we use a coloring technique with $M_c \leq M$ classes. Then in (27) M can be replaced by M_c . However, C in Theorem 3 still depends on M as

$$\sum_{j=1}^{M_c} \frac{1}{2} \|\operatorname{div}_{\Omega}^h \theta_j^h(\mathbf{p}^{h,*} - \mathbf{p}^{h,0})\|_X^2 \leq \|\operatorname{div}_{\Omega}^h(\mathbf{p}^{h,*} - \mathbf{p}^{h,0})\|_X^2 + c_1 \left(\max_{x \in \Omega^h} (|\mathbf{p}^{h,*}(x) - \mathbf{p}^{h,0}(x)|_{\ell^2}) \right)^2$$

where $c_1 \geq 0$ is a constant depending on M ; see Lee and Park (2019a).

Algorithm 6 Accelerated parallel non-overlapping algorithm for (14)

Pick an initial $p^0 = q^0 \in K^h$ and $t^0 = 1$.

for $n = 0, 1, \dots$ **do**

for $j = 1, \dots, M$ **do**

$$\mathbf{p}_j^{h,n+1} \in \arg \min_{\mathbf{p}_j^h \in Y} \frac{1}{2} \|\operatorname{div}_{\Omega}^h(M\mathbf{p}_j^h - (M-1)\theta_j^h \mathbf{q}^{h,n} + \sum_{i \neq j} \theta_i^h \mathbf{q}^{h,n}) + g^h\|_X^2 + \mathfrak{J}_{\lambda, \theta_j^h}(M\mathbf{p}_j^h - (M-1)\theta_j^h \mathbf{q}^{h,n})$$

end for

$$\mathbf{p}^{h,n+1} := \sum_{j=1}^M \mathbf{p}_j^{h,n+1}$$

$$t^{n+1} := \frac{1 + \sqrt{1 + 4(n)^2}}{2}$$

$$\mathbf{q}^{h,n+1} := \mathbf{p}^{n+1} + \frac{t^n - 1}{t^{n+1}} (\mathbf{p}^{h,n+1} - \mathbf{p}^{h,n})$$

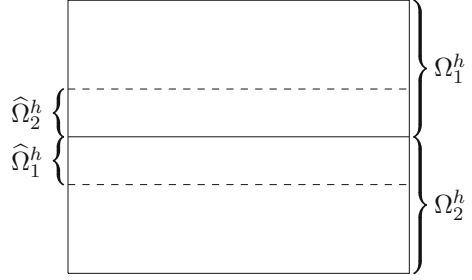
end for

An accelerated version Instead of relaxing the global approximations of two consecutive iterations (see Algorithm 5), in Lee and Park (2019a) the relaxation step is put inside the local solution operator and is performed before the local solutions are computed. Then applying FISTA (Beck and Teboulle 2009) yields Algorithm 6, which converges with order $\mathcal{O}\left(\frac{1}{n^2}\right)$ (Lee and Park 2019a). Note that if in Algorithm 6 $t^n = 1$ for all iterations $n \in \mathbb{N}$, i.e., not using FISTA, then the algorithm still converges but only with order $\mathcal{O}\left(\frac{1}{n}\right)$, as Algorithm 5.

Subdomain problems In order to restrict the subdomain problems in Algorithms 4, 5, and 6 to the respective subdomain plus a possible small stripe around the interface, a certain splitting property of the discrete divergence operator with respect to the disjoint decomposition of the spatial domain Ω^h is required, i.e.,

$$\sum_{x \in \Omega^h} \operatorname{div}_{\Omega}^h(\mathbf{p}_j^h + \mathbf{p}_{j^c}^h)(x) = \sum_{x \in \Omega_j^h \cup \widehat{\Omega}_j^h} \operatorname{div}_{\Omega_j \cup \widehat{\Omega}_j}^h(\mathbf{p}_j^h + \mathbf{p}_{j^c}^h)(x) + \zeta(\mathbf{p}_{j^c}^h),$$

Fig. 5 Non-overlapping domain decomposition of Ω^h into Ω_1^h and Ω_2^h with the small stripes $\widehat{\Omega}_i^h \subset (\Omega^h \setminus \Omega_i^h)$ for $i = 1, 2$



where $\mathbf{p}_j^h \in K_j^h$, $\mathbf{p}_{j^c}^h \in \sum_{i \neq j} K_i^h$, and ζ is a suitable function independent on \mathbf{p}_j^h , $j \in \{1, \dots, M\}$. Here $\text{div}_{\Omega_j \cup \widehat{\Omega}_j}^h$ is the usual discrete divergence on $\Omega_j^h \cup \widehat{\Omega}_j^h$, where $\widehat{\Omega}_j^h \subset \Omega^h \setminus \Omega_j^h$ is a small stripe around the interface between Ω_j^h and $\Omega^h \setminus \Omega_j^h$. A typical choice for $\widehat{\Omega}_j^h$ for which this splitting property holds is shown in Fig. 5 for a decomposition into two domains. Note that the stripe $\widehat{\Omega}_j^h$ may be arbitrarily small and hence in the limit case it may be viewed as the boundary of Ω_j^h inside Ω^h , i.e., $\widehat{\Omega}_j^h = \partial\Omega_j^h \setminus \partial\Omega^h$ and $\partial\Omega_j^h \cap \Omega_j^h = \emptyset$. Then using the above splitting property of the divergence operator, a solution $\mathbf{p}_j^{h,n+1}$ of the subspace minimization problem of Algorithms 5 and 6 in Ω_j is given as

$$\begin{aligned} \mathbf{p}_j^{h,n+1} \Big|_{\Omega_j^h \cup \widehat{\Omega}_j^h} &\in \arg \min_{\mathbf{p}_j^h \in \widehat{Y}_j} \frac{1}{2} \|\text{div}_{\Omega_j \cup \widehat{\Omega}_j}^h \mathbf{p}_j^h + f_j\|_{\widehat{X}_j}^2 + \mathfrak{J}_{\lambda\theta_j^h}(\mathbf{p}_j^h) \\ \mathbf{p}_j^{h,n+1} \Big|_{\Omega^h \setminus (\Omega_j^h \cup \widehat{\Omega}_j^h)} &= 0, \end{aligned} \quad (28)$$

where $f_j = \text{div}_{\Omega_j \cup \widehat{\Omega}_j}^h ((1 - \theta_j^h)\mathbf{p}^{h,n})|_{\Omega_j^h \cup \widehat{\Omega}_j^h} + g_{\Omega_j^h \cup \widehat{\Omega}_j^h}^h + \zeta((1 - \theta_j)\mathbf{p}^{h,n})|_{\Omega_j^h \cup \widehat{\Omega}_j^h}$, $\widehat{X}_j := \{u^h : \Omega_j^h \cup \widehat{\Omega}_j^h \rightarrow \mathbb{R}\}$, $\widehat{Y}_j := \widehat{X}_j \times \widehat{X}_j$, $\|u^h\|_{\widehat{X}_j}^2 := \sum_{x \in \Omega_j^h \cup \widehat{\Omega}_j^h} |u(x)|^2$ for $u^h \in \widehat{X}_j$, and $\widehat{K}_j^h := \{\mathbf{p}^h \in \widehat{Y}_j : |\mathbf{p}^h(x)|_{\ell^2} \leq \lambda\theta_j(x) \forall x \in \Omega_j^h \cup \widehat{\Omega}_j^h\}$. Hence, finding a solution of the subdomain problems reduces to solving an optimization problem on $\Omega_j^h \cup \widehat{\Omega}_j^h$ only. Note that due to the term $\mathfrak{J}_{\lambda\theta_j^h}(\mathbf{p}_j^h)$ the solution $\mathbf{p}_j^{h,n+1}(x) = 0$ for all $x \in \Omega^h \setminus \Omega_j^h$.

For $j = 1, \dots, M$ we define $\widehat{V}_j := \{u^h : \widehat{\Omega}_j^h \rightarrow \mathbb{R}\}$ and rewrite the minimization problem in (28) as a constrained optimization problem in the following form:

$$\min_{\xi_j \in \widehat{V}_j} \frac{1}{2} \|\text{div}_{\Omega_j \cup \widehat{\Omega}_j}^h(\xi_j) + g_{\Omega_j^h \cup \widehat{\Omega}_j^h}^h + \zeta((1 - \theta_j^h)\mathbf{p}^{h,n})|_{\Omega_j^h \cup \widehat{\Omega}_j^h}\|_{\widehat{X}_j}^2 \quad (29)$$

$$\text{s.t. } \text{proj}_{\widehat{V}_j} \xi_j = \text{proj}_{\widehat{V}_j} \mathbf{p}^{h,n} \quad \text{and } |\xi_j(x)|_{\ell^2} \leq \lambda \text{ for all } x \in \Omega_j^h \cup \widehat{\Omega}_j^h,$$

where $\text{proj}_{\widehat{V}_j}$ is the orthogonal projection onto \widehat{V}_j . In the case when $\widehat{\Omega}_j^h = \partial\Omega_j^h \setminus \partial\Omega^h$, the constraint $\text{proj}_{\widehat{V}_j} \boldsymbol{\xi}_j = \text{proj}_{\widehat{V}_j} \mathbf{p}^{h,n}$ is a respective inner boundary condition, which can be worked into the divergence operator. Let us define by $\widehat{\text{div}}^h$ the discrete divergence operator where this new boundary condition is considered. Then the optimization problem in the subdomains can be written as

$$\begin{aligned} \min_{\mathbf{p}_j^h \in Y_j} & \frac{1}{2} \|\widehat{\text{div}}^h(\mathbf{p}_j^h) + g_{|\Omega_j^h}^h + \zeta((1 - \theta_j^h)\mathbf{p}^{h,n})_{|\Omega_j^h}\|_{X_j}^2 \\ \text{s.t. } & |\mathbf{p}_j^h(x)|_{\ell^2} \leq \lambda \text{ for all } x \in \Omega_j^h \end{aligned} \tag{30}$$

or equivalently

$$\min_{\mathbf{p}_j^h \in Y_j} \frac{1}{2} \|\widehat{\text{div}}^h(\mathbf{p}_j^h) + g_{|\Omega_j^h}^h + \zeta((1 - \theta_j^h)\mathbf{p}^{h,n})_{|\Omega_j^h}\|_{X_j}^2 + \mathfrak{J}_\lambda(\mathbf{p}_j^h) \tag{31}$$

which is a minimization problem in Ω_j^h only. In a similar way the subdomain problems in Algorithm 6 can be restricted to the subdomains $\Omega_j^h, j = 1, \dots, M$.

In Hintermüller and Langer (2015) the augmented Lagrangian method (Bertsekas 2014; Ito and Kunisch 2008; Wu and Tai 2010) is used to solve (29). However, in view of (30) and (31), other known methods, as FISTA (Beck and Teboulle 2009) or a primal-dual algorithm (Chambolle and Pock 2011), may be utilized to solve the subspace minimization problems.

Finite Element Approach Based on FISTA

The main idea of this approach is based on the additivity property (17) of the objective of the preudal problem (14). Remark that the discrete divergence operator designed in section “Finite Difference Setting” in a finite difference framework does not satisfy this splitting property. The difficulty in constructing a suitable domain decomposition method based on splitting (17) is that it has to be ensured that an approximation \mathbf{p} obtained by a respective splitting method lies in $H_0(\text{div}, \Omega)$.

In this section we describe a domain decomposition method for (14) in a finite element setting, which is proposed in Lee and Park (2019b).

Let \mathcal{T} be the set of all elements in Ω , e.g., the pixels, and \mathcal{E} the set of edges between elements. Then we discretize (15) by using the lowest-order Raviart-Thomas element space (Raviart and Thomas 1977) defined as

$$Y := \{\mathbf{q} \in H_0(\text{div}, \Omega) : \mathbf{q}|_T \in \mathcal{RT}_0(T) \forall T \in \mathcal{T}, \llbracket \mathbf{q} \cdot \mathbf{n} \rrbracket_E = 0 \forall E \in \mathcal{E}\},$$

where $\mathcal{RT}_0(T) := \{\mathbf{q} : T \rightarrow \mathbb{R}^2 : \mathbf{q}(x) = a + bx, a, b \in \mathbb{R}^2\}$ being the smallest polynomial space with $(\mathbb{P}_0)^2 \subset \mathcal{RT}_0(T) \subset (\mathbb{P}_1)^2$ such that the divergence maps $\mathcal{RT}_0(T)$ onto \mathbb{P}_0 and $\llbracket \mathbf{q} \cdot \mathbf{n} \rrbracket_E$ denotes the jump across the edge E . Note that Y is a conforming approximation of $H_0(\text{div}, \Omega)$. Then $\mathbf{p} \in Y$ may be written as

$$\mathbf{p} = \sum_{i \in \mathcal{I}} (\mathbf{p})_i \psi_i,$$

where \mathcal{I} is the set of indices of the basis functions $(\psi_i)_{i \in \mathcal{I}}$ of Y and $(\mathbf{p})_i$ denotes the respective degree of freedom. Based on these definitions, the finite element discretization of (15) is

$$\min_{\mathbf{p} \in Y} \frac{1}{2} \|\operatorname{div} \mathbf{p} + g\|_{L^2(\Omega)}^2 + \chi_C(\mathbf{p}), \quad (32)$$

where $C := \{\mathbf{p} \in Y : |(\mathbf{p})_i|_{\ell^2} \leq \lambda \forall i \in \mathcal{I}\}$.

Associated with the non-overlapping decomposition $(\Omega_j)_{j=1}^M$ of Ω , we define the respective function spaces as

$$Y_j := \{\mathbf{q} \in H_0(\operatorname{div}, \Omega_j) : \mathbf{q}|_T \in \mathcal{RT}_0(T) \forall T \in \mathcal{T}_j, \llbracket \mathbf{q} \cdot \mathbf{n} \rrbracket_E = 0 \forall E \in \mathcal{E}_j\},$$

where \mathcal{T}_j and \mathcal{E}_j are the collections of all elements and edges in $\overline{\Omega}_j$ for $j = 1, \dots, M$. Let \mathcal{I}_j be the set of indices of the basis functions for Y_j and \mathcal{I}_Γ the set of indices of degree of freedom of Y on $\Gamma := \bigcup_{j < i} \partial\Omega_j \cap \partial\Omega_i$. By $Y_\Gamma := \operatorname{span}\{\psi_i\}_{i \in \mathcal{I}_\Gamma}$ we denote the interface function space. Further let $Y_I := \bigoplus_{j=1}^M Y_j$, $C_j := \{\mathbf{p} \in Y_j : |(\mathbf{p})_i|_{\ell^2} \leq \lambda \forall i \in \mathcal{I}_j\}$, $C_I := \bigoplus_{j=1}^M C_j$ and $C_\Gamma := \{\mathbf{p}_\Gamma \in Y_\Gamma : |(\mathbf{p}_\Gamma)_i|_{\ell^2} \leq \lambda \forall i \in \mathcal{I}_\Gamma\}$. Note that for $\mathbf{p} \in Y$ there exists a unique decomposition such that

$$\mathbf{p} = \mathbf{p}_I \oplus \mathbf{p}_\Gamma = \left(\bigoplus_{j=1}^M \mathbf{p}_j \right) \oplus \mathbf{p}_\Gamma$$

where $\mathbf{p}_j \in Y_j$ and $\mathbf{p}_\Gamma \in Y_\Gamma$. Define $\mathcal{H}_I : Y_\Gamma \rightarrow Y_I$ such that $\mathcal{H}_I \mathbf{p}_\Gamma$ solves

$$\min_{\mathbf{p}_I \in Y_I} \frac{1}{2} \|\operatorname{div}(\mathbf{p}_I \oplus \mathbf{p}_\Gamma) + g\|_{L^2(\Omega)}^2 + \chi_{C_I}(\mathbf{p}_I) \quad (33)$$

for a fixed $\mathbf{p}_\Gamma \in C_\Gamma$. We remark that thanks to the splitting property (17) a solution of (33) can be obtained by independently solving on each subspace

$$\min_{\mathbf{p}_j \in Y_j} \frac{1}{2} \|\operatorname{div}(\mathbf{p}_j \oplus \mathbf{p}_\Gamma|_{\Omega_j}) + g\|_{L^2(\Omega)}^2 + \chi_{C_j}(\mathbf{p}_j).$$

With these definitions instead of minimizing (32), one solves

$$\mathbf{p}_\Gamma \in \arg \min_{\mathbf{p}_\Gamma \in Y_\Gamma} \frac{1}{2} \|\operatorname{div}(\mathcal{H}_I \mathbf{p}_\Gamma \oplus \mathbf{p}_\Gamma) + g\|_{L^2(\Omega)}^2 + \chi_{C_\Gamma}(\mathbf{p}_\Gamma). \quad (34)$$

It can be shown that (i) if $\mathbf{p}^* \in Y$ is a solution of (32), then $\mathbf{p}_\Gamma^* = \mathbf{p}_\Gamma^*$ is a solution of (34) and (ii) if $\mathbf{p}_\Gamma^* \in Y_\Gamma$ is a solution of (34), then $\mathbf{p}^* = \mathcal{H}_I \mathbf{p}_\Gamma^* \oplus \mathbf{p}_\Gamma^*$ is a solution of (32) (Lee and Park 2019a). Using FISTA to solve (34) the domain decomposition algorithm presented in Algorithm 7 is obtained (Lee and Park 2019a), where proj_{C_Γ} is the orthogonal projection onto C_Γ .

Algorithm 7 Parallelizable FISTA for (14)

Choose $L \geq 4$. Pick an initial $\mathbf{p}_\Gamma^0 = \mathbf{q}_\Gamma^0 = 0_\Gamma$ and $t^0 = 1$.
for $n = 0, 1, \dots$ **do**
 $\mathcal{H}_I \mathbf{q}_\Gamma^n \in \arg \min_{\mathbf{p}_I \in Y_I} \frac{1}{2} \|\text{div}(\mathbf{p}_I \oplus \mathbf{p}_\Gamma^n) + g\|_{L^2(\Omega)}^2 + \chi_{C_I}(\mathbf{p}_I)$
 $\mathbf{p}_\Gamma^{n+1} := \text{proj}_{C_\Gamma} \left(\mathbf{q}_\Gamma^n - \frac{1}{L} \text{div}^* \left(\text{div}(\mathcal{H}_I \mathbf{q}_\Gamma^n \oplus \mathbf{q}_\Gamma^n) + g \right) \right)_{|_{Y_\Gamma}}$
 $t^{n+1} := \frac{1 + \sqrt{1 + 4(t^n)^2}}{2}$
 $\mathbf{q}_\Gamma^{n+1} := \mathbf{p}_\Gamma^{n+1} + \frac{t^n - 1}{t^{n+1}} (\mathbf{p}_\Gamma^{n+1} - \mathbf{p}_\Gamma^n)$
end for

We remark once more that the minimizer $\mathcal{H}_I \mathbf{q}_\Gamma^n$ in Algorithm 7 may be obtained by solving independently on each subdomain

$$\mathbf{p}_j^n \in \arg \min_{\mathbf{p}_j \in Y_j} \frac{1}{2} \|\text{div}(\mathbf{p}_j \oplus \mathbf{p}_\Gamma^n|_{\Omega_j}) + g\|_{L^2(\Omega)}^2 + \chi_{C_j}(p_j)$$

and setting $\mathcal{H}_I \mathbf{q}_\Gamma^n = \bigoplus_{j=1}^M \mathbf{p}_j^n$. Due to the utilization of FISTA, Algorithm 7 converges with order $\mathcal{O}(1/n^2)$ to a solution $\mathbf{p}_\Gamma^* \in Y_\Gamma$ of (34).

This approach relies on a splitting into a problem defined on Y_I and a subdomain problem defined on the interface Y_Γ , which are alternately solved. A similar decoupling approach is presented in Lee et al. (2019a), where the functional to be minimized is additively separated with respect to a finite difference discretization into a problem on disjoint subdomains and one interface problem. By utilizing the primal-dual algorithm of Chambolle and Pock (2011) these two problems are successively solved. Note that due to a disjoint splitting, a parallelization of the problem on these disjoint subdomains is possible. This method is used to minimize a functional consisting of a total variation term and an L^1 data fidelity term with applications to image denoising, inpainting, and deblurring. For block coordinate descent methods, a similar splitting approach is presented in Chambolle and Pock (2015).

A FETI Approach

In contrary to the above finite element approach, in Lee et al. (2019b) a further and different domain decomposition method is proposed, where the local function spaces \tilde{Y}_j are defined in the tearing-and-interconnecting fashion by

$$\tilde{Y}_j := \{\mathbf{q} \in H_0(\text{div}, \Omega_j) : \mathbf{q}|_T \in \mathcal{RT}_0(T) \forall T \in \mathcal{T}_j, [[\mathbf{q} \cdot \mathbf{n}]]_E = 0 \forall E \in \mathcal{E} \setminus \Gamma\}.$$

For $\mathbf{p} \in \tilde{Y}_j$ the jump $\llbracket \mathbf{p} \cdot \mathbf{n} \rrbracket_\Gamma$ might be non-zero, which is related to tearing the subdomain solutions apart. Further let $\tilde{\mathcal{I}}_j$ be the set of indices of the basis functions \tilde{Y}_j and $\tilde{Y} = \bigoplus_{j=1}^M \tilde{Y}_j$. Then based on the splitting (17) on each subdomain Ω_j , $j = 1, \dots, M$, the following optimization problem might be solved:

$$\tilde{\mathbf{p}}_j \in \arg \min_{\mathbf{p}_j \in \tilde{Y}_j} \frac{1}{2} \|\operatorname{div} \mathbf{p}_j + g\|_{L^2(\Omega_j)}^2 + \chi_{\tilde{C}_j}(\mathbf{p}_j), \tag{35}$$

where $\tilde{C}_j := \{\mathbf{p} \in \tilde{Y}_j : |(\mathbf{p})_i|_{\ell^2} \leq \lambda \forall i \in \tilde{\mathcal{I}}_j\}$. Then in order to ensure that $\tilde{\mathbf{p}} := \bigoplus_{j=1}^M \tilde{\mathbf{p}}_j \in Y$, where $\tilde{\mathbf{p}}_j$ is a solution of (35), we need to enforce that $\llbracket \tilde{\mathbf{p}} \cdot \mathbf{n} \rrbracket_E = 0$ for all $E \in \mathcal{E}$, i.e., we interconnect the subdomain solutions. For this purpose we define the operator $B : \tilde{Y} \rightarrow \mathbb{R}^{|\mathcal{I}_\Gamma|}$ such that $B\tilde{\mathbf{p}}|_E = \llbracket \tilde{\mathbf{p}} \cdot \mathbf{n} \rrbracket_E$ for E being an edge between two domain patches. Then (32) is equivalent to

$$\min_{\tilde{\mathbf{p}} \in \tilde{Y}} \sum_{j=1}^M \frac{1}{2} \|\operatorname{div} \tilde{\mathbf{p}}_j + g\|_{L^2(\Omega_j)}^2 + \chi_{\tilde{C}_j}(\tilde{\mathbf{p}}_j) \quad \text{s.t.} \quad B\tilde{\mathbf{p}} = 0.$$

Utilizing the method of Lagrange multiplier, this optimization problem can be formulated as a saddle point problem:

$$\min_{\tilde{\mathbf{p}} \in \tilde{Y}} \max_{\mu \in \mathbb{R}^{|\mathcal{I}_\Gamma|}} \sum_{j=1}^M \frac{1}{2} \|\operatorname{div} \tilde{\mathbf{p}}_j + g\|_{L^2(\Omega_j)}^2 + \chi_{\tilde{C}_j}(\tilde{\mathbf{p}}_j) + \langle B\tilde{\mathbf{p}}, \mu \rangle_{\mathbb{R}^{|\mathcal{I}_\Gamma|}}. \tag{36}$$

Since B is bounded, the saddle point problem (36) can be solved by the primal-dual algorithm proposed in Chambolle and Pock (2011) which yields Algorithm 8.

Algorithm 8 Primal-dual FETI algorithm for (14)

Choose $L \geq 2$, $\tau, \sigma > 0$ with $\tau\sigma = \frac{1}{L}$. Let $\tilde{\mathbf{p}}^0 = 0$ and $\lambda^0 = 0$.
for $n = 0, 1, \dots$ **do**
 $\lambda^{n+1} = \lambda^n + \sigma B(2\tilde{\mathbf{p}}^n - \tilde{\mathbf{p}}^{n-1})$
 $\tilde{\mathbf{p}}^{n+1} := \min_{\tilde{\mathbf{p}} \in \tilde{Y}} \sum_{j=1}^M \frac{1}{2} \|\operatorname{div} \tilde{\mathbf{p}}_j + g\|_{L^2(\Omega_j)}^2 + \chi_{\tilde{C}_j}(\tilde{\mathbf{p}}_j) + \frac{1}{2\tau} \|\tilde{\mathbf{p}}_j - \tilde{\mathbf{p}}_j^n + (\tau B^* \lambda^{n+1})|_{\Omega_j}\|_{L^2(\Omega_j)}^2$
end for

Note that the minimization problem in Algorithm 8 can be solved in parallel independently on each subdomain Ω_j , $j = 1, \dots, M$. Moreover, in Lee et al. (2019b) it is stated that this algorithm converges with $\mathcal{O}(1/n)$ to a primal solution $\tilde{\mathbf{p}}^* \in \tilde{Y}$ of (36), which follows from Chambolle and Pock (2016, Theorem 5.1). Moreover, there is an isomorphism $\phi : Y \rightarrow \ker(B) \subset \tilde{Y}$ such that $\phi^{-1}\tilde{\mathbf{p}}^*$ solves (32) (Lee et al. 2019b). This primal-dual domain decomposition approach has been extended to other but related functionals in Lee and Park (2019a), where also image inpainting and segmentation problems with either L^2 or L^1 data fidelity

terms are considered. Also for these applications, the convergence of the splitting method to a minimizer of the global problem is ensured. A similar tearing-and-interconnecting strategy together with the primal-dual algorithm (Chambolle and Pock 2011) has been used in Duan et al. (2016) for image segmentation, more precisely for the convex Chan-Vese model (Chan et al. 2006). However, in this setting the convergence of the algorithm to a minimizer of the global problem seems unclear, as the existence of an isomorphism, similar to the one above, is not shown. Note that in Duan et al. (2016) the minimization of the total variation is directly considered and not its predual counterpart.

Decomposition for Primal Total Variation

In this section we review domain decomposition methods which directly tackle the L^2 -TV model (12). Historically such methods for the L^2 -TV model were considered before its predual problem was suggested to be solved by splitting methods. In particular several domain decomposition methods for tackling directly total variation minimization are presented in Duan et al. (2016), Duan and Tai (2012), Fornasier et al. (2009, 2010), Fornasier and Schönlieb (2009), Hintermüller and Langer (2013, 2014), Lee et al. (2016), and Schönlieb (2009), which are not proven to converge to a solution of the global problem (12). Although in some of these works the proposed methods are theoretically investigated with respect to their convergence properties, in the best case only the criterion is derived under which the convergence to a minimizer of the global problem is achieved. We will review these decomposition methods and their convergence properties in section “[Basic Domain Decomposition Approach](#)”. In particular we even present an example for the minimization of the L^2 -TV model which shows that in general these methods cannot be guaranteed to converge to a minimizer of the global problem. After this quite negative result, we turn to domain decomposition methods for the L^2 -TV model which indeed converge to a solution of the original problem. These methods are based on the splitting methods for the predual problem (14) presented in section “[Decomposition for Predual Total Variation](#)”. We recall that the decomposition methods in section “[Decomposition for Predual Total Variation](#)” converge to a minimizer of the original global problem. Hence, they serve us as a role model for deriving domain decomposition methods for the L^2 -TV model with this desired convergence property. In particular by transforming the decomposition methods of the predual problem via dualization into the function space of the L^2 -TV model, such methods could be constructed, as we will discuss in section “[Domain Decomposition Approach Based on the \(Pre\)Dual](#)”.

Basic Domain Decomposition Approach

Following the general philosophy of subspace correction and inspired by the variational formulations (7) and (10), we seek to minimize J by decomposing

$L^2(\Omega)$ into $M \in \mathbb{N}$ appropriate subspaces U_j such that $L^2(\Omega) = \sum_{j=1}^M U_j$. In terms of domain decomposition, let Ω be separated into M subdomains Ω_j , $j = 1, \dots, M$. Here the decomposition of the domain may be overlapping or non-overlapping. Then $U_j := \{u \in L^2(\Omega) : \text{supp}(u) \subset \Omega_j\}$ for $j = 1, \dots, M$. With this splitting we aim to solve (12) by Algorithm 9.

Algorithm 9 Basic parallel domain decomposition algorithm for (12)

```

Initialise:  $u^0 \in L^2(\Omega)$ 
for  $n = 0, 1, \dots$  do
  for  $j = 1, \dots, M$  do
     $u_j^{n+1} \in \arg \min_{u_j \in U_j} J(u_j + (1 - \theta_j)u^n)$ 
  end for
   $u^{n+1} := \frac{(M-1)u^n + \sum_{j=1}^M u_j^{n+1}}{M}$ 
end for

```

Here $(\theta_j)_{j=1}^M \subset L^\infty(\Omega)$ is a partition of unity with the properties (i) $\sum_{i=1}^M \theta_i = 1$ and (ii) $\theta_j \in U_j$ for $j = 1, \dots, M$. From the assumptions on θ_j we obtain $u^n = \sum_{j=1}^M (\theta_j u^n)$. Further, if the U_j s are orthogonal, i.e., $U = \bigoplus_{j=1}^M U_j$, then $\theta_j u^n = u_j^n$ for all $n \in \mathbb{N}$ and hence there is no need to introduce a partition of unity. The successive version of Algorithm 9 is stated in Algorithm 10.

Algorithm 10 Basic successive domain decomposition algorithm for (12)

```

Initialise:  $u^0 \in U$ 
for  $n = 0, 1, \dots$  do
  for  $j = 1, \dots, M$  do
     $u_j^{n+1} \in \arg \min_{u_j \in U_j} J(u_j + \sum_{i < j} u_i^{n+1} + \sum_{i > j} (\theta_i) u^n)$ 
  end for
   $u^{n+1} := \sum_{j=1}^M u_j^{n+1}$ 
end for

```

We define the orthogonal complement of U_j in $L^2(\Omega)$ by U_j^c , i.e., $L^2(\Omega) = U_j \oplus U_j^c$, and we denote by proj_{U_j} the corresponding orthogonal projection onto U_j for $j = 1, \dots, M$. Moreover, we define the domain of a functional $\mathcal{J} : L^2(\Omega) \rightarrow \bar{\mathbb{R}}$ as the set $\text{Dom}(\mathcal{J}) = \{v \in L^2(\Omega) : \mathcal{J}(v) \neq \infty\}$.

Note that the subspace minimization problems in Algorithm 9 and in Algorithm 10 can be written as constrained optimization problems of the form

$$\min_{v \in L^2(\Omega)} J(v) \quad \text{s.t. } Av = b,$$

where $A : L^2(\Omega) \rightarrow L^2(\Omega)$ is a linear and continuous operator on $L^2(\Omega)$ and $b \in L^2(\Omega)$. In particular, we have

$$\min_{v \in L^2(\Omega)} J(v + b) \quad \text{s.t. } \text{proj}_{U_j^c} v = 0,$$

or equivalently

$$\min_{v \in L^2(\Omega)} J(v) \quad \text{s.t. } \text{proj}_{U_j^c}(v) = \text{proj}_{U_j^c}(b), \quad (37)$$

where $b = \sum_{i < j} u_i^{n+1} + \sum_{i > j} \theta_i u^n$ in Algorithm 10 and $b = (1 - \theta_j)u^n$ for the minimization problem in Algorithm 9 for $j = 1, \dots, M$. For any attainable $b \in U_j$, i.e., there exists an $u \in \text{Dom}(J)$ such that $\text{proj}_{U_j^c}(u) = \text{proj}_{U_j^c}(b)$, we observe that $\{u \in L^2(\Omega) : \text{proj}_{U_j^c}(u) = \text{proj}_{U_j^c}(b), J(u) \leq c\} \subset \{J \leq c\}$ for all $c > 0$, $j = 1, \dots, M$, and $i \in \{1, \dots, M\} \setminus \{j\}$. Hence, by the coercivity of J , the former set is bounded and thus (37) has a solution, as every u_j^n in Algorithms 9 and 10 is attainable.

Let us mention that such domain decomposition algorithms for (12) have been first considered in Fornasier and Schönlieb (2009) for a non-overlapping and in Fornasier et al. (2009, 2010) for an overlapping decomposition of the spatial domain in the context of image reconstruction.

Convergence Properties

It can be shown that Algorithms 9 and 10 generate sequences $(u^n)_n$ in $L^2(\Omega)$, which have subsequences that weakly converge in $L^2(\Omega)$ and $BV(\Omega)$, such that $(J(u^n))_n$ is non-increasing for all $n \in \mathbb{N}$ (Hintermüller and Langer 2013, Proposition 3.1). As a consequence $(J(u^n))_n$ is also convergent, since it is bounded from below. Unfortunately the limit point of such subsequences is not guaranteed to be a solution of the global problem (12), as the following one-dimensional ($d = 1$) counterexample demonstrates:

Example 3. Let $\Omega \subset \mathbb{R}^1$ be the interval (a_1, a_2) , $a_1 < a_2$, decomposed into two subintervals Ω_1 and Ω_2 such that $\Omega = \Omega_1 \cup \Omega_2$ and $|\Omega_j| = l$, $0 < l < a_2 - a_1$, for $j = 1, 2$. Further let $g = 1$ and $T = I$ in (12). We initialize Algorithms 9 and 10 with $u^0 = 0$. In the first iteration for the subspace minimization in U_1 , we solve

$$\min_{u_1 \in U_1} \frac{1}{2} \int_{\Omega} |u_1 + b - 1|^2 dx + \lambda \int_{\Omega} |D(u_1 + b)|,$$

where $b = 0$ since $u^0 = 0$. As $\text{proj}_{U_1^c} u_1 = 0$, one can reason that every minimizer has to be of the form $c1_{\Omega_1}$ for $c \in [0, 1]$, where $1_{\Omega_1}(x) = 1$ if $x \in \Omega_1$ and $1_{\Omega_1}(x) = 0$ otherwise. Therefore, we just need to solve

$$\min_{c \in [0, 1]} \frac{1}{2} \int_{\Omega_1} |c - 1|^2 dx + \lambda c.$$

The associated optimality condition for c is

$$l(c - 1) + \lambda = 0$$

which is equivalent to

$$c = 1 - \frac{\lambda}{l}.$$

Hence, for $\lambda = l$ (in particular for $\lambda \geq l$) the minimizer is $c = 0$ and hence $u_1^1 = 0$. In this situation $b = 0$ for all $j \in \{1, \dots, M\}$ and both algorithms. Consequently $u_j^1 = 0$ for all $j \in \{1, \dots, M\}$ and hence $u^1 = 0 = u^0$. If $\lambda = l$, a repetition of these steps shows that $u^n = 0$ for all $n \in \mathbb{N}$.

On the contrary the minimizer of the global optimization problem (12) is $u^* = 1$ for any $\lambda \geq 0$.

Note that this example works for an overlapping as well as for a non-overlapping decomposition of the spatial domain Ω . Moreover, this counterexample can be easily extended to a multi-domain decomposition and to \mathbb{R}^2 by letting $\Omega \subset \mathbb{R}^2$ be a rectangle decomposed into stripes, for example, as in Fig. 4b. A similar counterexample has been presented in Lee and Nam (2017) for a finite difference discretization by using the relation to the predual problem.

Despite this quite negative result, in a finite difference setting in Hintermüller and Langer (2013) an estimate of the distance of a limit point $u^{h,\infty}$ obtained by discrete version of Algorithm 9 or Algorithm 10 to the true global minimizer $u^{h,*}$ is obtained. Let us use the finite difference setting of section “Finite Difference Setting”, define $X_j^c := \sum_{i \neq j} X_i$ for $j \in \{1, \dots, M\}$, and consider the discrete version of J defined as

$$J^h(u^h) := \|T^h u^h - g^h\|_X^2 + \sum_{x \in \Omega} |\nabla_{\Omega}^h u^h(x)|,$$

where $T^h : X \rightarrow X$ is a bounded linear operator. Then, if $T^{h*} T^h$ is positive definite in the direction $u^{h,\infty} - u^{h,*}$ with smallest eigenvalue $\sigma > 0$ and $\hat{\eta}^h \in \arg \min_{\eta^h \in \bigcup_{j=1}^M (\partial J^h(u^{h,\infty}) \cap X_j^c)} \|\eta^h\|_X$, then

$$\|u^{h,\infty} - u^{h,*}\|_X \leq \frac{\|\hat{\eta}^h\|_X}{\alpha_2 \sigma}. \tag{38}$$

Note that the Lagrange multiplier $\hat{\eta}^h$ indicates the influence of the constraint on the solution. If $\hat{\eta}^h = 0$, then the minimizer of the discrete version of (37) is equivalent to the minimizer of J^h in X and hence is indeed a solution of the global problem. On the contrary, if $\hat{\eta}^h \neq 0$, then the discrete version of the constraint in

(37) has influence on the solution, which consequently does not coincide with the global solution. Hence, this estimate does not contradict with the counterexample, but instead provides an a posteriori upper bound to check whether the algorithm is indeed converged for a considered example. In particular if $\|\hat{\eta}_j^{h,n_k}\|_X \rightarrow 0$ for $k \rightarrow \infty$ along a suitable subsequence $(n_k)_k$ for at least one $j \in \{1, \dots, M\}$, then any accumulation point of the sequence $(u^{h,n})_n$ generated by the discrete version of Algorithm 9 or Algorithm 10 minimizes J^h . By this observation, with the help of this estimate in Hintermüller and Langer (2013), it is demonstrated by numerical experiments that Algorithms 9 and 10 generate sequences which seem to converge to the global minimizer, because $\|\hat{\eta}^h\|_X$ tends to zero.

It is worth mentioning that Algorithms 9 and 10 have not only been proposed for the L^2 -TV model but also for total variation minimization with a combined L^1/L^2 data fidelity term, which seems in particular suitable for removing simultaneously Gaussian and impulsive noise in images (Hintermüller and Langer 2013). For a non-overlapping decomposition of the domain Ω , these algorithms have been also utilized for total variation minimization with an H^{-1} constraint, i.e., for solving

$$\min_{u \in BV(\Omega)} \frac{1}{2} \|Tu - g\|_{-1}^2 + \int_{\Omega} |Du|,$$

where $\|\cdot\|_{-1}$ denotes the $H^{-1}(\Omega)$ norm (Schönlieb 2009). In Chang et al. (2014) a similar splitting method for minimizing the nonlocal total variation (see Gilboa and Osher (2009), Peyré et al. (2008), Zhang et al. (2010), and the references therein for more information on nonlocal total variation) is described without any rigorous theoretical analysis. For total variation image segmentation in Duan et al. (2016) and Duan and Tai (2012), the domain decomposition methods based on an additive decomposition of the objective have been proposed. Nevertheless, a proof of convergence of these methods to a solution of the global problem is missing.

Subspace Minimization

Algorithms 9 and 10 require that the subspace minimization problems are solved exactly, which is in general not easily possible. Moreover, due to the presence of the operator T , which acts on the variable to be minimized, a restriction of the subspace minimization problems to the respective subdomains and subspaces seems in general difficult, in particular if T is a global operator. Therefore, in Fornasier et al. (2010), Fornasier and Schönlieb (2009), and Hintermüller and Langer (2014) the subproblems are approximated by the so-called *surrogate* functionals (Daubechies et al. 2004): assume $a, u_j \in U_j, b \in \sum_{i \neq j} U_i$ and define

$$\begin{aligned}
J_j^s(u_j + b, a + b) &:= J(u_j + b) + \frac{1}{2} \left(\delta \|u_j + b - (a + b)\|_{L^2(\Omega)}^2 \right. \\
&\quad \left. - \|T(u_j + b - (a + b))\|_{L^2(\Omega)}^2 \right) \\
&= \frac{\delta}{2} \|u_j - \left(a + \frac{1}{\delta} T^*(g - T(a + b)) \right)\|_{L^2(\Omega)}^2 \\
&\quad + \lambda \int_{\Omega} |D(u_j + b)| + \Phi(a, b, g)
\end{aligned}$$

for $j = 1, \dots, M$, where $\delta > \|T\|^2$ and Φ is a function of a, b, g and independent of u_j . Now note that u_j is not anymore effected by T and J_j^s is strictly convex. Then a solution u_j^{n+1} of the subspace minimization problems in Algorithms 9 and 10 is realized by the following algorithm: for $u_j^{n,0} \in U_j$

$$u_j^{n,k+1} = \arg \min_{u_j \in U_j} J_j^s(u_j + b, u_j^{n,k} + b), \quad k \geq 0, \quad (39)$$

where $b = \sum_{i < j} u_i^{n+1} + \sum_{i > j} (\theta_i) u^n$ for the alternating algorithm (cf. Algorithm 10) and $b = (1 - \theta_j) u^n$ for the parallel version (cf. Algorithm 9) for $j = 1, \dots, M$. Note that the sequence $(u_j^{n,k})_k$ generated by (39) converges to a minimizer u_j^{n+1} of the corresponding subproblems of Algorithms 9 and 10 (Daubechies et al. 2007).

By introducing small stripes around the interfaces of the subdomains as in Fig. 5, i.e., $\widehat{\Omega}_j \subset \Omega \setminus \Omega_j$ is a small stripe around the interface between Ω_j and $\Omega \setminus \Omega_j$ and by the splitting property of the total variation

$$\begin{aligned}
\int_{\Omega} |D(u_j + b)| &= \int_{\Omega_j \cup \widehat{\Omega}_j} |D(u_j + b)|_{\Omega_j \cup \widehat{\Omega}_j} + \int_{\Omega \setminus (\Omega_j \cup \widehat{\Omega}_j)} |D(b)|_{\Omega \setminus (\Omega_j \cup \widehat{\Omega}_j)} \\
&\quad + \int_{\partial(\Omega_j \cup \widehat{\Omega}_j) \cap \partial(\Omega \setminus (\Omega_j \cup \widehat{\Omega}_j))} |b^+ - b^-| d\mathcal{H}^{d-1}(x),
\end{aligned}$$

where b is understood as above, we can restrict the minimization problem in (39) to the domain $\Omega_j \cup \widehat{\Omega}_j$ for $j = 1, \dots, M$, respectively. Then the respective subdomain problems can be written as constrained minimization problems of the form

$$\begin{aligned}
\min_{u_j \in U_j \oplus \widehat{U}_j} & \frac{\delta}{2} \|u_j - z_j\|_{L^2(\Omega_j \cup \widehat{\Omega}_j)}^2 + \lambda \int_{\Omega_j \cup \widehat{\Omega}_j} |D(u_j + b)|_{\Omega_j \cup \widehat{\Omega}_j} \\
\text{s.t. } & \text{proj}_{\widehat{U}_j} u_j = 0
\end{aligned} \quad (40)$$

where $\widehat{U}_j := \{u \in L^2(\Omega) : \text{supp}(u) \subset \widehat{\Omega}_j\}$, $z_j = \left(u_j^{n,k} + \frac{1}{\delta} T^* \left(g - T(u_j^{n,k} + b) \right) \right)_{|\Omega_j \cup \widehat{\Omega}_j}$

and $b \in \sum_{i \neq j} U_i$ as above. Note that such a splitting holds for overlapping and non-overlapping domain decompositions. Moreover, in case of an overlapping domain decomposition in Fornasier et al. (2010) for a discrete setting, the subproblems are completely restricted to Ω_j , $j = 1, \dots, M$, respectively, due to an induced trace condition, i.e., $\widehat{\Omega}_j$ is replaced by $\Gamma_j := \partial\Omega_j \setminus \partial\Omega$ and the constraint in (40) is then a trace condition on Γ_j . In Fornasier et al. (2010) and Fornasier and Schönlieb (2009) the resulting subspace minimization problems are solved by *oblique thresholding*, which is based on an iterative proximity map algorithm and the computation of a Lagrange multiplier by a fixed point iteration. In order to speed up the computation, in Langer et al. (2013) each subproblem is suggested to be solved by a *Bregmanized operator splitting – split Bregman* algorithm.

In practice in order to obtain an approximation of the subspace minimization problems of Algorithms 9 and 10, only a finite number of (inner) iterations of (39) can be performed. Nevertheless, the respective generated sequence $(u^n)_n$ of Algorithms 9 and 10 still satisfies the following convergence properties:

- (i) $J(u^n) \geq J(u^{n+1})$ for all $n \in \mathbb{N}$.
- (ii) $\lim_{n \rightarrow \infty} \|u^{n+1} - u^n\|_{L^2(\Omega)} = 0$.
- (iii) The sequence $(u^n)_n$ has subsequences that converge weakly in $L^2(\Omega)$ and $BV(\Omega)$.

Of course this does not imply the convergence of the sequence $(u^n)_n$ to a minimizer of J ; cf. Example 3. Nevertheless, it means that independently how accurately the subdomain problems are solved, the overall convergence is untouched. In a finite difference setting a similar estimate as the one in (38) can be shown (see Hintermüller and Langer (2014)), which again provides an upper bound of the distance between the obtained limit and a minimizer of the global problem.

Domain Decomposition Approach Based on the (Pre)Dual

We have seen that for the predual problem (14), the domain decomposition methods, which are guaranteed to converge to a minimizer of the original global problem, can be constructed. Based on these methods one can pursue the following strategy in order to design a domain decomposition method for problem (12): The domain decomposition methods in Algorithms 1, 2, 3, 4, and 5 are constituted by its subdomain problems. Then the dual problems of these subdomain problems are computed, yielding a sequence of subdomain problems of the primal problem. Due to predualization and dualization, the final constituted domain decomposition methods of the primal problem (12) look different than the splitting strategies presented in section “Basic Domain Decomposition Approach”. Using this idea in Langer and Gaspoz (2019) and Lee and Nam (2017) overlapping and non-overlapping

domain decomposition methods that converge to the minimizer of J with $T = I$ are designed. In particular, in Langer and Gaspoz (2019) overlapping domain decomposition methods in an infinite dimensional setting are proposed, while non-overlapping domain decomposition methods in a finite difference setting are constructed in Lee and Nam (2017). It turns out that in a discrete setting in the limit case when the overlapping size tends to 0, i.e., in the case of a non-overlapping decomposition, the approach in Langer and Gaspoz (2019) becomes the one in Lee and Nam (2017). In this vein in the following we concentrate on describing the derivation of the overlapping methods, as the construction of the non-overlapping methods runs analogously and is a special case of the overlapping method.

Derivation of the Methods

Let us focus now on the derivation of overlapping domain decomposition methods for solving (12) with $T = I$, i.e.,

$$\min_{u \in L^2(\Omega)} \frac{1}{2} \|u - g\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} |Du|. \quad (41)$$

Therefore, partition Ω into $M \in \mathbb{N}$ overlapping subdomains as in section “[Overlapping Domain Decomposition](#)”. For deriving the decomposition methods, we need to compute the dual problems of the subdomain problems of Algorithms 2 and 3. The subdomain problem in Ω_j , $j = \{1, \dots, M\}$, of these algorithms may be rewritten as

$$\arg \min \left\{ \frac{1}{2} \|\operatorname{div} \mathbf{v} + f\|_{L^2(\Omega)}^2 : \mathbf{v} \in H_0(\operatorname{div}, \Omega), |\mathbf{v}(x)|_{\ell^2} \leq \beta(x) \text{ f.a.a. } x \in \Omega \right\} \quad (42)$$

where $\beta := \lambda \theta_j$ with $\theta_j \geq 0$ defined as in (21), (22), and (23), $f = \operatorname{div} \left(\sum_{i < j} \mathbf{p}_i^{n+1} + \sum_{i > j} \theta_i \mathbf{p}^n \right) + g$ for Algorithm 2 and $f = \operatorname{div} \left(\sum_{i \neq j} \theta_i \mathbf{p}^n \right) + g$ for Algorithm 3 for any $n \geq 0$. If $\beta : \overline{\Omega} \rightarrow \mathbb{R}_0^+$, $\beta \in H^1(\Omega) \cap C(\overline{\Omega})$, $\|\nabla \beta\|_{L^\infty(\Omega)} < \infty$, and $\operatorname{supp}(\beta) \subseteq \overline{\Omega}$, then a Fenchel dual of (42) is given by

$$\arg \min_{u \in L^2(\Omega)} \left\{ \frac{1}{2} \|u - f\|_{L^2(\Omega)}^2 + \int_{\Omega} \beta |Du| \right\}, \quad (43)$$

whose minimizer is unique (Langer and Gaspoz 2019). Here and in the sequel, the expression $\int_{\Omega} \beta |Du|$ describes the integral of β on Ω with respect to the measure $|Du|$, where Du is the distributional gradient of u . Hence, the subdomain problems of our domain decomposition method are of the form (43). In order that (43) is well defined, a partition of unity function needs to have the following properties:

$$\sum_{i=1}^M \theta_j \equiv 1 \text{ and } \theta_j \geq 0 \text{ a.e. on } \overline{\Omega} \text{ for } j = 1, 2, \dots, M, \quad (44)$$

$$\text{supp } \theta_j \subset \overline{\Omega}_j \text{ for } j = 1, 2, \dots, M, \tag{45}$$

$$\theta_j \in H^1(\Omega) \cap C(\overline{\Omega}) \text{ and } \|\nabla \theta_j\|_{\mathbb{L}^\infty(\Omega)} < \infty \text{ for } j = 1, 2, \dots, M, \tag{46}$$

as $\beta = \lambda \theta_j$ for the subproblem in Ω_j , $j \in \{1, \dots, M\}$. In comparison to (21), (22), and (23) the additional requirements $\theta_j \in C(\overline{\Omega})$ and $\theta_j \geq 0$ a.e. on $\overline{\Omega}$ are needed such that $\int_{\Omega} \beta |Du|$ is well defined for $u \in L^2(\Omega)$. In the sequel of this section, we will only use a partition of unity function with the properties (44), (45), and (46) and denote it by $(\theta_j)_{j=1}^M$.

Now let us turn to the choice of f in (43). For this purpose we consider the basic successive algorithm (Algorithm 1) in domain Ω_M , where we have from the (predual) subdomain problem (42) that $f_M^{n+1} := f = \text{div} \left(\sum_{i=1}^{M-1} \mathbf{p}_i^{n+1} \right) + g$, where we introduced the subscript M and the superscript $n + 1$ to make the dependency of f on the domain and iteration visible. As we are designing a decomposition method for the L^2 -TV model, we do not want to compute in each iteration the dual variables \mathbf{p}_j^{n+1} , since then we could stick directly to Algorithm 1 of the predual problem. Note that for the solution u^* of (43) and a solution \mathbf{p}^* of (42), the relation (16) still holds, i.e.,

$$u^* = \text{div } \mathbf{p}^* + f.$$

Consequently, let \mathbf{p}_j^{n+1} be a solution of the predual subproblem in iteration $n + 1$, then $u_j^{n+1} = \text{div } \mathbf{p}_j^{n+1} + f_j^{n+1}$, $j = 1, \dots, M$. Plugging this into the definition of f_M^{n+1} , we obtain

$$f_M^{n+1} = \sum_{j=1}^{M-1} \left(u_j^{n+1} - f_j^{n+1} \right) + g.$$

This motivates the choice of f_j^{n+1} for all $j \in \{1, \dots, M\}$ as

$$f_j^{n+1} = \sum_{i>j} (u_i^n - f_i^n) + \sum_{i<j} (u_i^{n+1} - f_i^{n+1}) + g$$

for a successive algorithm; see Algorithm 11.

Let the partition of unity $(\theta_j)_{j=1}^M$ be as above, then we have the following convergence result (Langer and Gaspoz 2019).

Algorithm 11 Successive overlapping algorithm for (41)

Initialize: $u_j^0 (= 0) \in L^2(\Omega)$, $f_j^0 = 0 \in L^2(\Omega)$, $j = 1, \dots, M$
for $n = 0, 1, 2, \dots$ **do**
 for $j = 1, \dots, M$ **do**
 $f_j^{n+1} = \sum_{i>j} (u_i^n - f_j^n) + \sum_{i<j} (u_i^{n+1} - f_i^{n+1}) + g$
 $u_j^{n+1} = \arg \min_{u_j \in L^2(\Omega)} \frac{1}{2} \|u_j - f_j^{n+1}\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} \theta_j |Du_j|$
 end for
 $u^{n+1} = g + \sum_{j=1}^M u_j^{n+1} - f_j^{n+1} (= u_M^{n+1})$
end for

Theorem 4. Assume that $(f_j^n)_n$ is bounded in $L^2(\Omega)$ for $j = 1, \dots, M$, then Algorithm 11 generates a sequence $(u^n)_n$ which converges strongly in $L^2(\Omega)$ to a unique minimizer of (12) with $T = I$.

The strong convergence is due to the fact that $(\|u^n\|_{L^2(\Omega)})_n$ is monotonically decreasing. We remark that the boundedness assumption on $(f_j^n)_n$, $j = 1, \dots, M$, is essential for the convergence proof, but this assumption automatically holds in a finite dimensional setting, which is, for example, the situation when the considered problem is discretized.

The parallel version of Algorithm 11 is presented in Algorithm 12.

Algorithm 12 Parallel overlapping algorithm for (41)

Initialize: $v_j^0 = 0$ for $j = 1, \dots, M$
for $n = 0, 1, 2, \dots$ **do**
 $f_j^{n+1} = \sum_{i \neq j} v_i^n + g$, $j = 1, \dots, M$
 $u_j^{n+1} = \arg \min_{u_j \in L^2(\Omega)} \frac{1}{2} \|u_j - f_j^{n+1}\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} \theta_j |Du_j|$, $j = 1, \dots, M$
 $v_j^{n+1} = \frac{(M-1)v_j^n + u_j^{n+1} - f_j^{n+1}}{M}$, $j = 1, \dots, M$
 $u^{n+1} = g + \sum_{j=1}^M v_j^{n+1} (= \frac{\sum_{j=1}^M u_j^{n+1}}{M})$
end for

Note that here for the update of f_j^{n+1} an averaging (relaxation) is introduced, which is necessary for theoretical reasons in order to guarantee a similar convergence result as for the successive algorithm.

Theorem 5. Assume that $(f_j^n)_n$ is bounded in $L^2(\Omega)$ for $j = 1, \dots, M$, then Algorithm 12 generates a sequence $(u^n)_n$ which converges strongly in $L^2(\Omega)$ to a unique minimizer of (12) with $T = I$

Proof. Since $f_j^{n+1} = \sum_{i \neq j} v_i^n + g$ we get that

$$f_j^{n+1} - f_1^{n+1} = v_1^n - v_j^n \tag{47}$$

for all $j = 1, \dots, M$ and $n \geq 0$. Hence, this yields

$$f_M^{n+1} + \sum_{j=2}^{M-1} (f_j^{n+1} - f_1^{n+1}) = \sum_{j=1}^{M-1} v_j^n + g + \sum_{j=2}^{M-1} (v_1^n - v_j^n) = (M - 1)v_1^n + g$$

for $n \geq 0$. Due to the boundedness of $(f_j^{n+1})_n$ for $j = 1, \dots, M$ in $L^2(\Omega)$, also $(v_1^n)_n$ is bounded in $L^2(\Omega)$. Consequently by (47) the sequence $(v_j^n)_n$ for $j = 1, \dots, M$ is bounded in $L^2(\Omega)$.

The rest of the proof follows the lines of the proof of Langer and Gaspoz (2019, Theorem 2.12) by straightforwardly adjusting the arguments to a splitting into $M \in \mathbb{N}$ domains.

Subspace Minimization

Let us turn now to the question how to realize the subspace minimization problems of Algorithms 11 and 12 and restrict them to the respective subdomains. We consider, for example, the subspace minimization with respect to u_1 , i.e.,

$$u_1^{n+1} = \arg \min_{u_1 \in L^2(\Omega)} \frac{1}{2} \|u_1 - f_1^{n+1}\|_{L^2(\Omega)}^2 + \lambda \int_{\Omega} \theta_1 |Du_1|, \tag{48}$$

by anticipating that the arguments are analogue for the other subdomain problems. There are two different approaches on how to compute the solution of (48) by solving a minimization on Ω_1 only. These two approaches relate to “First optimize then discretize” and “First discretize then optimize,” where the optimization part allows to restrict the problem to the subdomain. Hence, the first approach restricts the minimization problem in an infinite dimensional setting before discretization, while the second approach first discretizes (48) and then restricts the optimization process to the subdomain Ω_1 .

First optimize then discretize The restriction of the subproblem is based on the following statement, cf. Langer and Gaspoz (2019, Lemma 2.2).

Lemma 1. *Let $u \in L^2(\Omega)$, $\theta : \overline{\Omega} \rightarrow \mathbb{R}_0^+$, $\theta \in H^1(\Omega) \cap C(\overline{\Omega})$, $\|\nabla \theta\|_{\mathbb{L}^\infty(\Omega)} < \infty$, $\text{supp}(\theta) \subseteq \overline{\Omega}$, and $K := \{p \in H_0(\text{div}, \Omega) : |p(x)|_{\ell^2} \leq \theta(x) \text{ f.a.a. } x \in \Omega\}$ then*

$$\int_{\Omega} \theta |Du| = \int_{\text{supp}(\lambda)} \theta |Du|.$$

Proof. Let $\Omega_0 := \Omega \setminus \text{supp}(\theta)$, then we get

$$\begin{aligned} \int_{\text{supp}(\theta)} \theta |Du| &= \int_{\Omega \setminus \Omega_0} \theta |Du| = \sup_{\mathbf{p} \in \mathcal{K}(1, C_0(\Omega \setminus \Omega_0, \mathbb{R}^2))} \langle \theta Du, \mathbf{p} \rangle_{C_0(\Omega \setminus \Omega_0, \mathbb{R}^2)' \times C_0(\Omega \setminus \Omega_0, \mathbb{R}^2)} \\ &= \sup_{\mathbf{p} \in \mathcal{K}(1, C_0(\Omega, \mathbb{R}^2))} \langle \theta Du, \mathbf{p} \rangle_{C_0(\Omega, \mathbb{R}^2)' \times C_0(\Omega, \mathbb{R}^2)} \\ &= \int_{\Omega} \theta |Du|, \end{aligned}$$

since $\theta \in C(\overline{\Omega})$ and $\theta(x) = 0$ f.a.a. $x \in \Omega_0$, where $\mathcal{K}(\theta, C_0(\Omega, \mathbb{R}^2)) := \{\mathbf{p} \in C_0(\Omega, \mathbb{R}^2) : |\mathbf{p}(x)|_{\ell^2} \leq \theta(x) \text{ f.a.a. } x \in \Omega\}$ with $C_0(\Omega, \mathbb{R}^2)$ denoting the space of \mathbb{R}^2 -valued continuous functions with compact support in Ω .

Utilizing Lemma 1 one can show that the minimizer of (48) can be computed by solving a minimization problem in Ω_1 only.

Proposition 1. *The solution $u_1^{n+1} \in L^2(\Omega)$ of the minimization problem in (48) is given by*

$$u_1^{n+1} = \begin{cases} f_1^{n+1} & \text{in } \Omega \setminus \Omega_1 \\ \arg \min_{u_1 \in L^2(\Omega_1)} \frac{1}{2} \|u_1 - f_1^{n+1}\|_{L^2(\Omega_1)}^2 + \lambda \int_{\Omega_1} \theta_1 |Du_1| & \text{in } \Omega_1. \end{cases} \quad (49)$$

Proof. Since the partition of unity is such that $\text{supp}(\theta_1) \subseteq \Omega_1$, we have due to Lemma 1 that $\int_{\Omega} \theta_1 |Du_1| = \int_{\Omega_1} \theta_1 |Du_1|$. Hence, by the optimality of u_1^{n+1} we get $f_1^{n+1} - u_1^{n+1} \in \partial \lambda \int_{\Omega_1} \theta_1 |Du_1^{n+1}|$. That is,

$$(f_1^{n+1} - u_1^{n+1}, v - u_1^{n+1}) + \lambda \int_{\Omega_1} \theta_1 |Du_1^{n+1}| \leq \lambda \int_{\Omega_1} \theta_1 |Dv| \quad \forall v \in L^2(\Omega).$$

This inequality holds if

$$\begin{aligned} \int_{\Omega \setminus \Omega_1} (f_1^{n+1} - u_1^{n+1})(v - u_1^{n+1}) dx &\leq 0 \quad \text{and} \\ \int_{\Omega_1} (f_1^{n+1} - u_1^{n+1})(v - u_1^{n+1}) dx + \lambda \int_{\Omega_1} \theta_1 |Du_1^{n+1}| &\leq \lambda \int_{\Omega_1} \theta_1 |Dv| \end{aligned}$$

for all $v \in L^2(\Omega)$. Hence, u_1^{n+1} fulfilling these two latter inequalities is a minimizer of the subspace minimization problem (48). By the uniqueness of the minimizer, we therefore obtain (49).

Due to the presence of the function θ_1 , the usual total variation minimization techniques cannot be used directly to compute a minimizer of the optimization problem in (49), but may be used after being adapted to locally weighted total variation minimization. We note that the minimization of locally weighted total variation has been already considered in the literature (see, for example, Langer (2017a)), where an algorithm for solving a minimization problem of the type (48) is already presented. An alternative method modifying the split Bregman algorithm (Goldstein and Osher 2009) to locally weighted total variation minimization is proposed in Langer and Gaspoz (2019). Utilizing one of these methods for a practical implementation would then require a suitable discretization.

First discretize then optimize Since Algorithms 11 and 12 are designed for an overlapping splitting, let Ω^h be a discrete rectangular image domain containing $N_1 \times N_2$ pixels, $N_1, N_2 \in \mathbb{N}$, and decomposed into overlapping subdomains Ω_i^h , $i = 1, \dots, M$ such that $\Omega^h = \bigcup_{i=1}^M \Omega_i^h$ and for any $i \in \{1, \dots, M\}$ there exists at least one $j \in \{1, \dots, M\} \setminus \{i\}$ such that $\Omega_i^h \cap \Omega_j^h \neq \emptyset$. Moreover, we use the finite difference discretization introduced in section “[Finite Difference Setting](#)”. Then the discretized version of (48) is written as

$$u_1^{h,n+1} = \arg \min_{u_1^h \in X} \frac{1}{2} \|u_1^h - f_1^{h,n+1}\|_X^2 + \lambda \sum_{x \in \Omega^h} \theta_1^h(x) |\nabla_{\Omega}^h u_1^h(x)|_{\ell^2}, \quad (50)$$

where $\theta_1^h \in X$ is the discrete version of the above introduced θ_1 satisfying (44), (45), and (46). Since $\theta_1^h(x) = 0$ for all $x \in \Omega^h \setminus \Omega_1^h$ we can write the above minimization problem as

$$u_1^{h,n+1} = \begin{cases} f_1^{h,n+1} & \text{in } \Omega^h \setminus \Omega_1^h \\ \arg \min_{u_1^h|_{\Omega_1^h} \in X_1} \frac{1}{2} \|u_1^h - f_1^{h,n+1}\|_{X_1}^2 + \lambda \sum_{x \in \Omega_1^h} \theta_1^h(x) |\nabla_{\Omega}^h u_1^h(x)|_{\ell^2} & \text{in } \Omega_1^h, \end{cases} \quad (51)$$

where $u_1^h \in X$ is such that $u_1^h(x) = f_1^{h,n+1}(x)$ for $x \in \Omega^h \setminus \Omega_1^h$. Hence, in order to obtain $u_1^{h,n+1}$, only a minimization problem in Ω_1^h has to be solved, i.e.,

$$\arg \min_{u_1^h|_{\Omega_1^h} \in X_1} \frac{1}{2} \|u_1^h - f_1^{h,n+1}\|_{X_1}^2 + \lambda \sum_{x \in \Omega_1^h} \theta_1^h(x) |(\nabla_{\Omega}^h u_1^h)|_{\Omega_1^h}(x)|_{\ell^2}.$$

Note that ∇_{Ω}^h is not a local operator, but nonetheless quite local, i.e., it affects only the neighboring pixels. Hence, by carefully considering the restriction to Ω_1^h (i.e., we use Dirichlet boundary conditions on the interface between Ω_1^h and $\Omega^h \setminus \Omega_1^h$), $u_{1,\Omega_1^h}^{h,n+1} \in X_1$ is obtained by solving an optimization in Ω_1^h only. Consequently locally weighted total variation minimization techniques may be used by carefully

adjusting the gradient operator of the total variation term. An implementation based on the split Bregman algorithm is presented in Langer and Gaspoz (2019), which allows to obtain $u_{1,\Omega_1^h}^{h,n+1}$ by solving a linear system only of size $|\Omega_1^h|$.

Let us mention that all the results presented in this section hold symmetrically for the minimization with respect to u_i , $i = 2, \dots, M$ and that the notations should be just adjusted accordingly.

Limit Case: Non-overlapping Decomposition

We remark that in a discrete setting the continuity assumption on θ_j^h , for $j = 1, \dots, M$, is obsolete. Hence, we may let the overlapping size go to 0, yielding a non-overlapping decomposition. That is,

$$\theta_j^h(x) = \begin{cases} 1 & \text{if } x \in \Omega_i^h \\ 0 & \text{else} \end{cases}$$

for $j = 1, \dots, M$. Then the subspace minimization problems read as

$$\arg \min_{u_j \in X_j} \frac{1}{2} \|u_j^h - f_j^{h,n+1}\|_{X_j}^2 + \lambda \sum_{x \in \Omega_j^h} |\nabla_{\Omega}^h u_j^h(x)|_{\ell^2},$$

$j = 1, \dots, M$. Thus, in a discrete setting, using this discretization and restriction approach, in the limit case of a non-overlapping decomposition, Algorithms 11 and 12 become the successive domain decomposition (block Gauss-Seidel) and parallel domain decomposition (relaxed block Jacobi) method of Lee and Nam (2017), respectively. Moreover, in Lee and Nam (2017) these methods have been extended to (12) with $T \neq I$ by using the surrogate functional idea (cf. section “Subspace Minimization”), on J^h , i.e., for $u^h, a^h \in X$ we define

$$J^{h,s}(u^h, a^h) := J^h(u^h) + \frac{1}{2} \left(\delta \|u^h - a^h\|_X^2 - \|T^h(u^h - a^h)\|_X^2 \right),$$

where $\delta > \|T^h\|^2$. Then we have

$$\begin{aligned} \arg \min_{u^h \in X} J^{h,s}(u^h, a^h) &= \arg \min_{u^h \in X} \frac{1}{2} \|u^h - \frac{1}{\delta} (T^{h*} g^h + (\delta - T^{h*} T^h) a^h)\|_X^2 \\ &\quad + \frac{\lambda}{\delta} \sum_{x \in \Omega^h} |\nabla u^h(x)|_{\ell^2} \end{aligned}$$

and an approximation of the minimizer of J is obtained by iteratively minimizing

$$u^{h,0} = 0, \quad u^{h,n+1} = \arg \min_{u^h \in X} J^{h,s}(u^h, u^{h,n}) \quad n \geq 0. \quad (52)$$

Since in each iteration we have to solve a problem which is of the same type as (12) with $T = I$, we may use Algorithm 11 or Algorithm 12 now in a non-overlapping and finite difference setting to speed up the solution process, leading to Algorithms 13 and 14.

Algorithm 13 Successive non-overlapping algorithm for (12)

Initialize: $u_j^{h,0} := 0, v_j^{h,0} := 0$ for $j = 1, \dots, M$
for $n = 0, 1, 2, \dots$ **do**
 $f^{h,n+1} = \frac{1}{\delta}(T^{h*}g^h + (\delta - T^{h*}T^h)u^{h,n}, q_i^{h,0} = v_j^{h,n}$ for $j = 1, \dots, M$ and $k = 1$
while $J^{h,s}(f^{h,n+1} - \sum_{j=1}^M q_j^{h,k}, u^{h,n}) \leq J^h(u^{h,n})$ **do**
for $j = 1, \dots, M$ **do**
 $f_j^{h,k} = f^{h,n+1} - \sum_{i>j} q_i^{h,k-1} - \sum_{i<j} q_i^{h,k}$ in Ω_j^h
 $u_j^{h,k} = \arg \min_{u_j^h \in X} \frac{1}{2} \|u_j^h - f_j^{h,k}\|_X^2 + \frac{\lambda}{\delta} \sum_{x \in \Omega_j^h} |\nabla u_j^h(x)|$
 $q_j^{h,k} = f_j^{h,k} - u_i^{h,k}$
 $k = k + 1$
end for
end while
 $u^{h,n+1} = f^{h,n+1} - \sum_{j=1}^M q_j^{h,k}$ and $v_j^{h,n+1} = q_j^{h,k}$ for $j = 1, \dots, M$
end for

Algorithm 14 Parallel non-overlapping algorithm for (12)

Initialize: $u_j^{h,0} := 0, v_j^{h,0} := 0$ for $j = 1, \dots, M$
for $n = 0, 1, 2, \dots$ **do**
 $f^{n+1} = \frac{1}{\delta}(T^{h*}g^h + (\delta - T^{h*}T^h)u^{h,n}, q_i^{h,0} = v_j^{h,n}$ for $j = 1, \dots, M$ and $k = 1$
while $J^{h,s}(f^{h,n+1} - \sum_{j=1}^M q_j^{h,k}, u^{h,n}) \leq J^h(u^{h,n})$ **do**
for $j = 1, \dots, M$ **do**
 $f_j^{h,k} = g^h - \sum_{i \neq j} q_i^{h,k-1}$ in Ω_j^h
 $u_j^{h,k} = \arg \min_{u_j^h \in X} \frac{1}{2} \|u_j^h - f_j^{h,k}\|_X^2 + \frac{\lambda}{\delta} \sum_{x \in \Omega_j^h} |\nabla u_j^h(x)|$
 $q_j^{h,k} = \frac{(M-1)q_j^{h,k-1} + f_j^{h,k} - u_j^{h,k}}{M}$
 $k = k + 1$
end for
end while
 $u^{h,n+1} = f^{h,n+1} - \sum_{j=1}^M q_j^{h,k}$ and $v_j^{h,n+1} = q_j^{h,k}$ for $j = 1, \dots, M$
end for

In Lee and Nam (2017) it is shown for $M = 2$ that these algorithms produce sequences $(u^n)_n$ whose accumulation points are minimizers of J^h .

Conclusion

Domain decomposition methods are known to be one of the most successful methods to construct efficient solvers for large-scale problems. Nevertheless, only quite recently such methods are developed for total variation minimization. Therefore, the research in this direction is far from being complete, as only very little is known yet. We summarize that the domain decomposition algorithms for total variation minimization with a theoretical guarantee to convergence to the minimizer of the global problem are till now given for (i) the discrete predual problem with a non-overlapping decomposition using finite differences (Hintermüller and Langer 2015) or finite elements (Lee et al. 2019b; Lee and Park 2019b), (ii) the continuous predual problem with an overlapping decomposition (Chang et al. 2015), (iii) the discrete primal problem with a non-overlapping decomposition (Lee and Nam 2017), (iv) and the continuous primal problem with an overlapping decomposition (Langer and Gaspoz 2019). This list of achievements indicates that constructing overlapping domain decomposition methods in an infinite dimensional setting seems easier than non-overlapping domain decomposition methods. A reason for this may be guessed when one looks at the Poisson problem (see section “[Basic Idea of Domain Decomposition](#)”). There one sees that in order to construct convergent non-overlapping methods, the subdomain problems differ in each subdomain due to the interface conditions, while in the overlapping situation all subdomain problems are of the same type. This ostensible flexibility in creating subdomain problems for a non-overlapping splitting may lead to additional difficulties for problems where the solution is discontinuous, as the interface conditions are not clear. In particular, neither of the interface conditions in (2) are suitable.

For the domain decomposition methods tackling the predual problem (14), not only the convergence but also the convergence order is known. We note that the decomposition methods for the continuous problems only cover the image denoising case, i.e., the L^2 -TV model with $T = I$, while the methods for the discretized objectives can also handle image inpainting and image segmentation problems. The primal-dual approach in Lee et al. (2019a) is even successfully applied to image deblurring. Of course, by using the surrogate idea (also called operator splitting (Combettes and Wajs 2005)), the L^2 -TV model can be cast to an image denoising type of problem for any operator T . But it is in general unclear how accurately the solution of the domain decomposition iteration has to be computed in order to guarantee the convergence of the outer surrogate iteration. Interesting tasks arising, for example, in medical imaging where T might be a sampled Fourier transform or Radon transform, which are very global operators, have not yet been thoroughly considered.

References

- Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Probl.* **10**(6), 1217–1229 (1994)
- Alliney, S.: A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. *IEEE Trans. Signal Process.* **45**(4), 913–917 (1997)
- Ambrosio, L., Fusco, N., Pallara, D.: *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford Mathematical Monographs. The Clarendon Press/Oxford University Press, New York (2000)
- Attouch, H., Buttazzo, G., Michaille, G.: *Variational Analysis in Sobolev and BV Spaces*. MOS-SIAM Series on Optimization, 2nd edn. Society for Industrial and Applied Mathematics (SIAM)/Mathematical Optimization Society, Philadelphia (2014). Applications to PDEs and optimization
- Aubert, G., Aujol, J.-F.: A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.* **68**(4), 925–946 (2008)
- Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**(1), 183–202 (2009)
- Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York (2014)
- Burger, M., Sawatzky, A., Steidl, G.: First order algorithms in variational image processing. In: *Splitting Methods in Communication, Imaging, Science, and Engineering*. Scientific Computation, pp. 345–407. Springer, Cham (2016)
- Cai, J.-F., Chan, R.H., Nikolova, M.: Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise. *Inverse Probl. Imaging* **2**(2), 187–204 (2008)
- Calatroni, L., De Los Reyes, J.C., Schönlieb, C.-B.: Infimal convolution of data discrepancies for mixed noise removal. *SIAM J. Imaging Sci.* **10**(3), 1196–1233 (2017)
- Carstensen, C.: Domain decomposition for a non-smooth convex minimization problem and its application to plasticity. *Numer. Linear Algebra Appl.* **4**(3), 177–190 (1997)
- Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* **20**(1–2), 89–97 (2004). Special issue on mathematics and image analysis
- Chambolle, A., Pock, T.: A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Chambolle, A., Pock, T.: A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI J. Comput. Math.* **1**, 29–54 (2015)
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
- Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T.: An introduction to total variation for image analysis. *Theor. Found. Numer. Methods Sparse Recovery* **9**, 263–340 (2010)
- Chan, T.F., Mathew, T.P.: Domain decomposition algorithms. In: *Acta Numerica*, pp. 61–143. Cambridge University Press, Cambridge (1994)
- Chan, T.F., Shen, J.J.: *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia (2005)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006)
- Chang, H., Zhang, X., Tai, X.-C., Yang, D.: Domain decomposition methods for nonlocal total variation image restoration. *J. Sci. Comput.* **60**(1), 79–100 (2014)
- Chang, H., Tai, X.-C., Wang, L.-L., Yang, D.: Convergence rate of overlapping domain decomposition methods for the Rudin–Osher–Fatemi model based on a dual formulation. *SIAM J. Imaging Sci.* **8**(1), 564–591 (2015)
- Chen, K., Tai, X.-C.: A nonlinear multigrid method for total variation minimization from image restoration. *J. Sci. Comput.* **33**(2), 115–138 (2007)
- Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **4**(4), 1168–1200 (electronic) (2005)

- Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**(11), 1413–1457 (2004)
- Daubechies, I., Teschke, G., Vese, L.: Iteratively solving linear inverse problems under general convex constraints. *Inverse Probl. Imaging* **1**(1), 29–46 (2007)
- Dolean, V., Jolivet, P., Nataf, F.: An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, vol. 144. SIAM, Philadelphia (2015)
- Duan, Y., Tai, X.-C.: Domain decomposition methods with graph cuts algorithms for total variation minimization. *Adv. Comput. Math.* **36**(2), 175–199 (2012)
- Duan, Y., Chang, H., Tai, X.-C.: Convergent non-overlapping domain decomposition methods for variational image segmentation. *J. Sci. Comput.* **69**(2), 532–555 (2016)
- Fornasier, M.: Domain decomposition methods for linear inverse problems with sparsity constraints. *Inverse Probl. Int. J. Theory Pract. Inverse Probl. Inverse Methods Comput. Inversion Data* **23**(6), 2505–2526 (2007)
- Fornasier, M., Schönlieb, C.-B.: Subspace correction methods for total variation and l_1 -minimization. *SIAM J. Numer. Anal.* **47**(5), 3397–3428 (2009)
- Fornasier, M., Langer, A., Schönlieb, C.-B.: Domain decomposition methods for compressed sensing. In: Proceedings of the International Conference of SampTA09, Marseilles, arXiv preprint arXiv:0902.0124 (2009)
- Fornasier, M., Langer, A., Schönlieb, C.-B.: A convergent overlapping domain decomposition method for total variation minimization. *Numerische Mathematik* **116**(4), 645–685 (2010)
- Fornasier, M., Kim, Y., Langer, A., Schönlieb, C.: Wavelet decomposition method for L_2 /TV-image deblurring. *SIAM J. Imaging Sci.* **5**(3), 857–885 (2012)
- Getreuer, P., Tong, M., Vese, L.A.: A variational model for the restoration of mr images corrupted by blur and rician noise. In: International Symposium on Visual Computing, pp. 686–698. Springer (2011)
- Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2009)
- Giusti, E.: Minimal Surfaces and Functions of Bounded Variation. Monographs in Mathematics, vol. 80. Birkhäuser Verlag, Basel (1984)
- Goldstein, T., Osher, S.: The split Bregman method for L_1 -regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
- Hintermüller, M., Kunisch, K.: Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM J. Appl. Math.* **64**(4), 1311–1333 (2004)
- Hintermüller, M., Langer, A.: Subspace correction methods for a class of nonsmooth and nonadditive convex variational problems with mixed L^1/L^2 data-fidelity in image processing. *SIAM J. Imaging Sci.* **6**(4), 2134–2173 (2013)
- Hintermüller, M., Langer, A.: Surrogate functional based subspace correction methods for image processing. In: Domain Decomposition Methods in Science and Engineering XXI, pp. 829–837. Springer, Cham (2014)
- Hintermüller, M., Langer, A.: Non-overlapping domain decomposition methods for dual total variation based image denoising. *J. Sci. Comput.* **62**(2), 456–481 (2015)
- Hintermüller, M., Rautenberg, C.: On the density of classes of closed convex sets with pointwise constraints in sobolev spaces. *J. Math. Anal. Appl.* **426**(1), 585–593 (2015)
- Hintermüller, M., Rautenberg, C.N.: Optimal selection of the regularization function in a weighted total variation model. Part I: Modelling and theory. *J. Math. Imaging Vis.* **59**(3), 498–514 (2017)
- Ito, K., Kunisch, K.: Lagrange Multiplier Approach to Variational Problems and Applications, vol. 15. SIAM, Philadelphia (2008)
- Langer, A.: Automated parameter selection for total variation minimization in image restoration. *J. Math. Imaging Vis.* **57**(2), 239–268 (2017a)
- Langer, A.: Automated parameter selection in the L^1 - L^2 -TV model for removing Gaussian plus impulse noise. *Inverse Probl.* **33**(7), 74002 (2017b)
- Langer, A.: Locally adaptive total variation for removing mixed Gaussian–impulse noise. *Int. J. Comput. Math.* **96**(2), 298–316 (2019)

- Langer, A., Gaspoz, F.: Overlapping domain decomposition methods for total variation denoising. *SIAM J. Numer. Anal.* **57**(3), 1411–1444 (2019)
- Langer, A., Osher, S., Schönlieb, C.-B.: Bregmanized domain decomposition for image restoration. *J. Sci. Comput.* **54**(2–3), 549–576 (2013)
- Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by poisson noise. *J. Math. Imaging Vis.* **27**(3), 257–263 (2007)
- Lee, C.-O., Nam, C.: Primal domain decomposition methods for the total variation minimization, based on dual decomposition. *SIAM J. Sci. Comput.* **39**(2), B403–B423 (2017)
- Lee, C.-O., Park, J.: Fast nonoverlapping block Jacobi method for the dual Rudin–Osher–Fatemi model. *SIAM J. Imaging Sci.* **12**(4), 2009–2034 (2019a)
- Lee, C.-O., Park, J.: A finite element nonoverlapping domain decomposition method with lagrange multipliers for the dual total variation minimizations. *J. Sci. Comput.* **81**(3), 2331–2355 (2019b)
- Lee, C.-O., Lee, J.H., Woo, H., Yun, S.: Block decomposition methods for total variation by primal–dual stitching. *J. Sci. Comput.* **68**(1), 273–302 (2016)
- Lee, C.-O., Nam, C., Park, J.: Domain decomposition methods using dual conversion for the total variation minimization with L^1 fidelity term. *J. Sci. Comput.* **78**(2), 951–970 (2019a)
- Lee, C.-O., Park, E.-H., Park, J.: A finite element approach for the dual Rudin–Osher–Fatemi model and its nonoverlapping domain decomposition methods. *SIAM J. Sci. Comput.* **41**(2), B205–B228 (2019b)
- Lions, J.L.: *Optimal Control of Systems Governed by Partial Differential Equations. Die Grundlehren der mathematischen Wissenschaften, vol. 170.* Springer (1971)
- Lions, P.-L.: On the Schwarz alternating method. I. In: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris*, pp. 1–42 (1988)
- Marini, L.D., Quarteroni, A.: A relaxation procedure for domain decomposition methods using finite elements. *Numerische Mathematik* **55**(5), 575–598 (1989)
- Mathew, T.: *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations, vol. 61.* Springer Science & Business Media, Berlin (2008)
- Nikolova, M.: Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM J. Numer. Anal.* **40**(3), 965–994 (electronic) (2002)
- Nikolova, M.: A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**(1–2), 99–120 (2004)
- Peyré, G., Bougleux, S., Cohen, L.: Non-local regularization of inverse problems. In: *European Conference on Computer Vision*, pp. 57–68. Springer (2008)
- Pock, T., Unger, M., Cremers, D., Bischof, H.: Fast and exact solution of total variation models on the gpu. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8. IEEE (2008)
- Quarteroni, A., Valli, A.: *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, New York (1999)
- Raviart, P.-A., Thomas, J.-M.: A mixed finite element method for 2-nd order elliptic problems. In: *Mathematical Aspects of Finite Element Methods*, pp. 292–315. Springer, Berlin (1977)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1), 259–268 (1992)
- Schönlieb, C.-B.: Total variation minimization with an H^{-1} constraint. *CRM Ser.* **9**, 201–232 (2009)
- Schwarz, H.A.: Über einige Abbildungsaufgaben. *Journal für die reine und angewandte Mathematik* **1869**(70), 105–120 (1869)
- Smith, B., Bjorstad, P., Gropp, W.: *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, Dordrecht (2004)
- Tai, X.-C.: Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numerische Mathematik* **93**(4), 755–786 (2003)
- Tai, X.-C., Tseng, P.: Convergence rate analysis of an asynchronous space decomposition method for convex minimization. *Math. Comput.* **71**(239), 1105–1135 (2002)
- Tai, X.-C., Xu, J.: Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comput.* **71**(237), 105–124 (2002)

- Toselli, A., Widlund, O.: *Domain Decomposition Methods: Algorithms and Theory*, vol. 34. Springer Science & Business Media, Dordrecht (2006)
- Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **109**(3), 475–494 (2001)
- Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. *Math. Prog.* **117**(1–2), 387–423 (2009)
- Vonesch, C., Unser, M.: A fast multilevel algorithm for wavelet-regularized image restoration. *IEEE Trans. Image Process.* **18**(3), 509–523 (2009)
- Warga, J.: Minimizing Certain Convex Functions. *J. Soc. Indust. Appl. Math.* **11**, 588–593 (1963)
- Wright, S.J.: Coordinate descent algorithms. *Math. Prog.* **151**(1), 3–34 (2015)
- Wu, C., Tai, X.-C.: Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. *SIAM J. Imaging Sci.* **3**(3), 300–339 (2010)
- Xu, J., Tai, X.-C., Wang, L.-L.: A two-level domain decomposition method for image restoration. *Inverse Probl. Imaging* **4**(3), 523–545 (2010)
- Xu, J., Chang, H.B., Qin, J.: Domain decomposition method for image deblurring. *J. Comput. Appl. Math.* **271**, 401–414 (2014)
- Zhang, X., Burger, M., Bresson, X., Osher, S.: Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.* **3**(3), 253–276 (2010)



Fast Numerical Methods for Image Segmentation Models

12

Noor Badshah

Contents

Introduction	428
Mathematical Models for Image Segmentation	429
Two-Phase Segmentation Models	429
Snakes: Active Contour Model	429
Geodesic Active Contour Model (GAC)	430
Chan-Vese Model	431
Fast Numerical Methods:	433
Multigrid Solver for Solving a Class of Variational Problems with Application to Image Segmentation	446
Sobolev Gradient Minimization of Curve Length in Chan-Vese Model	449
Multiphase Image Segmentation	452
Multigrid Method for Multiphase Segmentation Model	452
Multigrid Method with Typical and Modified Smoother	454
Local Fourier Analysis and a Modified Smoother	455
Convex Multiphase Image Segmentation Model	460
A Three-Stage Approach for Multiphase Segmentation Degraded Color Images	466
Stage 2: Dimension Lifting with Secondary Color Space	468
Selective Segmentation Models	469
Image Segmentation Under Geometrical Conditions	469
Active Contour-Based Image Selective Model	471
Dual-Level Set Selective Segmentation Model	475
One-Level Selective Segmentation Model	477
Reproducible Kernel Hilbert Space-Based Image Segmentation	479
An Optimization-Based Multilevel Algorithm for Selective Image Segmentation Models	485

N. Badshah (✉)

Department of Basic Sciences, University of Engineering and Technology, Peshawar, Pakistan

Machine/Deep Learning Techniques for Image Segmentation.....	492
Machine Learning with Region-Based Active Contour Models in	
Medical Image Segmentation.....	492
ResBCU-Net: Deep Learning Approach for Segmentation of Skin Images.....	494
Conclusion.....	498
References.....	499

Abstract

In this chapter, three different types of segmentation problems are studied, namely, two-phase segmentation problems, multiphase segmentation problems, and selective segmentation problems. Three types of numerical methods are discussed here as well. Some of them are time marching schemes, multigrid methods, and multilevel methods. Two types of minimization techniques are discussed, like L^2 gradient minimization and Sobolev gradient-based minimization techniques. At the end two deep/machine learning approaches for segmentation of images are also presented.

Keywords

Image segmentation · Euler-Lagrange’s equations · Sobolev gradient · Finite differences · Machine learning · Deep learning

Introduction

Image segmentation is one of the fundamental tasks in image analysis and computer vision. The purpose of image segmentation is to partition a given image into different meaningful regions based on the intensity homogeneity, pattern similarities, colors similarities, etc. The goal of image segmentation is to represent an image in such a way that could be easily analyzed. There are two main concerns related to image segmentation: (i) modeling image segmentation problems and (ii) fast and advanced numerical methods for the solution of partial differential equations arising from the minimization of these models. There are many algorithms/models present in the literature for the solution of image segmentation problems. Among these, some of them use edge or region information of the image for segmentation purpose. The most basic edge-based model is the geodesic snake model (Kass et al. 1988; Caselles et al. 1997), which is based on edge information in the image, and a gradient flow is used as a stopping term to get correct boundaries with sudden changes in the gradient for attracting the contour to the object boundary. The Chan-Vese model (Chan and Vese 2001) is based on the variation in regions. For that purpose it uses region statistics as a stopping criterion.

The Allen-Cahn (AC) equation was originally introduced as a phenomenological model for antiphase domain coarsening in a binary alloy (Allen and Cahn 1979). This equation can be used to model flow problems based on mean curvature. This type of flow is one of the important element for active contour-based

image segmentation models. For these types of methods, there exists a very fast computational method such as multigrid method (Badshah and Chen 2008). This chapter is dedicated to the minimization techniques of various models developed for the segmentation of images, which leads to a highly nonlinear partial differential equation. Some well-known numerical methods for the solution of these partial differential equations have been discussed.

Mathematical Models for Image Segmentation

Image segmentation links low-level vision with high-level vision. It is the process of partitioning an image into a collection of objects which can, later on, be used for performing high-level tasks like object detection, tracking, recognition, etc. The current section is about the existing mathematical models developed for image segmentation. Active contour models have attained much attention in image segmentation nowadays. These models for segmentation of images are divided into two groups, namely, (i) edge-based segmentation models and (ii) region-based segmentation models. In the next section, edge-based active models are discussed in detail.

Two-Phase Segmentation Models

Snakes: Active Contour Model

A snake is an energy-based active contour model which minimizes the deformable curve combined with some constraints and/or drag or pull forces that will pull the contour toward object boundaries, whereas the internal energies will resist the deformation in the contour. The first active contour model was developed by Kass et al. (1988). This type of model locates abrupt changes in the intensity through the deformation of a curve Υ in the image z . Those type of abrupt changes in the intensities usually occur at the edges of objects in an image z . The energy functional of the snake model has external and internal forces. The image energy/force is responsible to push the contour/snake toward image features like lines, edges, etc. Whereas the internal energy works for the smoothness of the contour, the external energy pulls/draggs the contour/snake toward the desired boundary of the object (local minima of the functional). For a parametric planar curve $\Upsilon(p) = (x(p), y(p)) \in \Omega, 0 \leq p \leq 1$, the following energy functional is proposed:

$$\begin{aligned}
 F^K(\Upsilon(p)) = & \varpi \int_0^1 \left| \frac{\partial \Upsilon(p)}{\partial p} \right|^2 dp + \beta \int_0^1 \left| \frac{\partial^2 \Upsilon(p)}{\partial^2 p} \right|^2 dp \\
 & + \lambda \int_0^1 e^2(\nabla(z * K_\sigma)(\Upsilon(p))) dp,
 \end{aligned} \tag{1}$$

where $\varpi > 0, \beta > 0$ and $\lambda >$ are the trade-off parameters. Also e is the edge detector function and is given by the following:

$$e(\nabla(z * K_\sigma)) = \frac{1}{1 + \gamma|\nabla(z * K_\sigma)|^2}, \tag{2}$$

where $K_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{(x - \mu_x)^2 + (y - \mu_y)^2}{2\sigma^2}\right)$ is the well-known Gaussian kernel and γ is a positive parameter. F^K is nonconvex functional (Kass et al. 1988) and can be easily stuck at local minima. The local minima of F^K can be the solution of the following Euler-Lagrange’s equation:

$$-\varpi \frac{\partial^2 \gamma}{\partial p^2} + \beta \frac{\partial^4 \gamma}{\partial p^4} + \lambda \nabla e^2 = 0. \tag{3}$$

The numerical solution of this fourth-order partial differential equation can be found by using finite difference method (Kass et al. 1988).

Geodesic Active Contour Model (GAC)

In 1997, Caselles et al. proposed another edge-based model by using a new type of curve parametrization. This is an improvement in snake energy functional (Kass et al. 1988). The energy functional of the GAC model is given by the following:

$$F^C(\gamma(p)) = \int_0^1 e(|\nabla z(\gamma(p))|)|\gamma'(p)|dp. \tag{4}$$

Given that $L(\gamma)$ represents the Euclidean length of the moving contour γ and since $L(\gamma) = \int_0^1 |\gamma'(p)|dp = \int_0^{L(\gamma)} ds$, where ds is the Euclidean length element, Eq. (4) may be written as follows:

$$F^C(\gamma(p)) = \int_0^{L(\gamma)} e(|\nabla z(\gamma(p))|)ds \tag{5}$$

This energy functional introduces a new length through weighted Euclidean differential length ds by the edge detector e which uses edge information (Aubert and Kornprobst 2002). The function e is the same as given in (2). The equivalence between minimizing F^C and minimizing F^K at $\beta = 0$ was studied in Caselles et al. (1997). Hence the direction for which F^C decreases most rapidly provides us the following minimization flow: more details of its derivation can be found in Caselles et al. (1997):

$$\frac{\partial \gamma}{\partial t} = e\kappa \vec{N} - (\nabla e \cdot \vec{N})\vec{N}, \tag{6}$$

where κ represents curvature and \mathcal{N} is the unit normal vector. This equation leads toward the optimal length of the contour. The steady-state solution of (6) will be the solution of Euler-Lagrange's equation for the energy functional given in Eq. (5). By introducing level set idea, the evolution equation takes the following form:

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \left(\nabla \cdot \left(e \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu_1 e \right), \quad (7)$$

ϕ is a level set function and the contour \mathcal{Y} is the zero level set $\phi(x, y) = 0$. A balloon term $\nu_1 e$, $\nu_1 > 0$ is included to speed up the convergence.

These models are based on the edge detector e which uses the gradient of the image so these models can only detect objects whose boundaries are defined by gradient. Also, in practice, the discrete gradients are bounded, and hence the stopping function e may not vanish on the boundaries, and the contour may leak through the image edges (Chan and Vese 2001). These models may not work very well in noisy images.

Chan-Vese Model

In 2001, Chan and Vese proposed a region-based energy functional which uses data fitting statistics as a stopping process and is a special case of piecewise constant Mumford-Shah model (Mumford and Shah 1989). Let z be the known bounded function (image data) and assume that z has two regions (say foreground and background) of approximately constant intensities z_i and z_o . Assume that the object to be detected is represented by the region with intensity z_i and its boundary is Γ_0 . Let the average intensity approximating z_i and z_o be c_1 and c_2 , respectively. Let Γ by the interface separating the regions where the average intensities are c_1 and c_2 . Based on constant average intensities in two different regions, the following energy is introduced:

$$F^{CV}(\Gamma, c_1, c_2) = \mu \cdot (\text{len}(\Gamma)) + \nu \cdot \text{area}(\text{inside}(\Gamma)) \\ + \eta \int_{\text{inside}(\Gamma)} |z - c_1|^2 d\Omega + \gamma \int_{\text{outside}(\Gamma)} |z - c_2|^2 d\Omega, \quad (8)$$

where c_1 and c_2 are unknown constants and $\mu \geq 0$, $\nu \geq 0$, $\eta, \gamma > 0$ are fixed parameters. In Chan and Vese (2001) $\eta = \gamma = 1$, Γ is generally a hypersurface in \mathbb{R}^n , and “ $\text{len}(\Gamma)$ ” is the length in $\mathcal{H}^{n-1}(\Gamma)$. In most of the cases, $\nu = 0$ is taken and only length constraint is imposed. Thus Chan and Vese in (2001) proposed the following energy functional for minimization:

$$\inf_{\Gamma, c_1, c_2} F^{CV}(\Gamma, c_1, c_2). \quad (9)$$

where F^{CV} is given in Eq. (8). This functional is a special case of the piecewise constant Mumford and Shah energy functional (Mumford and Shah 1989).

Level Set Representation of the Model

Consider a Lipschitz function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$, whose zero level set represents the region interface Γ and has opposite signs in different regions (Osher and Sethian 1988). In the level set representation of an unknown curve Γ , transform it from lower low dimension into higher dimension.

So by using level set representation, the Eq. (8) becomes the following:

$$\begin{aligned}
 F^{CV}(\phi, c_1, c_2) = & \mu \int_{\Omega} |\nabla H(\phi)| d\Omega + \nu \int_{\Omega} H(\phi) d\Omega \\
 & + \eta \int_{\Omega} |z - c_1|^2 H(\phi) d\Omega \\
 & + \gamma \int_{\Omega} |z - c_2|^2 (1 - H(\phi)) d\Omega. \tag{10}
 \end{aligned}$$

Once the optimal value ϕ is obtained, the final solution (segmented image) can be found by using the following:

$$u = c_1 H(\phi) + c_2 (1 - H(\phi)).$$

For the existence of minimizers and its relation with the Mumford and Shah model, please see Chan and Vese (2001). It must be noted that c_1, c_2 are the optimal average constant intensities inside and outside curve $\phi = 0$. $H(\phi)$ is the Heaviside function and is used as region descriptor. Due to discontinuity of Heaviside function at origin, a regularized Heaviside function $H_\epsilon(\phi)$ is introduced, and the above functional (10) is minimized with respect to ϕ to the get the following differential equation:

$$\begin{cases}
 \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \eta(z - c_1)^2 + \gamma(z - c_2)^2 \right] = 0 & \text{in } \Omega, \\
 \frac{\delta_\epsilon(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0 & \text{on } \partial \Omega.
 \end{cases} \tag{11}$$

The corresponding unsteady parabolic partial differential equation is considered (Chan and Vese 2001) by introducing an artificial time t .

$$\begin{cases}
 \frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \eta(z - c_1)^2 + \gamma(z - c_2)^2 \right] & \text{in } \Omega, \\
 \phi(t, x, y) = \phi_0(x, y) & \text{in } \Omega \\
 \frac{\partial \phi}{\partial n} = 0 & \text{on } \partial \Omega.
 \end{cases} \tag{12}$$

Note that the steady-state solution of this parabolic partial differential equation will give solution of the corresponding elliptic partial differential equation given in Eq. (11). This is a nonlinear partial differential equation whose solution is done through fast numerical methods which are discussed in the next section.

Fast Numerical Methods:

Solution of nonlinear partial differential equations is a challenging task and is an open problem. In this section, a brief survey on some well-known fast numerical methods for the solution of partial differential equations arising from the minimization of mathematical models for segmentation problems are given. One of the simplest and easy to implement method for this task is explicit method, but this method is stable for small time step, which leads toward a large number of iterations for convergence. Here some well-known stable methods are discussed.

Semi-implicit Method

Consider the following evolution problem which is obtained from minimization of Chan-Vese model:

$$\left\{ \begin{array}{l} c_1(\phi) = \frac{\int_{\Omega} z H_{\epsilon}(\phi) d\Omega}{\int_{\Omega} H_{\epsilon}(\phi) dx dy d\Omega} \quad c_2(\phi) = \frac{\int_{\Omega} z(1 - H_{\epsilon}(\phi)) d\Omega}{\int_{\Omega} (1 - H_{\epsilon}(\phi)) d\Omega} \\ \frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \eta(z - c_1)^2 + \gamma(z - c_2)^2 \right] \quad \text{in } \Omega, \\ \phi(0, x, y) = \phi_0(x, y) \quad \text{in } \Omega, \\ \frac{\partial \phi}{\partial n} = 0 \quad \text{on } \partial \Omega. \end{array} \right. \quad (13)$$

For given initial ϕ , the constant average intensities $c_1(\phi)$ and $c_2(\phi)$ are computed first. And then ϕ is computed by solving the nonlinear PDE given in Eq. (13). Steps of the semi-implicit method for solution of this equation are given here. Suppose that the size of given input image z is $m_1 \times m_2$. Finite difference scheme is used for discretization. Let $x, y \in \Omega$ be the spatial variables, h_1, h_2 be the horizontal and vertical space step size, and Δt be the time step. Divide the image domain into $m_1 \times m_2$ grid points, and let $(x_i, y_j) = (ih_1, jh_2)$, for $i = 1, 2, \dots, m_1$ and $j = 1, 2, \dots, m_2$. Also let $\phi_{i,j}^k = \phi(k\Delta t, x_i, y_j)$ be an approximation of $\phi(t, x, y)$ in the k th iteration, where $k \geq 0$ and $\phi^0 = \phi_0$ be the initial value.

Discretize the parabolic PDE given in Eq. (13) by using finite differences to get the following nonlinear difference equation to be used for updating $\phi^{(k)}$:

$$\begin{aligned} \frac{\phi_{ij}^{k+1} - \phi_{ij}^k}{\Delta t} = & \delta_\epsilon(\phi_{ij}^k) \left[\frac{\mu}{h_1^2} \Delta_x^- \left(\frac{\Delta_x^+ \phi_{ij}^{k+1}}{\sqrt{(\Delta_x^+ \phi_{ij}^k/h_1)^2 + ((\phi_{i,j+1}^k - \phi_{i,j-1}^k)/2h_2)^2 + \beta_1}} \right) \right. \\ & + \frac{\mu}{h_2^2} \Delta_y^- \left(\frac{\Delta_y^+ \phi_{ij}^{k+1}}{\sqrt{((\phi_{i+1,j}^k - \phi_{i-1,j}^k)/2h_1)^2 + (\Delta_y^+ \phi_{ij}^k/h_2)^2 + \beta_1}} \right) \\ & \left. - \nu - \eta(z_{ij} - c_1(\phi^k))^2 + \gamma(z_{ij} - c_2(\phi^k))^2 \right]. \end{aligned}$$

Here $\beta_1 > 0$ is a parameter which avoid singularity. Let $h_1 = h_2 = h = 1$ for simplicity but this is not fixed; different values may be used. Linearizing the above difference equation and denoting the coefficients of $\phi_{i+1,j}^{k+1}, \phi_{i-1,j}^{k+1}, \phi_{i,j+1}^{k+1}, \phi_{i,j-1}^{k+1}$ by A_1, A_2, A_3, A_4 , respectively, lead to the following system of linear equations:

$$\begin{aligned} & \phi_{ij}^{k+1} \left[1 + \mu \delta_\epsilon(\phi_{ij}^k) (A_1 + A_2 + A_3 + A_4) \right] \\ & = \phi_{ij}^k + \Delta t \delta_\epsilon(\phi_{ij}^k) \left[\mu \left(A_1 \phi_{i+1,j}^{k+1} + A_2 \phi_{i-1,j}^{k+1} + A_3 \phi_{i,j+1}^{k+1} \right. \right. \\ & \left. \left. + A_4 \phi_{i,j-1}^{k+1} \right) - \nu - \eta \left(z_{ij} - c_1(\phi^k) \right)^2 + \gamma \left(z_{ij} - c_2(\phi^k) \right)^2 \right]. \end{aligned} \tag{14}$$

If the coefficients A_1, A_2, A_3, A_4 are frozen on the previous iteration, then the above system of nonlinear equations will become a linear system of equation:

$$A\phi^{(k+1)} = f^{(k)},$$

where A is a block tri-diagonal matrix, which can be solved by using any iterative method.

Re-initialization of the level set function is done to prevent the level set function from becoming too flat. This may be done by solving the following initial value problem; see for reference Sussman et al. (1994):

$$\begin{cases} \frac{\partial \xi}{\partial t} = \text{sgn}(\phi(t))(1 - |\nabla \xi|) \\ \xi(0, t) = \phi(t), \end{cases} \tag{15}$$

where ϕ is obtained from solution of Eq. (14) (Chan and Vese 2001).

Algorithm 1 Chan-Vese (CV) algorithm for two-phase image segmentation

$$\phi^{k+1} \rightarrow CV(\phi^k, \mu, tol)$$

1. For given ϕ_0 , calculate average intensities c_1 and c_2 using first two formulas in Eq. (13).
2. Keep c_1 and c_2 fixed, and find numerical solution of the PDE in Eq. (13), to have ϕ^{k+1} .
3. Compute c_1 and c_2 using ϕ^{k+1} .
4. If $|\phi^{k+1} - \phi^k| < tol$ stop else.
5. Re-initialize ϕ , by solving Equation (15), and do step 2.

Note that the semi-implicit method for the solution of parabolic partial differential equations is unconditionally stable (Weickert and Kühne 2002) so will be convergent for large time steps in lower-dimensional problems. As the dimension of the problem increases, the bandwidth of the system matrix becomes much larger and results in a big condition number if the time step is taken larger, whereas the small time step, in that case, would require a large number of iterations, which lead toward slow convergence. This drawback of semi-implicit method was also observed in experimental results; see for details Badshah and Chen (2008) and Weickert et al. (1997).

Additive Operator Splitting (AOS) Method

Operator splitting methods for the solution of PDE have attained much attention from researchers in recent times. Some of these operator splitting methods are additive operator splitting (AOS) (Weickert et al. 1997; Lu et al. 1992), multiplicative operator splitting (MOS) (Barash et al. 2003), and additive+multiplicative operator splitting (AMOS) (Geiser and Bartecki 2017). Only AOS method is discussed here in detail. Weickert et al. (1997) solved the nonlinear diffusion problem by using an additive operator splitting (AOS) method. In this method, a m -dimensional differential operator is converted into m one-dimensional differential operators, and each one-dimensional problem is solved by using the semi-implicit method. The solution in m dimension is the simple algebraic mean of m one-dimensional solutions. Jeon et al. (2005) used AOS method solution of parabolic PDE obtained in minimization of Chan-Vese model for image segmentation. Let us consider the PDE (13):

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \eta(z - c_1)^2 + \gamma(z - c_2)^2 \right]. \quad (16)$$

Corresponding one-dimensional PDE is considered to be discretized. Let k and i represent time and spatial indices, respectively, and $h = 1$ is the spatial step size. Let $\phi_i^k = \phi(i, k)$, and then at the i th grid, the one-dimensional semi-implicit discretization of Eq. (16) is given by the following:

$$\frac{\phi_i^{k+1} - \phi_i^k}{\Delta t} = \delta_\epsilon(\phi_i^k) \left(\frac{\phi_{i+1}^{k+1} - \phi_i^{k+1}}{|\Delta_+^x \phi_i^k|} - \frac{\phi_i^{k+1} - \phi_{i-1}^{k+1}}{|\Delta_+^x \phi_{i-1}^k|} + F_i \right), \quad (17)$$

where $F_i = [-\nu - \eta(z_i - c_1)^2 + \gamma(z_i - c_2)^2]$. Let:

$$A_1 = \frac{1}{|\Delta_+^x \phi_i^k|} \text{ and } A_2 = \frac{1}{|\Delta_+^x \phi_{i-1}^k|},$$

so Equation (17) becomes the following:

$$\phi_i^{k+1} = \phi_i^k + \Delta t \delta_\epsilon(\phi_i^k) (A_1 \phi_{i+1}^{k+1} - (A_1 + A_2) \phi_i^{k+1} + A_2 \phi_{i-1}^{k+1} + F_i). \quad (18)$$

Thus with AOS method, solve problems in x - and y -directions with double time step to get two separate solutions say ϕ_1 and ϕ_2 , and then find the average as follows:

$$\phi = \frac{1}{2}(\phi_1 + \phi_2).$$

Although no stability constraint on the time step is present when the AOS scheme is utilized, the size of the time step cannot be very large because splitting-related artifacts associated with loss of rotational invariance will emerge. The practical implication of this is that the number of iterations needed for the contour to converge remains quite large. For images of large sizes, the methods discussed in this chapter are very slow in convergence. To avoid this problem, multigrid method is the best option.

Multigrid Method

A multigrid method for the Chan-Vese model proposed by Badshah and Chen (2008) is presented here. The proposed method is based on the global formulation of the Chan-Vese model proposed by Chan et al. (2006). Consider Euler-Lagrange's equation deduced from the minimization of Chan-Vese energy functional given in (11):

$$\delta_\epsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \eta(z(x, y) - c_1)^2 + \gamma(z(x, y) - c_2)^2 \right] = 0,$$

$\delta_\epsilon(\phi)$ has non-compact support, so the above equation may be written as follows:

$$\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \eta(z(x, y) - c_1)^2 + \gamma(z(x, y) - c_2)^2 = 0. \quad (19)$$

Equation (19) is Euler-Lagrange's equation of the following functional:

$$\mu \int_{\Omega} |\nabla \phi| d\Omega + \int_{\Omega} (\eta(z(x, y) - c_1)^2 - \gamma(z(x, y) - c_2)^2) \phi(x, y) d\Omega. \quad (20)$$

This is the convex formulation of the Chan-Vese model (Chan and Vese 2001) proposed by Chan et al. in (2006). But the functional given in Equation (20) is

homogenous in ϕ of degree 1 Chan et al. (2006). This means that this functional has no stationary point, so it needs to impose some extra constraints on ϕ that is $0 \leq |\phi| \leq 1$.

Use finite difference scheme to discretize Equation (19) for ϕ . The corresponding discrete equation at a grid point (i, j) is given by the following:

$$\left[\mu \left\{ \frac{\Delta_x^-}{h_1} \left(\frac{\Delta_+^x \phi_{i,j} / h_1}{\sqrt{(\Delta_+^x \phi_{i,j} / h_1)^2 + (\Delta_+^y \phi_{i,j} / h_2)^2 + \beta_1}} \right) + \frac{\Delta_y^-}{h_2} \left(\frac{\Delta_+^y \phi_{i,j} / h_2}{\sqrt{(\Delta_+^x \phi_{i,j} / h_1)^2 + (\Delta_+^y \phi_{i,j} / h_2)^2 + \beta_1}} \right) \right\} - \eta(z_{i,j} - c_1)^2 + \gamma(z_{i,j} - c_2)^2 \right] = 0, \quad (21)$$

where $\beta_1 > 0$ is a small parameter to avoid zero denominator. Equation (21) may be written in the following way:

$$\left[\mu \left\{ \Delta_-^x \left(\frac{\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) + \lambda^2 \Delta_-^y \left(\frac{\Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) \right\} - \eta(z_{i,j} - c_1)^2 + \gamma(z_{i,j} - c_2)^2 = 0, \quad (22) \right.$$

$$\Rightarrow \mu \left\{ \Delta_-^x \left(\frac{\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) + \lambda^2 \Delta_-^y \left(\frac{\Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) \right\} = \eta(z_{i,j} - c_1)^2 - \gamma(z_{i,j} - c_2)^2, \quad (23)$$

where $\underline{\mu} = \mu/h_1$, $\bar{\beta} = h_1^2\beta_1$, and $\lambda = h_1/h_2$, with Neumann’s boundary conditions:

$$\begin{aligned} \phi_{i,0} = \phi_{i,1}, \quad \phi_{i,m_2+1} = \phi_{i,m_2}, \quad \phi_{0,j} = \phi_{1,j}, \quad \phi_{m_1+1,j} = \phi_{m_1,j}, \quad (24) \\ \text{for } i = 1, \dots, m_1, \quad j = 1, \dots, m_2 \text{ and } 0 \leq |\phi_{i,j}| \leq 1. \end{aligned}$$

Note that the left side of Eq. (23) resembles with the denoising model by Rudin et al. (1992) using the total variation (TV) regularization. The parameter $\beta > 0$ should be a small quantity to avoid the singularities.

The Full Approximation Scheme

Multigrid scheme usually known as full approximation scheme (FAS) constitutes three main steps, namely, smoothers, interpolation, and coarse grid solvers; for details see Brandt (1977) and Briggs (1999). Denote the system of nonlinear equations given in Equation (23) and (24) by the following:

$$N^h(\phi^h) = f^h \tag{25}$$

where $h_1 = h_2 = h$, ϕ^h and f^h are grid functions on a $m_1 \times m_2$ cell-centered rectangular grid Ω^h with spacing (h_1, h_2) . Let Ω^{2h} denote the $m_1/2 \times m_2/2$ cell-centered grid which results from standard coarsening of Ω^h . Let $e^h = \phi^h - \Phi^h$ be the solution’s error, where Φ^h is a good approximation to solution of (25) in the sense that e^h is smooth. Such smoothness can only be achieved by a careful choice of suitable smoothers – a major task in developing a working multigrid method.

Let $r^h = f^h - N^h(\Phi^h)$ be the residual. Then the nonlinear residual equation will be as follows:

$$N^h(\Phi^h + e^h) - N^h(\Phi^h) = r^h. \tag{26}$$

Smooth components of error e^h may not be visible on fine grid Ω^h and hence cannot be well approximated. But that can be well approximated on coarse grid Ω^{2h} . Therefore any iterative method which smooths the error on the fine grid can be further well approximated by the use of the coarse grid correction. Note that on coarse grid the residual equation is solved which is less expensive as there will be half the number of grid points. Once a coarse grid approximation of the error is obtained, then it will be transferred back to the fine grid to correct the approximation Φ^h . This is known as a two-grid cycle, and the recursive use of two-grid cycle is termed as a multigrid method. Restriction and interpolation operators for transferring grid functions between Ω^h and Ω^{2h} for cell-centered discretization are defined here:

Restriction

$$I_h^{2h} \Psi^h = \Psi^{2h}$$

where:

$$\Psi_{\ell,m}^{2h} = \frac{1}{4} \left(\Psi_{2\ell-1,2m-1}^h + \Psi_{2\ell-1,2m}^h + \Psi_{2\ell,2m-1}^h + \Psi_{2\ell,2m}^h \right),$$

$$1 \leq \ell \leq m_1/2, \quad 1 \leq m \leq m_2/2.$$

is a full weighting operator (Chen 2005; Trottenberg and Schuller 2001).

Interpolation

$$I_{2h}^h \Psi^{2h} = \Psi^h$$

where:

$$\Psi_{2\ell,2m}^h = \frac{1}{16} \left(9\Psi_{\ell,m}^{2h} + 3\Psi_{\ell+1,m}^{2h} + 3\Psi_{\ell,m+1}^{2h} + \Psi_{\ell+1,m+1}^{2h} \right),$$

$$\Psi_{2\ell-1,2m}^h = \frac{1}{16} \left(9\Psi_{\ell,m}^{2h} + 3\Psi_{\ell-1,m}^{2h} + 3\Psi_{\ell,m+1}^{2h} + \Psi_{\ell-1,m+1}^{2h} \right),$$

$$\Psi_{2\ell,2m-1}^h = \frac{1}{16} \left(9\Psi_{\ell,m}^{2h} + 3\Psi_{\ell+1,m}^{2h} + 3\Psi_{\ell,m-1}^{2h} + \Psi_{\ell+1,m-1}^{2h} \right),$$

$$\Psi_{2\ell-1,2m-1}^h = \frac{1}{16} \left(9\Psi_{\ell,m}^{2h} + 3\Psi_{\ell-1,m}^{2h} + 3\Psi_{\ell,m-1}^{2h} + \Psi_{\ell-1,m-1}^{2h} \right),$$

for $1 \leq \ell \leq m_1/2, \quad 1 \leq m \leq m_2/2.$

is known as a bilinear interpolation operator.

It remains to discuss the most important ingredient of a MG: smoothing. Two types of smoothers, namely, local and global smoothers, are discussed here in detail.

Smoother I: Local Smoother

This smoother is proposed in Badshah and Chen (2008). In this method the system of nonlinear equations is linearized locally, by computing differential coefficients $D(\phi)$ on each grid (i, j) locally to get a system of linear equations. Note that the Gauss-Seidel has the best smoothing property, so apply the Gauss-Seidel method to derive system of linear equations to smooth the error. Using a few steps of this smoother to smooth the error will ensure a convergent nonlinear multigrid. Equation (23) can be written as follows:

$$\begin{aligned} & \underline{\mu} \left\{ \left[\frac{\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} - \frac{\Delta_+^x \phi_{i-1,j}}{\sqrt{(\Delta_+^x \phi_{i-1,j})^2 + (\lambda \Delta_+^y \phi_{i-1,j})^2 + \bar{\beta}}} \right] \right. \\ & \left. + \lambda^2 \left[\frac{\Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} - \frac{\Delta_+^y \phi_{i,j-1}}{\sqrt{(\Delta_+^x \phi_{i,j-1})^2 + (\lambda \Delta_+^y \phi_{i,j-1})^2 + \bar{\beta}}} \right] \right\} \\ & = \eta(z_{i,j} - c_1)^2 - \gamma(z_{i,j} - c_2)^2. \end{aligned}$$

Denoting the differential coefficients in the above equation (intended below to be frozen in local linearization) by $D(\phi)_{i,j}$, $D(\phi)_{i-1,j}$, $D(\phi)_{i,j-1}$ gives the following linear equation:

$$\begin{aligned} & \underline{\mu} \left\{ \left[D(\phi)_{i,j}(\phi_{i+1,j} - \phi_{i,j}) - D(\phi)_{i-1,j}(\phi_{i,j} - \phi_{i-1,j}) \right] \right. \\ & \left. + \lambda^2 \left[D(\phi)_{i,j}(\phi_{i,j+1} - \phi_{i,j}) - D(\phi)_{i,j-1}(\phi_{i,j} - \phi_{i,j-1}) \right] \right\} \quad (27) \\ & = \eta(z_{i,j} - c_1)^2 - \gamma(z_{i,j} - c_2)^2 = f_{i,j}. \end{aligned}$$

Note that all differential coefficients $D(\phi)_{i,j}$, $D(\phi)_{i-1,j}$, and $D(\phi)_{i,j-1}$ contain $\phi_{i,j}$, which will be evaluated at previous time step, and the same values will be used in the rest of the process. Let $\tilde{\varphi}$ be an approximation to ϕ . By putting the values of $\tilde{\varphi}$ at each grid point in Eq. (27) other than the grid point (i, j) and also computing D at each grid point (i, j) , a linear equation in $\phi_{i,j}$ will be obtained:

$$\begin{aligned} & \left\{ \left[D(\tilde{\varphi})_{i,j}(\tilde{\varphi}_{i+1,j} - \phi_{i,j}) - D(\tilde{\varphi})_{i-1,j}(\phi_{i,j} - \tilde{\varphi}_{i-1,j}) \right] \right. \\ & \left. + \lambda^2 \left[D(\tilde{\varphi})_{i,j}(\tilde{\varphi}_{i,j+1} - \phi_{i,j}) - D(\tilde{\varphi})_{i,j-1}(\phi_{i,j} - \tilde{\varphi}_{i,j-1}) \right] \right\} \equiv f_{i,j}/\underline{\mu} \equiv \bar{f}_{i,j}. \end{aligned}$$

Algorithm for solving this equation for $\phi_{i,j}$ to update the approximation at each pixel (i, j) :

Smoother II: Global Smoother

This smoother is proposed in Savage and Chen (2005) for image denoising model and extended to segmentation model in Badshah and Chen (2008). In this method the system of nonlinear equations is linearized globally at each step by computing differential coefficients $D(\phi)$ on each grid point (i, j) . To the resulting system of linear equations, Gauss-Seidel relaxation is applied. Note that the global smoother is different from the local smoother defined above. The algorithm is given as follows:

Algorithm 2 Algorithm for smoother I

$$\phi^h \leftarrow \text{Smoother1}(\phi^h, \bar{f}^h, ITER, tol)$$

where $ITER$ is the maximum number of inner iterations.

for $i = 1 : m_1$

 for $j = 1 : m_2$

 for iter=1:ITER

$\tilde{\varphi}^h \leftarrow \phi^h$

$$\phi_{i,j} = \frac{\left[\left\{ D(\tilde{\varphi}^h)_{i,j} \tilde{\varphi}_{i+1,j}^h + D(\tilde{\varphi}^h)_{i-1,j} \tilde{\varphi}_{i-1,j}^h + \lambda^2 D(\tilde{\varphi}^h)_{i,j} \tilde{\varphi}_{i,j+1}^h + \lambda^2 D(\tilde{\varphi}^h)_{i,j-1} \tilde{\varphi}_{i,j-1}^h \right\} - \bar{f}_{i,j} \right]}{D(\tilde{\varphi}^h)_{i,j} + D(\tilde{\varphi}^h)_{i-1,j} + \lambda^2 (D(\tilde{\varphi}^h)_{i,j} + D(\tilde{\varphi}^h)_{i,j-1})}$$

 if $|\phi_{i,j} - \tilde{\varphi}_{i,j}^h| < tol$ then stop
 end

 end

end

Algorithm 3 Algorithm for smoother II

$$\phi^h \leftarrow \text{Smoother2}(\phi^h, \bar{f}^h, ITER, tol)$$

for $i = 1 : m_1$

 for $j = 1 : m_2$

$$D(\phi^h)_{i,j} = \sqrt{[(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}]}$$

 end

end

$\varphi^h = \phi^h$

for iter = 1 : maxit

 for $i = 1 : n$

 for $j = 1 : m$

$\tilde{\varphi}^h \leftarrow \varphi^h$

$$\varphi_{i,j} = \frac{\left[\left\{ D(\tilde{\varphi}^h)_{i,j} \tilde{\varphi}_{i+1,j}^h + D(\tilde{\varphi}^h)_{i-1,j} \tilde{\varphi}_{i-1,j}^h + \lambda^2 D(\tilde{\varphi}^h)_{i,j} \tilde{\varphi}_{i,j+1}^h + \lambda^2 D(\tilde{\varphi}^h)_{i,j-1} \tilde{\varphi}_{i,j-1}^h \right\} - \bar{f}_{i,j} \right]}{D(\tilde{\varphi}^h)_{i,j} + D(\tilde{\varphi}^h)_{i-1,j} + \lambda^2 (D(\tilde{\varphi}^h)_{i,j} + D(\tilde{\varphi}^h)_{i,j-1})}$$

 end

 end

end

$\phi^h \leftarrow \varphi$

Here updating of the coefficients needs to be done at the beginning of each smoothing step globally and to be stored for relaxation use. This was found to be necessary for the total variation denoising model of Rudin et al. (1992).

The Multigrid Algorithm

The algorithm for solving equation given in Eq. (25) by using the multigrid method is given here. For further details see Chen (2005), Trottenberg and Schuller (2001) and references therein:

Algorithm 4 Multigrid algorithm

Set up the following multigrid parameters:

- it_1 Number of steps required for pre-smoothing on each level
- it_2 Number of steps required for post-smoothing on each level
- $\gamma = 1$ or 2 Selection of V-cycle or W-cycle

rr: Relative residual

For given Φ^h compute \bar{f}^h and keep it fixed. One-step V-cycle of nonlinear multigrid method for CV model is presented here.

FAS

Start

$$\phi^h \leftarrow FASCYC(\phi^h, \bar{f}^h, ITER, it_1, it_2, \gamma, tol)$$

$$\Phi_0 = \Phi^h$$

1. On the coarsest grid, solve Eq. (25) by using SI or AOS methods (Weickert et al. 1997) and then stop.
On finer grids do smoothing, i.e.:

$$\phi^h \leftarrow Smoother^{it_1}(\phi^h, \bar{f}^h, ITER, it_1, it_2, \gamma). \quad (\text{Pre-Smoothing})$$

2. **Restriction:**

$$\phi^{2h} = I_h^{2h} \phi^h, \quad \bar{\phi}^{2h} = \phi^{2h}.$$

$$\bar{f}^{2h} = I_h^{2h}(\bar{f}^h - H^h \phi^h) + N^{2h}(\phi^{2h})$$

$$\phi^{2h} \leftarrow FASCYC_{\gamma}^{2h}(\phi^h, \bar{f}^h, ITER, it_1, it_2, \gamma)$$

3. **Interpolation:**

$$\phi^h \leftarrow \phi^h + I_{2h}^h(\phi^{2h} - \bar{\phi}^{2h})$$

- 4.

$$\phi^h \leftarrow Smoother^{it_2}(\phi^h, \bar{f}^h, ITER, it_1, it_2, \gamma). \quad (\text{Post-Smoothing})$$

Update \bar{f}^h .

If $rr = \frac{\|\phi^h - \phi_0\|_2}{\|\phi_0\|_2} < tol$, stop.

Else go to **Start**.

Local Fourier Analysis of Smoothers

The standard FAS multilevel algorithm (such as Algorithm 4) does not automatically converge for many problems if simple smoothers are used. The key for convergence lies in effective smoothers or reduction of residuals to a smoothed form (Chen 2005; Trottenberg and Schuller 2001). Here local Fourier analysis (LFA) is done to check the effectiveness of the smoothers (say smoother I and smoother II).

Note that LFA cannot be applied to nonlinear smoothers in general. However, for linearized smoothers, the analysis can only be done for each individual smoothing iteration, and the obtained smoothing rates change from iteration to iteration. However, the general trends, e.g., if the three consecutive smoothing rates are 0.58, 0.60, and 0.45 (instead of a constant rate say 0.4), the underlying smoother is effective. Likewise, if the consecutive rates are such that 1.4, 0.99, and 1.09, then the smoother may not be that much effective.

Let us assume that the image domain is a square say $m = m_1 = m_2$. Denote $h = h_1 = h_2$. The typical grid equation on Ω^h is as follows:

$$D(\phi_{i,j})(\phi_{i+1,j} - \phi_{i,j}) - D(\phi_{i-1,j})(\phi_{i,j} - \phi_{i-1,j}) \\ + \lambda^2 [D(\phi_{i,j})(\phi_{i,j+1} - \phi_{i,j}) - D(\phi_{i,j-1})(\phi_{i,j} - \phi_{i,j-1})] = \tilde{f}_{i,j}.$$

For the local smoother, introduce the following notations $g_1 = D(\tilde{\phi})_{i-1,j} = D(\phi^{(k)})_{i-1,j}$, $g_2 = D(\tilde{\phi})_{i,j} = D(\phi^{(k)})_{i,j}$, and $g_3 = D(\tilde{\phi})_{i,j-1} = D(\phi^{(k)})_{i,j-1}$, and similarly for the global smoother, g_1, g_2, g_3 will be computed globally as follows: $g_1 = D(\tilde{\Phi})_{i-1,j}$, $g_2 = D(\tilde{\Phi})_{i,j}$, and $g_3 = D(\tilde{\Phi})_{i,j-1}$ where $\tilde{\Phi}$ is the iterate at the previous sweep (global fixed point). Also as $h_1 = h_2$, so $\lambda^2 = 1$. So:

$$-(g_1 + 2g_2 + g_3)\phi_{i,j}^{k+1} + g_1\phi_{i-1,j}^{k+1} + g_3\phi_{i,j-1}^{k+1} + g_2(\phi_{i,j+1}^k + \phi_{i+1,j}^k) = \tilde{f}_{i,j}.$$

The corresponding error equation will be as follows:

$$-(g_1 + 2g_2 + g_3)e_{i,j}^{k+1} + g_1e_{i-1,j}^{k+1} + g_3e_{i,j-1}^{k+1} + g_2(e_{i,j+1}^k + e_{i+1,j}^k) = 0, \quad (28)$$

where $e_{i,j}^{k+1} = \phi_{i,j} - \phi_{i,j}^{k+1}$ and $e_{i,j}^k = \phi_{i,j} - \phi_{i,j}^k$ are the local error functions after and before the pre(post) smoothing step, respectively.

Recall that the local Fourier analysis (LFA) measures the largest amplification factor in a relaxation scheme (Brandt 1977; Chen 2005; Trottenberg and Schuller 2001). Let the general Fourier component be as follows:

$$B_{\theta_1, \theta_2}(x_i, y_j) = \exp\left(\mathbf{i}\alpha_1 \frac{x_i}{h} + \mathbf{i}\alpha_2 \frac{y_j}{h}\right) = \exp\left(\frac{2\mathbf{i}\theta_1 i \pi}{m} + \frac{2\mathbf{i}\theta_2 j \pi}{m}\right), \quad \mathbf{i} = \sqrt{-1}.$$

Here $\alpha_1 = \frac{2\theta_1 \pi}{m}$, $\alpha_2 = \frac{2\theta_2 \pi}{m} \in [-\pi, \pi]$. The LFA involves expanding the following:

$$e^{k+1} = \sum_{\theta_1, \theta_2 = -m/2}^{m/2} \psi_{\theta_1, \theta_2}^{k+1} B_{\theta_1, \theta_2}(x_i, y_j), \quad e^k = \sum_{\theta_1, \theta_2 = -m/2}^{m/2} \psi_{\theta_1, \theta_2}^k B_{\theta_1, \theta_2}(x_i, y_j)$$

in Fourier components. Now estimate the maximum ratio:

$$\bar{\mu} = \max_{\theta_1, \theta_2} \mu(\theta_1, \theta_2) = |\psi_{\theta_1, \theta_2}^{k+1} / \psi_{\theta_1, \theta_2}^k|$$

in the high-frequency range $(\alpha_1, \alpha_2) \in [-\pi, \pi] \setminus \left[\frac{-\pi}{2}, \frac{\pi}{2} \right]$ which defines the smoothing rate (Trottenberg and Schuller 2001). Now replace all grid functions by their Fourier series and essentially consider the so-called amplification factor, i.e., the ratio between ψ_{θ}^{k+1} and ψ_{θ}^k for each θ where $\theta = (\theta_1, \theta_2)$. Then for the Fourier component of the error functions $e_{i,j}^k$ and $e_{i,j}^{k+1}$ before and after relaxation sweep, consider the following:

$$e_{i,j}^k = \psi_{\theta}^k e^{\mathbf{i}(2\pi\theta_1 i + 2\pi\theta_2 j)/m} \quad \text{and} \quad e_{i,j}^{k+1} = \psi_{\theta}^{k+1} e^{\mathbf{i}(2\pi\theta_1 i + 2\pi\theta_2 j)/m}, \tag{29}$$

putting these values in Equation (28) and defining the following:

$$\mu(\theta) = \left| \frac{\psi_{\theta}^{k+1}}{\psi_{\theta}^k} \right|$$

and introducing $|\theta| = \max(|\theta_1|, |\theta_2|)$; the smoothing factor $\bar{\mu}$ is then obtained as follows:

$$\bar{\mu} = \max_{\hat{\rho}\pi \leq |\theta| \leq \pi} \mu(\theta),$$

where $\hat{\rho}$ is the mesh size ratio and the range $\hat{\rho}\pi \leq |\theta| \leq \pi$ is the suitable range of high- frequency components, i.e., the range of components that cannot be approximated on the coarser grid. For standard coarsening $\hat{\rho} = \frac{1}{2}$, Brandt (1977). The smoothing factor $\bar{\mu}$ is computed for both smoothers. To proceed with an analysis, compute g_1, g_2 and g_3 or the following function:

$$D(\phi) = \sqrt{(\Delta_+^x \phi)^2 + (\Delta_+^y \phi)^2 + \bar{\beta}},$$

Numerically, and work out the smoothing factor $\bar{\mu}$ for each set of coefficients $g_1, g_2,$ and g_3 within a smoother. Use the complete set of coefficients g_1, g_2 and g_3 for computing the smoothing factors $\bar{\mu}$, and display the maximum of such factors:

$$\hat{\mu} = \max_{g_1, g_2, g_3} \bar{\mu} = \max_{g_1, g_2, g_3} \max_{\theta} \mu(\theta).$$

Table 1 $\hat{\mu}$ in the first 4 cycles of out MG algorithm

MG cycle	Smoothing steps	Rate I: $\hat{\mu}^I$	Rate II: $\hat{\mu}^{II}$
1	Pre-smoothing-1	0.4942	0.6776
	Pre-smoothing-2	0.4941	0.9317
	Post-smoothing-1	0.4942	0.9135
	Post-smoothing-2	0.4942	0.9427
2	Pre-smoothing-1	0.6003	0.9561
	Pre-smoothing-2	0.6003	0.9174
	Post-smoothing-1	0.6003	0.9581
	Post-smoothing-2	0.6003	0.9577
3	Pre-smoothing-1	0.7760	0.9533
	Pre-smoothing-2	0.7760	0.9193
	Post-smoothing-1	0.7757	0.9092
	Post-smoothing-2	0.7749	0.9040
4	Pre-smoothing-1	0.6025	0.9594
	Pre-smoothing-2	0.6026	0.9456
	Post-smoothing-1	0.6026	0.9286
	Post-smoothing-2	0.6026	0.9678

As such a linear analysis is based on freezing the nonlinear coefficients, the results should be viewed only as a guide to smoother's effectiveness and a way to distinguish smoothers.

Take a test example of size to 32×32 , and display $\hat{\mu}$ in the first four cycles of the MG algorithm as in Table 1 where Pre-1 refers to the case of "pre-smoothing" and Post-1 to "post-smoothing," etc. By considering the average rate from all pixels, the averages are, respectively, 0.49 and 0.71 for smoothers I and II. Clearly in this example smoother I appears to be more effective than smoother II in terms of rates. For experimental results and comparison, the readers are referred to Badshah and Chen (2008).

In Table 2, the comparison of multigrid, SI, and AOS methods is given. The terms used in the heading of Table 2 have the following meanings:

- Size:** The size of given image $m \times n$.
Itr: Number of iterations used to get the required result.
CPU(s): Time in seconds required for CPU to perform these iterations.
SI: Semi-implicit method.
AOS: Additive operator splitting.
MG: Multigrid method.
ART: Artificial image and **REAL:** Real image like MRI.
****:** Results with high CPU or out of memory.
S-I: Smoother I.
S-II: Smoother II.

Table 2 Comparison of MG with SI and AOS methods

Prob.	Size	AOS method		AOS multi-resolution		SI method		MG method (S-I)		MG method (S-II)	
		Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)
ART	128 ²	60	4.8	60	4.8	80	16.5	2	8.5	2	8
	256 ²	140	50	80	34	100	90.3	2	9.4	3	13.4
	512 ²	280	421	170	277	439	1.3 × 10 ⁴	2	13	3	17
	1024 ²	1200	7661	240	1630	**	**	2	27	3	32
	2048 ²	**	**	**	**	**	**	2	90	3	100
REAL	128 ²	100	10.5	100	10.5	130	32.2	3	12.8	4	15
	256 ²	280	110.5	156	68	180	450	3	14	4	22.2
	512 ²	800	1230	312	503	**	1 × 10 ⁴	3	19.2	4	22.2
	1024 ²	**	**	**	**	**	**	3	40.7	4	42
	2048 ²	**	**	**	**	**	**	3	133	4	136.9

AOS multi-resolution: AOS method is implemented in coarse to fine-level manner, i.e., AOS method is used to solve the problem on the coarsest level and interpolate the solution to the fine level and use it as initial guess, to solve the problem on fine level using AOS method and so on until the finest level is reached.

From Table 2, it can be observed that the MG method is as fast as the SI method and AOS method for images of small sizes, but it is more efficient for images having large sizes, where the abovementioned methods are very slow or not working.

Multigrid Solver for Solving a Class of Variational Problems with Application to Image Segmentation

In section “Multigrid Method”, a multigrid method is discussed in detail for a specific type of image segmentation model, namely, Chan-Vese two-phase model (Chan and Vese 2001). In Roberts et al. (2019), the author proposed a new multigrid method for the following unconstraint model:

$$\min_u \left\{ \mu \int_{\Omega} g(|\nabla z(\mathbf{x})|) |\nabla u| d\Omega + \lambda \int_{\Omega} \mathcal{F}u \, d\Omega + \theta \int_{\Omega} \mathcal{D}u \, d\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(u) d\Omega \right\} \tag{30}$$

where \mathcal{F} is the data fitting term, \mathcal{D} is the distance metric, and v_{ε_2} is the convex-relaxation penalty term which enforces the constraint that $0 \leq u \leq 1$; see Chan et al. (2006) for choice of v_{ε_2} . The corresponding Euler-Lagrange equation is obtained by minimizing the above functional and is given by the following:

$$\mu \nabla \cdot \left(g(|\nabla z(x)|) \frac{\nabla u}{|\nabla u|_{\varepsilon_1}} \right) - \lambda \mathcal{F} - \theta \alpha \mathcal{D} - v'_{\varepsilon_2}(u) = 0 \quad (31)$$

with Neumann boundary conditions and where $\varepsilon_1, \varepsilon_2$ are small positive parameters. Multigrid methods discussed in sections “[Multigrid Method](#)” and “[Multigrid Method for Multiphase Segmentation Model](#)” may not be applied for solution of type of PDE given in (31), due to the following reasons:

1. In the PDE given in (31), the Euler-Lagrange equation arose from minimization of convex formulation of CV model, which has an extra constraint of $0 \leq u \leq 1$, which means that the solution of the PDE will be a binary function everywhere. And hence there will be significant jumps in the values of $v'_{\varepsilon_2}(u)$; this leads to instability of pixel-wise fixed point smoother, and hence the basic multigrid method fails.
2. Small value of ε_2 can lead to the divergence of the algorithm due to discontinuity of the function $v'_{\varepsilon_2}(u)$, whereas large value of ε_2 may guarantee the convergence of the algorithm but change the nature of the problem.
3. ε_1 is the parameter which avoids singularity in the PDE. Most of the multigrid method’s convergence depends on the value of ε_1 ; small value can lead to the nonconvergence of the algorithm, and large value changes the nature of the problem.
4. In the discretization step, all functions will be approximated at the half pixels and due to nonsmoothness of the edge function, its approximation at the half pixel may be very inaccurate.
5. Divergence term in the PDE (31) is highly nonlinear. Approximation of this term around the interfaces in g and u may be inaccurate due to the use of singularity parameter ε_1 as discussed above.

To address these bullets and to apply multigrid methods, the authors in Roberts et al. (2019) introduced a new formulation of the models given in (30).

First Algorithm

Model in (30) is reformulated by removing the penalty term $v_{\varepsilon_2}(u)$ which is done by introducing a new variable v . The new reformulated model becomes the following:

$$\min_{u,v} \left\{ \mu \int_{\Omega} g(|\nabla z(x)|) |\nabla u| d\Omega + \lambda \int_{\Omega} \mathcal{F} v d\Omega + \theta \int_{\Omega} \mathcal{D} v d\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(v) d\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 \right\}, \quad (32)$$

where θ_B is a tuning parameter. This model will be minimized with respect to u and v . To minimize with respect to u , the above model reduces to the following:

$$\min_u \left\{ \mu \int_{\Omega} g(|\nabla z(\mathbf{x})|) |\nabla u| d\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 \right\}. \tag{33}$$

Minimization with respect to u leads to the following PDE:

$$\mu \nabla \cdot \left(g(|\nabla z|) \frac{\nabla u}{|\nabla u|_{\varepsilon_1}} \right) = 0 \tag{34}$$

With Neumann boundary condition. In the minimization problem for v , the following minimization problem is considered:

$$\min_v \left\{ \lambda \int_{\Omega} \mathcal{F}v \, d\Omega + \theta \int_{\Omega} \mathcal{D}v \, d\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(v) d\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 \right\}, \tag{35}$$

whose solution is as follows:

$$v^{(k+1)} = v = \min \left\{ \max \left\{ u - \frac{\lambda \mathcal{F} + \theta \mathcal{D}}{\theta_B}, 0 \right\}, 1 \right\}. \tag{36}$$

It can be noted that both PDEs do not contain v'_{ε_2} , which is the one of the achievement of the proposed algorithm. For detailed steps of the algorithm, see Roberts et al. (2019).

Furthermore, the authors introduced Split-Bregman iterations for removing nonlinearity in the weighted TV term. This is done by introducing a new variable \mathbf{d} for the weighted TV, and hence the minimization problem given in (30) becomes the following:

$$\min_{u, \mathbf{d}} \left\{ \mu \int_{\Omega} |\mathbf{d}|_g d\Omega + \lambda \int_{\Omega} \mathcal{F}u \, d\Omega + \theta \int_{\Omega} \mathcal{D}u \, d\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(u) d\Omega + \frac{\lambda_B}{2} \|\mathbf{d} - \nabla u - b\|_{L^2}^2 \right\}, \tag{37}$$

where $|\mathbf{d}|_g = g(|\nabla z|) |\nabla u|$ and $\lambda_B \geq 0$. Note that b is the Bregman update which has a simple update formula. To find optimal value of u , the following minimization problem will be solved:

$$\min_u \left\{ \lambda \int_{\Omega} \mathcal{F}u \, d\Omega + \theta \int_{\Omega} \mathcal{D}u \, d\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(u) d\Omega + \frac{\lambda_B}{2} \|\mathbf{d} - \nabla u - b\|_{L^2}^2 \right\}. \tag{38}$$

Minimization problem for \mathbf{d} takes the following form:

$$\min_{\mathbf{d}} \left\{ \mu \int_{\Omega} |\mathbf{d}|_g d\Omega + \frac{\lambda_B}{2} \|\mathbf{d} - \nabla u - b\|_{L^2}^2 \right\}, \quad (39)$$

whose closed form solution is given as follows:

$$\mathbf{d} = \text{shrink} \left(\nabla u + b \frac{\mu g(|\nabla z|)}{\lambda_B} \right), \quad (40)$$

where $\text{shrink}(a, b) = \text{sgn}(a) \max\{|a| - b, 0\}$. b given in Equation (37) can be updated as follows:

$$b^{(k+1)} = b^{(k)} + \nabla u^{(k+1)} - \mathbf{d}^{(k+1)}. \quad (41)$$

In the Bregman iterations, Equation (38) is remaining to be solved which is still nonlinear and is not amenable to fast multigrid method. To solve this problem, the authors reformulate the minimization problem (33) to the following:

$$\min_{u, \mathbf{d}} \left\{ \mu \int_{\Omega} |\mathbf{d}|_g d\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 + \frac{\lambda_B}{2} \|\mathbf{d} - \nabla u - b\|_{L^2}^2 \right\} \quad (42)$$

using Bregman splitting where θ_B and λ_B are fixed nonnegative parameters. The following subproblem is considered for u :

$$u^{(k+1)} = \arg \min_u \left\{ \frac{\theta_B}{2} \|u - v^{(k)}\|_{L^2}^2 + \frac{\lambda_B}{2} \|\mathbf{d}^{(k)} - \nabla u - b^{(k)}\|_{L^2}^2 \right\} \quad (43)$$

and the minimizer is the solution of the following:

$$-\lambda_B \Delta u + \theta_B u = \theta_B v^{(k)} - \lambda_B \nabla \cdot (\mathbf{d}^{(k)} - b^{(k)}) \quad (44)$$

with Neumann boundary conditions. This is a linear PDE which can be solved by a multigrid method. \mathbf{d} , b , and v will be updated as given in (40), (41), and (36), respectively. PDEs obtained from minimization of various subproblems are solved by using additive operator splitting method and multigrid methods; for detail see Sect. 5 in Roberts et al. (2019).

Sobolev Gradient Minimization of Curve Length in Chan-Vese Model

In Yuan and He (2012), the Sobolev gradient is used to minimize the length term in the Chan-Vese segmentation model. Denote the length term in Chan-Vese model by $\mathcal{L}(\phi) = \int_{\Omega} \delta_{\epsilon}(\phi) |\nabla \phi| d\Omega$. The Sobolev gradient of the curve length functional $L(\phi)$ may be represented through L^2 gradient. As done earlier, the Gâteaux derivative of $\mathcal{L}(\phi)$ in the direction of a test function $h \in C_0^{\infty}$ is given by the following:

$$\mathfrak{L}'(\phi)h = \lim_{\epsilon \rightarrow 0} \frac{\mathfrak{L}(\phi + \epsilon h) - \mathfrak{L}(\phi)}{\epsilon} = \left\langle \delta(\phi) \frac{\nabla \phi}{|\nabla \phi|}, \nabla h \right\rangle_{L^2(\Omega)^2} + \int_{\Omega} \delta'(\phi) |\nabla \phi| h d\Omega. \tag{45}$$

The inner product can be simplified by using integration by parts, which will happen if ϕ belongs to Sobolev space $H^{2,2}(\Omega)$. The Gâteaux derivative of length term $\mathfrak{L}'(\phi)h$ is defined to be the unique element that represents the bounded linear functional $\mathfrak{L}'(\phi)$ in $L^2(\Omega)$ as follows:

$$\mathfrak{L}'(\phi)h = \langle \nabla \mathfrak{L}(\phi), h \rangle_{L^2(\Omega)} \tag{46}$$

where $\nabla \mathfrak{L}(\phi)$ is the gradient of $\mathfrak{L}(\phi)$ in L^2 space. Integration by parts is applied on Eq. (45) to get the following:

$$\nabla \mathfrak{L}(\phi) = -\delta_{\epsilon}(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \tag{47}$$

with Neumann boundary conditions.

To find the Sobolev gradient of $L(\phi)$, integration by parts will not be used to integrate the inner product term in Eq. (45). Define the following:

$$D\phi = \begin{pmatrix} \phi \\ \nabla \phi \end{pmatrix}$$

where $\phi \in H^{1,2}(\Omega)$. In $\phi \in H^{1,2}(\Omega)$, the inner product may be defined as follows:

$$\langle \phi, h \rangle_{H^{1,2}(\Omega)} = \int_{\Omega} \phi h + \langle \nabla \phi, \nabla h \rangle_{H^{1,2}(\Omega)^2} = \langle D\phi, Dh \rangle_{L^2(\Omega)^3}, \quad h \in H^{1,2}(\Omega).$$

For any function $\phi, h \in H^{1,2}(\Omega)$, it is well known that the Gâteaux derivative $\mathfrak{L}'(\phi)$ which is given in Eq. (45) exists and is a bounded linear functional on $H^{1,2}(\Omega)$. By the Riesz theorem, the Gâteaux derivative $\mathfrak{L}'(\phi)h$ is defined to be the unique element $R(\phi)$ that represents the bounded linear functional $\mathfrak{L}'(\phi)$ on $H^{1,2}(\Omega)$ as follows:

$$\mathfrak{L}'(\phi)h = \langle R(\phi), h \rangle_{H^{1,2}(\Omega)}. \tag{48}$$

Here, $R(\phi)$ is the Sobolev gradient which is denoted by $\nabla_s \mathfrak{L}(\phi)$. For $\phi \in H^{2,2}(\Omega)$, using integration by parts on Eq. (45), $\mathfrak{L}'(\phi)$ can be represented by L^2 gradient as follows:

$$\mathfrak{L}'(\phi)h = \langle R(\phi), h \rangle_{H^{1,2}(\Omega)} = \langle D(\nabla_s \mathfrak{L}(\phi)), Dh \rangle_{L^2(\Omega)^3} = \langle D^* D(\nabla_s \mathfrak{L}(\phi)), h \rangle_{L^2(\Omega)} \tag{49}$$

where $D^* = (I, -\nabla)$ is the adjoint of D . The two gradients may be related in the following way:

$$D^* D(\nabla_s \mathbf{L}(\phi)) = \nabla \mathbf{L}(\phi) \text{ or } \nabla_s \mathbf{L}(\phi) = (D^* D)^{-1} \nabla \mathbf{L}(\phi) \quad (50)$$

it can be noted that

$$D^* D = (I, -\nabla) \begin{pmatrix} I \\ \nabla \end{pmatrix} = I - \Delta.$$

Combine all these results to get the Sobolev gradient of the length term, which is given as follows:

$$\nabla \mathbf{L}(\phi) = -(I - \Delta)^{-1} \left(\delta_\epsilon(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \right). \quad (51)$$

The data fitting term of the Chan-Vese model:

$$E(\phi) = \eta \int_{\Omega} |z - c_1|^2 H(\phi) d\Omega + \gamma \int_{\Omega} |z - c_2|^2 (1 - H(\phi)) d\Omega$$

is minimized by using L^2 gradient $\nabla E(\phi)$. Thus the combined evolution equation is given by the following:

$$\begin{cases} \frac{\partial \phi}{\partial t} = (I - \Delta)^{-1} \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \eta(z - c_1)^2 + \gamma(z - c_2)^2 \right] & \text{in } \Omega, \\ \phi(t, x, y) = \phi_0(x, y) & \text{in } \Omega \\ \frac{\partial \phi}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (52)$$

Numerical Method

The evolution equation given in Eq. (52) is solved in the following way: the Sobolev gradient term is computed by introducing an intermediate variable say Φ , i.e.:

$$\Phi = (I - \Delta)^{-1} \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] \quad (53)$$

or:

$$(I - \Delta)\Phi = \delta_\epsilon(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right]. \quad (54)$$

Table 3 Speed comparison of L^2 and Sobolev gradients

Prob.	Prob 1		Prob 2		Prob 3		Prob 4		Prob 5	
L^2 gradient	Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)	Itr	CPU(s)
	400	313	60	13	50	9	700	632	100	70
L^2 +Sobolev grads	40	8	28	7	17	4	88	64	60	12

For given value of $\phi_{i,j}^{(k)}$, the above equation will be solved by using fast Poisson solver to get $\Phi(\phi_{i,j}^{(k)}, \phi_{i,j}^{(k+1)})$. To find numerical solution of evolution equation given in Eq. (52), the following procedure will be followed. Starting with the initial value of ϕ , compute c_1 and c_2 . Then the numerical approximation of the Eq. (52) can be found by solving the following discrete equation:

$$\frac{\phi_{i,j}^{(k+1)} - \phi_{i,j}^{(k)}}{\Delta t} = \mu\Phi(\phi_{i,j}^{(k)}, \phi_{i,j}^{(k+1)}) + \delta(\phi_{i,j}^{(k)}) \left[-\lambda_1(z_{i,j} - c_1)^2 + \lambda_2(z_{i,j} - c_2)^2 \right]. \tag{55}$$

For more details and algorithm, please see Yuan and He (2012).

Speed comparison of both type gradients, i.e., L^2 and L^2 combined with Sobolev gradients in Table 3. Both methods are tested on five different type of problems, and their number of iterations and CPU time in seconds is recorded. It is seen from the table that L^2 combined with Sobolev gradients showed good results compared to L^2 gradient only.

Multiphase Image Segmentation

Multigrid Method for Multiphase Segmentation Model

In the previous section, Chan-Vese model was discussed which divides a gray image into two phases say foreground and background. Another model proposed by Vese and Chan (2002), which divides an image into four phases, will be discussed here. By using one level set function, an image will be divided into two phases, whereas increasing the number of level set functions will increase the number of phases. To segment an image into n phases, $\log_2 n$ number of level set functions will be required.

Consider $p = \log_2 n$ level set function $\phi_\ell : \Omega \rightarrow \mathbb{R}$ for $\ell = 1, 2, \dots, p$. The union of the zero level sets of all ϕ_ℓ will determine the set of edges in the segmented image. For $1 \leq s \leq n = 2^p$, denote by $c_s = \text{mean}(z)$ the average value of image gray scales in phase s and by χ_s the characteristic function for phase s . The following energy functional is proposed; see for detail Vese and Chan (2002):

$$\begin{aligned}
F_n(c, \Phi) &= \sum_{1 \leq s \leq n} \int_{\Omega} (z(x, y) - c_s)^2 \chi_s dx dy \\
&\quad + \mu \sum_{1 \leq \ell \leq p} \int_{\Omega} |\nabla H(\phi_{\ell})| dx dy
\end{aligned} \tag{56}$$

where $c = (c_1, c_2, \dots, c_n)$ and $\Phi = (\phi_1, \phi_2, \dots, \phi_p)$; note $n = 2^p$. In this section, main focus will be on the four-phase segmentation, i.e., $n = 4$ or $p = 2$.

Consider the following minimization problem for four-phase segmentation:

$$\min_{c, \Phi} F_4(c, \Phi), \tag{57}$$

where:

$$\begin{aligned}
F_4(c, \Phi) &= \int_{\Omega} (z(x, y) - c_{11})^2 H(\phi_1) H(\phi_2) dx dy \\
&\quad + \int_{\Omega} (z(x, y) - c_{10})^2 H(\phi_1) (1 - H(\phi_2)) dx dy \\
&\quad + \int_{\Omega} (z(x, y) - c_{01})^2 (1 - H(\phi_1)) H(\phi_2) dx dy \\
&\quad + \mu \int_{\Omega} |\nabla H(\phi_1)| dx dy \\
&\quad + \int_{\Omega} (z(x, y) - c_{00})^2 (1 - H(\phi_1)) (1 - H(\phi_2)) dx dy \\
&\quad + \mu \int_{\Omega} |\nabla H(\phi_2)| dx dy
\end{aligned} \tag{58}$$

where $c = (c_{11}, c_{10}, c_{01}, c_{00})$ is the vector of average intensities in different phases of the given image and $\Phi = (\phi_1, \phi_2)$ is the vector of level sets used for segmentation of an image into various phases. Minimization of (57) with respect to Φ leads to the following system of equations:

$$\begin{cases} \delta_{\epsilon}(\phi_1) \left[\mu \nabla \cdot \frac{\nabla \phi_1}{|\nabla \phi_1|} - [T_1 H_{\epsilon}(\phi_2) + T_2 (1 - H_{\epsilon}(\phi_2))] \right] = 0, \\ \delta_{\epsilon}(\phi_2) \left[\mu \nabla \cdot \frac{\nabla \phi_2}{|\nabla \phi_2|} - [T_1 H_{\epsilon}(\phi_1) + T_2 (1 - H_{\epsilon}(\phi_1))] \right] = 0, \end{cases} \tag{59}$$

with Neumann boundary conditions, where $T_1 = (z - c_{11})^2 - (z - c_{01})^2$ and $T_2 = (z - c_{10})^2 - (z - c_{00})^2$. This system of coupled partial differential equations is usually solved by introducing artificial time variable and using well-known time marching schemes like semi-implicit and additive operator splitting methods which

are discussed in previous section, which are effective in problems with small sizes. For large-size problems, the best option is the multigrid method.

Multigrid Method with Typical and Modified Smoother

Using finite difference schemes to discretize (59) for ϕ_ℓ , the equations at a pixel point (i, j) are given by the following:

$$\left\{ \begin{aligned} \delta_\epsilon(\phi_1)_{i,j} & \left\{ \frac{\Delta_x^-}{h_1} \frac{\mu \Delta_+^x(\phi_1)_{i,j}/h_1}{\sqrt{(\Delta_+^x(\phi_1)_{i,j}/h_1)^2 + (\Delta_+^y(\phi_1)_{i,j}/h_2)^2 + \beta}} - (T_1)_{i,j} H_\epsilon(\phi_2)_{i,j} + \right. \\ & \left. \frac{\Delta_y^-}{h_2} \frac{\mu \Delta_+^y(\phi_1)_{i,j}/h_2}{\sqrt{(\Delta_+^x(\phi_1)_{i,j}/h_1)^2 + (\Delta_+^y(\phi_1)_{i,j}/h_2)^2 + \beta}} - (T_2)_{i,j}(1 - H_\epsilon(\phi_2)_{i,j}) \right\} = 0, \\ \delta_\epsilon(\phi_2)_{i,j} & \left\{ \frac{\Delta_x^-}{h_1} \frac{\mu \Delta_+^x(\phi_2)_{i,j}/h_1}{\sqrt{(\Delta_+^x(\phi_2)_{i,j}/h_1)^2 + (\Delta_+^y(\phi_2)_{i,j}/h_2)^2 + \beta}} - (T_1)_{i,j} H_\epsilon(\phi_1)_{i,j} + \right. \\ & \left. \frac{\Delta_y^-}{h_2} \frac{\mu \Delta_+^y(\phi_2)_{i,j}/h_2}{\sqrt{(\Delta_+^x(\phi_2)_{i,j}/h_1)^2 + (\Delta_+^y(\phi_2)_{i,j}/h_2)^2 + \beta}} - (T_2)_{i,j}(1 - H_\epsilon(\phi_1)_{i,j}) \right\} = 0, \end{aligned} \right. \tag{60}$$

Let $\mu = \mu/h_1$, $\bar{\beta} = h_1^2\beta$ and $\lambda = h_1/h_2$. Also denote $(f_1)_{i,j} = (T_1)_{i,j} H_\epsilon(\phi_2)_{i,j} + (T_2)_{i,j}(1 - H_\epsilon(\phi_2)_{i,j})$ and $(f_2)_{i,j} = (T_1)_{i,j} H_\epsilon(\phi_1)_{i,j} + (T_2)_{i,j}(1 - H_\epsilon(\phi_1)_{i,j})$.

For $\ell = 1, 2$, introducing the following notation for the differential coefficients as follows:

$$\begin{aligned} D_\ell(\phi_\ell)_{i,j} &= \frac{1}{\sqrt{(\Delta_+^x(\phi_\ell)_{i,j})^2 + (\lambda \Delta_+^y(\phi_\ell)_{i,j})^2 + \bar{\beta}}}, \\ D_\ell(\phi_\ell)_{i-1,j} &= \frac{1}{\sqrt{(\Delta_+^x(\phi_\ell)_{i-1,j})^2 + (\lambda \Delta_+^y(\phi_\ell)_{i-1,j})^2 + \bar{\beta}}}, \\ D_\ell(\phi_\ell)_{i,j-1} &= \frac{1}{\sqrt{(\Delta_+^x(\phi_\ell)_{i,j-1})^2 + (\lambda \Delta_+^y(\phi_\ell)_{i,j-1})^2 + \bar{\beta}}}. \end{aligned}$$

Thus locally linearized form of Equation (60) is given by the following:

$$\begin{aligned} & \left[D_\ell(\phi_\ell)_{i,j}((\phi_\ell)_{i+1,j} - (\phi_\ell)_{i,j}) - D_\ell(\phi_\ell)_{i-1,j}((\phi_\ell)_{i,j} - (\phi_\ell)_{i-1,j}) \right] \\ & + \lambda^2 \left[D_\ell(\phi_\ell)_{i,j}((\phi_\ell)_{i,j+1} - (\phi_\ell)_{i,j}) - D_\ell(\phi_\ell)_{i,j-1}((\phi_\ell)_{i,j} - (\phi_\ell)_{i,j-1}) \right] \\ & = (\tilde{f}_\ell)_{i,j}, \end{aligned} \tag{61}$$

where $\tilde{f}_\ell = f_\ell/\mu$.

Let $\tilde{\phi}_\ell$ be the approximation to ϕ_ℓ at the current iteration. Then from Equation (61), pursuing only local unknowns ϕ_ℓ at (i, j) in the following linear equations:

$$\begin{aligned} & \left[D_\ell(\tilde{\phi}_\ell)_{i,j}((\tilde{\phi}_\ell)_{i+1,j} - (\phi_\ell)_{i,j}) - D_\ell(\tilde{\phi}_\ell)_{i-1,j}((\phi_\ell)_{i,j} - (\tilde{\phi}_\ell)_{i-1,j}) \right] \\ & + \lambda^2 \left[D_\ell(\tilde{\phi}_\ell)_{i,j}((\tilde{\phi}_\ell)_{i,j+1} - (\phi_\ell)_{i,j}) - D_\ell(\tilde{\phi}_\ell)_{i,j-1}((\phi_\ell)_{i,j} - (\tilde{\phi}_\ell)_{i,j-1}) \right] \\ & = (\bar{f}_\ell)_{i,j}. \end{aligned} \quad (62)$$

These equations will be solved for $(\phi_\ell)_{i,j}$, and store their values in $(\tilde{\phi}_\ell)_{i,j}$, to use it in the next iteration. This equation is used as a smoother in the multigrid Algorithm 4. For further details, see Badshah and Chen (2009). Local Fourier analysis is usually used to check the convergence of the smoother, and this is discussed in the next section.

Local Fourier Analysis and a Modified Smoother

Local Fourier analysis (LFA) is a suitable tool to analyze the convergence rate of any iterative method for linear equations. However, the underlying equations are nonlinear, so LFA will consider a linearized equation, and as linearization occurs locally at each pixel, the maximum rate from all pixel locations will be considered.

Consider a square image with $m = m_1 = m_2$ and $h_1 = h_2 = h$ for simplicity, then $\lambda = 1$. Given the previous iterate at step k , $\tilde{\phi}_\ell = \phi_\ell^{(k)}$, denote $a_1 = D_1(\tilde{\phi}_1)_{i-1,j}$, $a_2 = D_1(\tilde{\phi}_1)_{i,j}$, $a_3 = D_1(\tilde{\phi}_1)_{i,j-1}$, $b_1 = D_2(\tilde{\phi}_2)_{i-1,j}$, $b_2 = D_2(\tilde{\phi}_2)_{i,j}$, $b_3 = D_2(\tilde{\phi}_2)_{i,j-1}$ which are to be considered as local constants. From (61), the grid equation at (i, j) is the following local smoother:

$$\begin{cases} - (a_1 + 2a_2 + a_3)(\phi_1)_{i,j}^{(k+1)} + a_1(\phi_1)_{i-1,j}^{(k+1)} + a_3(\phi_1)_{i,j-1}^{(k+1)} \\ + a_2[(\phi_1)_{i+1,j}^{(k)} + (\phi_1)_{i,j+1}^{(k)}] = (\bar{f}_1)_{i,j}, - (b_1 + 2b_2 + b_3)(\phi_2)_{i,j}^{(k+1)} \\ + b_1(\phi_2)_{i-1,j}^{(k+1)} + b_3(\phi_2)_{i,j-1}^{(k+1)} + b_2[(\phi_2)_{i+1,j}^{(k)} + (\phi_2)_{i,j+1}^{(k)}] = (\bar{f}_2)_{i,j}. \end{cases} \quad (63)$$

Define the error functions by $e_1^{(k)} = \phi_1 - \phi_1^{(k)}$ and $e_2^{(k)} = \phi_2 - \phi_2^{(k)}$. Then using (127) and (63) with frozen $(\bar{f}_1)_{i,j}$ and $(\bar{f}_2)_{i,j}$, the error equations are as follows:

$$\begin{cases} a_1(e_1)_{i-1,j}^{(k+1)} + a_3(e_1)_{i,j-1}^{(k+1)} + a_2[(e_1)_{i+1,j}^{(k)} + (e_1)_{i,j+1}^{(k)}] \\ -(a_1 + 2a_2 + a_3)(e_1)_{i,j}^{(k+1)} = 0b_1(e_2)_{i-1,j}^{(k+1)} \\ +b_3(e_2)_{i,j-1}^{(k+1)} + b_2[(e_2)_{i+1,j}^{(k)} + (e_2)_{i,j+1}^{(k)}] - (b_1 + 2b_2 + b_3)(e_2)_{i,j}^{(k+1)} = 0. \end{cases} \tag{64}$$

Recall that the LFA measures the largest amplification factor in a relaxation scheme (Brandt 1977; Chen 2005; Trottenberg and Schuller 2001). Let a general Fourier component be the following:

$$\Theta_{\alpha,\beta}(x_i, y_j) = \exp\left(\mathbf{i}\theta_\alpha \frac{x_i}{h} + \mathbf{i}\theta_\beta \frac{y_j}{h}\right) = \exp\left(\frac{2\mathbf{i}\alpha i \pi}{m} + \frac{2\mathbf{i}\beta j \pi}{m}\right).$$

Note that $\theta_\alpha, \theta_\beta \in [-\pi, \pi]$. The LFA expands:

$$e_1^{(k)} = \sum_{\alpha,\beta=-m/2}^{m/2} (\psi_1^{(k)})_{\alpha,\beta} \Theta_{\alpha,\beta}(x_i, y_j), \quad e_2^{(k)} = \sum_{\alpha,\beta=-m/2}^{m/2} (\psi_2^{(k)})_{\alpha,\beta} \Theta_{\alpha,\beta}(x_i, y_j)$$

in Fourier components. Taking the largest spectral radius (maximum eigenvalue) of the amplification matrix $\mathcal{A}_{\alpha,\beta}$ (Trottenberg and Schuller 2001):

$$\begin{bmatrix} (\psi_1^{(k+1)})_{\alpha,\beta} \\ (\psi_2^{(k+1)})_{\alpha,\beta} \end{bmatrix} = \mathcal{A}_{\alpha,\beta} \begin{bmatrix} (\psi_1^{(k)})_{\alpha,\beta} \\ (\psi_2^{(k)})_{\alpha,\beta} \end{bmatrix}.$$

After substituting these components into (64) for $e_1^{(k+1)}, e_1^{(k)}$ and $e_2^{(k+1)}, e_2^{(k)}$:

$$\mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \frac{a_2\left(e^{\frac{2\mathbf{i}\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right)}{\left(a_1 + 2a_2 + a_3 - a_1 e^{-\frac{2\mathbf{i}\alpha\pi}{m}} - a_3 e^{-\frac{2\mathbf{i}\beta\pi}{m}}\right)} & 0 \\ 0 & \frac{b_2\left(e^{\frac{2\mathbf{i}\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right)}{\left(b_1 + 2b_2 + b_3 - b_1 e^{-\frac{2\mathbf{i}\alpha\pi}{m}} - b_3 e^{-\frac{2\mathbf{i}\beta\pi}{m}}\right)} \end{bmatrix}.$$

At the k th iteration, each rate $\bar{\mu}^{(k)}(i, j) = \max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$ in the high-frequency range $(\theta_\alpha, \theta_\beta) \in [-\pi, \pi] \setminus [-\frac{\pi}{2}, \frac{\pi}{2}]$, measuring the effectiveness of a smoother (Brandt 1977), is dependent on $a_\ell, b_\ell, \ell = 1, 2, 3$, which in turn depends on the pixel location (I, j) . Therefore looking for the largest smoothing rate for all i, j (i.e., among all such pixels):

$$\hat{\mu} = \max_{a_1, a_2, a_3, b_1, b_2, b_3} \bar{\mu}^{(k)}(i, j).$$

Table 4 The smoothing rate for a local smoother with 3 inner iterations

Outer iterations s	The smoothing rate $\hat{\mu}_s$	The smoothing rate taking out “odd pixels” $\hat{\mu}_s^*$
1	0.6862	0.5720
2	0.6861	0.3170
3	0.6861	0.2747

However, due to the high nonlinearity, it is useful to define the smoothing rate as the maximum of the above-accumulated rates out of all s relaxation steps by the following:

$$\hat{\mu}_s = \max_{i,j} \bar{\mu}^{(1)}(i,j) \bar{\mu}^{(2)}(i,j) \cdots \bar{\mu}^{(s)}(i,j).$$

Clearly for linear equations, where a_ℓ, b_ℓ are constants, $\bar{\mu} = \bar{\mu}^{(k)}$ is a constant so $\hat{\mu}_s = \bar{\mu}^{(s)}$. Here, as a_ℓ, b_ℓ are not constants, with this particular definition, and allowing the possibility of $\bar{\mu}^{(k)}(i,j) \approx 1$ for some i, j, k . As long as $\hat{\mu}_s \ll 1$, then a smoother will be effective. In Table 4, smoothing rates for an artificial image of size $m = 32$ are given; note that similar results are obtained with $m = 64$. Here, in Table 4, the “odd pixels” refer to positions where the relative ratios between a_2 and $\max(a_1, a_3)$, or the ratios between b_2 and $\max(b_1, b_3)$, are quite large. Clearly our smoother is ineffective overall due to these odd pixels. This prompted to consider how to improve the overall smoothing rate (column 2 in Table 4).

A modified smoother. To motivate the idea, consider the particular case of an odd pixel assigned with the following:

$$a_1 = 0.3536, a_2 = 10,000, a_3 = 0.3536, b_1 = 0.3536, b_2 = 10,000, b_3 = 0.3536 \quad (65)$$

for which LFA as described above gives a local (large) rate of $\mu = 0.99996$. An alternative to (63), the following under-relaxation smoothing scheme at these odd pixels:

$$\left\{ \begin{array}{l} a_1(\phi_1)_{i-1,j}^{(k+1)} + a_3(\phi_1)_{i,j-1}^{(k+1)} + a_2 \left[(\phi_1)_{i+1,j}^{(k)} + (\phi_1)_{i,j+1}^{(k)} \right] \\ - (a_1 + 2a_2 + a_3)(1 + \omega)(\phi_1)_{i,j}^{(k+1)} + \omega(a_1 + 2a_2 + a_3)(\phi_1)_{i,j}^{(k)} = (\bar{f}_1)_{i,j}, \\ b_1(\phi_2)_{i-1,j}^{(k+1)} + b_3(\phi_2)_{i,j-1}^{(k+1)} + b_2 \left[(\phi_2)_{i+1,j}^{(k)} + (\phi_2)_{i,j+1}^{(k)} \right] \\ - (b_1 + 2b_2 + b_3)(1 + \omega)(\phi_2)_{i,j}^{(k+1)} + \omega(b_1 + 2b_2 + b_3)(\phi_2)_{i,j}^{(k)} = (\bar{f}_2)_{i,j}, \end{array} \right. \quad (66)$$

for some $0 \leq \omega \leq 1$ (note $\omega = 0$ reduces to the previous local smoother). The new error equation is as follows:

$$\left\{ \begin{array}{l} a_1(e_1)_{i-1,j}^{(k+1)} + a_3(e_1)_{i,j-1}^{(k+1)} + a_2 \left[(e_1)_{i+1,j}^{(k)} + (e_1)_{i,j+1}^{(k)} \right] \\ \quad - (a_1 + 2a_2 + a_3)(1 + \omega)(e_1)_{i,j}^{(k+1)} + \omega(a_1 + 2a_2 + a_3)(e_1)_{i,j}^{(k)} = 0, \\ b_1(e_2)_{i-1,j}^{(k+1)} + b_3(e_2)_{i,j-1}^{(k+1)} + b_2 \left[(e_2)_{i+1,j}^{(k)} + (e_2)_{i,j+1}^{(k)} \right] \\ \quad - (1 + \omega)(b_1 + 2b_2 + b_3)(e_2)_{i,j}^{(k+1)} + \omega(b_1 + 2b_2 + b_3)(e_2)_{i,j}^{(k)} = 0. \end{array} \right. \quad (67)$$

Then the corresponding new Fourier amplification matrix is as follows:

$$\mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \frac{a_2 \left(e^{\frac{2\alpha\pi}{m}} + e^{\frac{2i\beta\pi}{m}} \right) + \omega(a_1 + 2a_2 + a_3)}{\left((1 + \omega)(a_1 + 2a_2 + a_3) - a_1 e^{-\frac{2i\alpha\pi}{m}} - a_3 e^{-\frac{2i\beta\pi}{m}} \right)} & 0 \\ 0 & \frac{b_2 \left(e^{\frac{2i\alpha\pi}{m}} + e^{\frac{2i\beta\pi}{m}} \right) + \omega(b_1 + 2b_2 + b_3)}{\left((1 + \omega)(b_1 + 2b_2 + b_3) - b_1 e^{-\frac{2i\alpha\pi}{m}} - b_3 e^{-\frac{2i\beta\pi}{m}} \right)} \end{bmatrix}.$$

Equation (66) with $\omega = 0.7$, this new scheme yields a much better rate of $\mu = 0.75026$. The choice of $\omega = 0$ is based on numerical experience.

Therefore, the modified smoother will be (66) using a variable ω written in a form similar to (62) as follows:

$$\begin{aligned} & D_\ell(\tilde{\phi}_\ell)_{i,j} \left[(\tilde{\phi}_\ell)_{i+1,j} - (1 + \omega)(\phi_\ell)_{i,j} + \omega(\tilde{\phi}_\ell)_{i,j} \right] \\ & - D_\ell(\tilde{\phi}_\ell)_{i-1,j} \left[(1 + \omega)(\phi_\ell)_{i,j} - \omega(\tilde{\phi}_\ell)_{i,j} - (\tilde{\phi}_\ell)_{i-1,j} \right] \\ & + \lambda^2 D_\ell(\tilde{\phi}_\ell)_{i,j} \left[(\tilde{\phi}_\ell)_{i,j+1} - (1 + \omega)(\phi_\ell)_{i,j} + \omega(\tilde{\phi}_\ell)_{i,j} \right] \\ & - \lambda^2 D_\ell(\tilde{\phi}_\ell)_{i,j-1} \left[(1 + \omega)(\phi_\ell)_{i,j} - \omega(\tilde{\phi}_\ell)_{i,j} - (\tilde{\phi}_\ell)_{i,j-1} \right] = (\tilde{f}_\ell)_{i,j}. \end{aligned} \quad (68)$$

An algorithm for the modified smoother is given by the following:

Table 5 The smoothing rate for a modified local smoother

Outer iterations s	The smoothing rate $\hat{\mu}_s$
1	0.5720
2	0.3170
3	0.2747

Algorithm 5 Modified smoother for multiphase model

Implementation steps of the modified smoother given in Eq.(68) are demonstrated here as an algorithm:

$$\phi_\ell^h \leftarrow \text{Smoother}(\phi_\ell^h, \tilde{f}_\ell^h, \text{maxit}, \omega, K, \text{tol})$$

where $\ell = 1, 2$ and h is the step size on level Ω^h . Set $K = 100$.

for $i = 1 : m_1$

 for $j = 1 : m_2$

 for $iter = 1 : \text{maxit}$

 if $|D_\ell(\tilde{\phi}_\ell^h)_{i,j}| \geq K \max(|D_\ell(\tilde{\phi}_\ell^h)_{i-1,j}|, |D_\ell(\tilde{\phi}_\ell^h)_{i,j-1}|)$ for any ℓ , set $\omega = 0.7$;
 otherwise set $\omega = 0$.

$$\tilde{\phi}_\ell^h \leftarrow \phi_\ell^h,$$

$$A_\ell = D_\ell(\tilde{\phi}_\ell^h)_{i,j}((\tilde{\phi}_\ell^h)_{i+1,j}^h + \omega(\tilde{\phi}_\ell^h)_{i,j}^h) + D_\ell(\tilde{\phi}_\ell^h)_{i-1,j}^h((\tilde{\phi}_\ell^h)_{i-1,j}^h + \omega(\tilde{\phi}_\ell^h)_{i,j}^h),$$

$$B_\ell = D_\ell(\tilde{\phi}_\ell^h)_{i,j}^h((\tilde{\phi}_\ell^h)_{i,j+1}^h + \omega(\tilde{\phi}_\ell^h)_{i,j}^h) + D_\ell(\tilde{\phi}_\ell^h)_{i,j-1}^h((\tilde{\phi}_\ell^h)_{i,j-1}^h + \omega(\tilde{\phi}_\ell^h)_{i,j}^h),$$

$$(\phi_\ell^h)_{i,j} = \frac{A_\ell + \lambda^2 B_\ell - \tilde{f}_{i,j}}{(1 + \omega)(D_\ell(\tilde{\phi}_\ell^h)_{i,j}^h + D_\ell(\tilde{\phi}_\ell^h)_{i-1,j}^h) + \lambda^2 (D_\ell(\tilde{\phi}_\ell^h)_{i,j}^h + D_\ell(\tilde{\phi}_\ell^h)_{i,j-1}^h)}$$

 if $|(\phi_\ell^h)_{i,j} - (\tilde{\phi}_\ell^h)_{i,j}| < \text{tol}$ Stop

 end

 end

 end

The smoothing analysis of the improved smoother is done again in the same steps and is given in Table 4. Clearly, the smoothing rates of the modified smoother are much more acceptable (note the accumulated number of smoothing steps is $3s$ since 3 inner iterations for each outer step are used) (Table 5).

In Table 6, speed comparison of the multigrid with typical local smoother (MG1), multigrid with modified smoother (MG1m), and additive operator splitting method (AOS) in terms of the number of iterations and CPU time is given. Fast convergence of the MG method can clearly be observed from the table. MG algorithms yield a computation time of $\mathcal{O}(N \log N)$ where $N = m_1 \times m_2$.

Table 6 Speed comparison of MG1 (multigrid with typical local smoother), MG1m (multigrid with modified smoother), and AOS methods in terms of number of iterations (“Itr”) and CPU time (“CPU”). Here “–” implies no convergence (to the tolerance) or very slow convergence

Image size	AOS		MG1		MG1m	
	Itr	CPU	Itr	CPU	Itr	CPU
128 × 128	80	22	3	5	2	2
256 × 256	150	193	4	13	2	7
512 × 512	1500	42,600	4	74	2	33
1024 × 1024	–	–	4	525	2	148

Convex Multiphase Image Segmentation Model

The Vese-Chan model (Vese and Chan 2002) discussed in previous section has the advantage that the segmented phases cannot produce vacuum or overlap by construction. Moreover, it considerably reduces the number of level set functions needed and can represent complex boundaries. One of the drawbacks of the Vese-Chan model is that the energy functional of the model is a nonconvex and hence may stuck at local minima. This local minima may lead toward wrong segmentation. In Yang et al. (2014), a convex formulation of the Vese-Chan model (Vese and Chan 2002) is proposed. The energy functional of the Vese-Chan model is given in Equation 57. The convex model is then solved by using the Bregman iterations (Bregman 1967), which are discussed here.

The Bregman Iterations

Some basic definitions and theorems related to Bregman distance and Bregman iterations (Bregman 1967) are given here.

Definition 1. For an energy functional $E(\cdot)$, the Bregman distance between two functions say u and v is given by the following:

$$D_E^q(u, v) = E(u) - E(v) - \langle q, u - v \rangle,$$

where q is in the sub-gradient of $E(\cdot)$, i.e., $\partial E(v)$ at v .

To solve a minimization problem of the following form:

$$\min_u E(u) + \beta W(u), \quad \beta > 0 \quad (69)$$

where $W(\cdot)$ is convex with $\min_u W(u) = 0$, Bregman iterations are defined in the following way:

Definition 2. For given parameter $\beta > 0$, the Bregman iterations are defined as follows:

$$u^{(k+1)} = \arg \min_u D_E^{q^{(k)}}(u, u^{(k)}) + \beta W(u), \quad q^{(k)} \in \partial E(u^{(k)}).$$

The next theorem plays an important role in minimization of the problem types given in (69).

Theorem 1. *The minimization problem given in (69) can be solved by the following Bregman iterations:*

$$u^{(k+1)} = \arg \min_u D_E^{q^{(k)}}(u, u^{(k)}) + \beta W(u) \quad (70)$$

$$= \arg \min_u E(u) - \langle q^{(k)}, u - u^{(k)} \rangle + \beta W(u) \quad (71)$$

where $q^{(k)} \in \partial E(u^{(k)})$. Suppose that $W(\cdot)$ is differentiable; then:

$$q^{(k+1)} = q^{(k)} - \beta \nabla W(u^{(k+1)}). \quad (72)$$

Convergence of the Bregman iterations is proven by Osher et al. in 2005 by stating the following convergence theorem:

Theorem 2. *Consider a minimization problem of type given in (69) and satisfying the condition given therein. Then $u^{(k)}$ obtained by Bregman iterations will satisfy the following conditions:*

1. $u^{(k)}$ decreases monotonically on W : $W(u^{(k+1)}) \leq W(u^{(k)})$.
2. If u^* is a minimizer of W , then $W(u^{(k)}) \leq W(u^*) + \frac{D_E^{q^{(0)}}(u^*, u^{(0)})}{\beta^{(k)}}$.

Convex Multiphase Model

In 2014, Yang et al. proposed a convex formulation of the Vese-Chan four-phase model. For this reconsider Equation (59):

$$\begin{cases} \delta_\epsilon(\phi_1) \left[\mu \nabla \cdot \frac{\nabla \phi_1}{|\nabla \phi_1|} - [T_1 H_\epsilon(\phi_2) + T_2(1 - H_\epsilon(\phi_2))] \right] = 0, \\ \delta_\epsilon(\phi_2) \left[\mu \nabla \cdot \frac{\nabla \phi_2}{|\nabla \phi_2|} - [T_1 H_\epsilon(\phi_1) + T_2(1 - H_\epsilon(\phi_1))] \right] = 0, \end{cases} \quad (73)$$

with Neumann boundary conditions, where $T_1 = (z - c_{11})^2 - (z - c_{01})^2$ and $T_2 = (z - c_{10})^2 - (z - c_{00})^2$. Note that $H_\epsilon(w)$ has a non-compact support, so its derivative $\delta'_\epsilon(w) \neq 0$ for all $w \in R$. Thus the above system of equations may be written as follows:

$$\begin{cases} \left[\mu \nabla \cdot \frac{\nabla \phi_1}{|\nabla \phi_1|} - [T_1 H_\epsilon(\phi_2) + T_2(1 - H_\epsilon(\phi_2))] \right] = 0, \\ \left[\mu \nabla \cdot \frac{\nabla \phi_2}{|\nabla \phi_2|} - [T_1 H_\epsilon(\phi_1) + T_2(1 - H_\epsilon(\phi_1))] \right] = 0. \end{cases} \tag{74}$$

For simplification, suppose the following:

$$\begin{cases} r_1 = [T_1 H_\epsilon(\phi_2) + T_2(1 - H_\epsilon(\phi_2))], \\ r_2 = [T_1 H_\epsilon(\phi_1) + T_2(1 - H_\epsilon(\phi_1))]. \end{cases} \tag{75}$$

Thus the simplified gradient flow equation for (74) becomes the following:

$$\begin{cases} \frac{\partial \phi_1}{\partial t} = \mu \nabla \cdot \frac{\nabla \phi_1}{|\nabla \phi_1|} - r_1, \\ \frac{\partial \phi_2}{\partial t} = \mu \nabla \cdot \frac{\nabla \phi_2}{|\nabla \phi_2|} - r_2. \end{cases} \tag{76}$$

These partial differential equations are Euler-Lagrange’s equation of the following energy functional:

$$\tilde{F}(\phi_1, \phi_2) = \mu |\nabla \phi_1|_1 + \mu |\nabla \phi_2|_1 + \langle \phi_1, r_1 \rangle + \langle \phi_2, r_2 \rangle, \tag{77}$$

where $|\nabla(\cdot)|_1$ is the total variation (TV) norm and $\langle \cdot, \cdot \rangle$ is the inner product, respectively, and may be written as follows:

$$\begin{cases} |\nabla \phi_i|_1 = \int_{\Omega} |\nabla \phi_i(x)| dx = TV(\phi_i) \\ \langle \phi_i, r_i \rangle = \int_{\Omega} \phi_i(x) r_i(x) dx. \end{cases} \tag{78}$$

The energy functional given in Equation (77) is homogeneous in ϕ_i and does not have a minimizer in general. In order to make the minimizer well defined, introduction of some extra constraints on ϕ_i is necessary. As a result the following functional will be considered for minimization:

$$\min_{0 \leq \phi_1, \phi_2 \leq 1} \tilde{F}(\phi_1, \phi_2) = \min_{0 \leq \phi_1, \phi_2 \leq 1} (\mu |\nabla \phi_1|_1 + \mu |\nabla \phi_2|_1 + \langle \phi_1, r_1 \rangle + \langle \phi_2, r_2 \rangle). \tag{79}$$

As $0 \leq \phi_1, \phi_2 \leq 1$, there is no need of using Heaviside function H_i .

$$\begin{cases} \bar{r}_1 = [T_1\phi_2 + T_2(1 - \phi_2)], \\ \bar{r}_2 = [T_1\phi_1 + T_2(1 - \phi_1)]. \end{cases} \quad (80)$$

To use edge information of the image, they used weighted TV norm, which is given by the following:

$$TV_g(\phi_i) = \int_{\Omega} g(|\nabla z|)|\nabla\phi_i|dx = |\nabla\phi_i|_g, \quad (81)$$

where $g(w) = \frac{1}{1+c|z|^2}$ is the edge detector function. Now by using these terms, the energy functional given in Equation (79) becomes the following:

$$\min_{0 \leq \phi_1, \phi_2 \leq 1} \tilde{F}(\phi_1, \phi_2) = \min_{0 \leq \phi_1, \phi_2 \leq 1} (\mu|\nabla\phi_1|_g + \mu|\nabla\phi_2|_g + \langle\phi_1, \bar{r}_1\rangle + \langle\phi_2, \bar{r}_2\rangle). \quad (82)$$

After solving this minimization problem, the following four-phase segmentation domains can be defined by thresholding the level set functions ϕ_1 and ϕ_2 :

$$\begin{cases} \Omega_1 = \{x : \phi_1(x) > 0.5, \phi_2(x) > 0.5\} \\ \Omega_2 = \{x : \phi_1(x) > 0.5, \phi_2(x) < 0.5\} \\ \Omega_3 = \{x : \phi_1(x) < 0.5, \phi_2(x) > 0.5\} \\ \Omega_4 = \{x : \phi_1(x) < 0.5, \phi_2(x) < 0.5\}. \end{cases} \quad (83)$$

Also note that $\mathbf{c} = [c_{11}, c_{10}, c_{01}, c_{00}]$ is the vector of average intensities of image inside $\Omega_1, \Omega_2, \Omega_3, \Omega_4$, respectively. The model given in Equation (82) will give four-phase segmentation and can be extended to n phases, for which $m = \log_2 n$ level set functions will be required. The functional can be written as follows:

$$\min_{0 \leq \phi_i \leq 1} \tilde{F}_n(\phi_1, \phi_2, \dots, \phi_m) = \min_{0 \leq \phi_i \leq 1} \left(\mu \sum_{i=1}^m |\nabla\phi_i|_g + \sum_{i=1}^m \langle\phi_i, \bar{r}_i\rangle \right). \quad (84)$$

Split Bregman Method for the Model

The minimization problem given in (79) can be solved by using the split Bregman method. For this one must introduce two new auxiliary variables $\mathbf{p}_i = \nabla\phi_i, i = 1, 2$. Thus the minimization problem given in Equation (79) can be converted into the following equivalent constrained minimization problem:

$$\begin{aligned} & \min_{\substack{0 \leq \phi_1, \phi_2 \leq 1 \\ \mathbf{p}_1, \mathbf{p}_2}} (\mu|\mathbf{p}_1|_g + \mu|\mathbf{p}_2|_g + \langle \phi_1, \bar{r}_1 \rangle + \langle \phi_2, \bar{r}_2 \rangle), \\ & \text{such that } \mathbf{p}_i = \nabla \phi_i, \quad i = 1, 2. \end{aligned} \tag{85}$$

Corresponding unconstrained minimization problem can be obtained by introducing two quadratic penalty terms $\|\mathbf{p}_i - \nabla \phi_i\|^2, i = 1, 2$, which is given by the following:

$$\begin{aligned} (\phi_1^*, \phi_2^*, \mathbf{p}_1^*, \mathbf{p}_2^*) = \arg \min_{\substack{0 \leq \phi_1, \phi_2 \leq 1 \\ \mathbf{p}_1, \mathbf{p}_2}} & \left(\mu|\mathbf{p}_1|_g + \mu|\mathbf{p}_2|_g + \langle \phi_1, \bar{r}_1 \rangle \right. \\ & \left. + \langle \phi_2, \bar{r}_2 \rangle + \frac{\alpha}{2} \|\mathbf{p}_1 - \nabla \phi_1\|^2 + \frac{\alpha}{2} \|\mathbf{p}_2 - \nabla \phi_2\|^2 \right), \end{aligned} \tag{86}$$

where $\alpha > 0$ is a constant. Bregman iterations for the solution; this unconstrained minimization problem is given in the following theorem:

Theorem 3. *The minimization problem (79) of the proposed model can be converted to a series of optimization problems:*

$$\begin{aligned} (\phi_1^{(k+1)}, \phi_2^{(k+1)}, \mathbf{p}_1^{(k+1)}, \mathbf{p}_2^{(k+1)}) = \arg \min_{\substack{0 \leq \phi_1, \phi_2 \leq 1 \\ \mathbf{p}_1, \mathbf{p}_2}} & \left(\mu|\mathbf{p}_1|_g + \mu|\mathbf{p}_2|_g + \langle \phi_1, \bar{r}_1 \rangle \right. \\ & \left. + \langle \phi_2, \bar{r}_2 \rangle + \frac{\alpha}{2} \|\mathbf{p}_1 - \nabla \phi_1 - \mathbf{b}_1^{(k)}\|^2 \right. \\ & \left. + \frac{\alpha}{2} \|\mathbf{p}_2 - \nabla \phi_2 - \mathbf{b}_2^{(k)}\|^2 \right), \end{aligned} \tag{87}$$

where $\mathbf{b}_i = (b_{ix}, b_{iy}), i = 1, 2$ are the Bregman variables, which can be updated by the following Bregman iterations with initial values $\mathbf{b}_i^0 = (0, 0), i = 1, 2$:

$$\mathbf{b}_i^{(k+1)} = \mathbf{b}_i^{(k)} + \nabla \phi_i^{(k+1)} - \mathbf{p}_i^{(k+1)}, \text{ for } i = 1, 2. \tag{88}$$

To solve the minimization problem given in Equation (79), it is enough to solve the minimization problem given in Equation (87). The iterative minimization scheme can be achieved through the following two steps for solution of Equation (87).

- Keeping $\mathbf{p}_1^{(k)}$ and $\mathbf{p}_2^{(k)}$ and minimizing Equation (87) with respect to ϕ_1 and ϕ_2 give the following:

$$\begin{aligned} (\phi_1^{(k+1)}, \phi_2^{(k+1)}) = \arg \min_{0 \leq \phi_1, \phi_2 \leq 1} & \left(\langle \phi_1, \bar{r}_1^{(k)} \rangle + \langle \phi_2, \bar{r}_2^{(k)} \rangle + \frac{\alpha}{2} \|\mathbf{p}_1 - \nabla \phi_1 - \mathbf{b}_1^{(k)}\|^2 \right. \\ & \left. + \frac{\alpha}{2} \|\mathbf{p}_2 - \nabla \phi_2 - \mathbf{b}_2^{(k)}\|^2 \right). \end{aligned} \tag{89}$$

- Secondly, keeping $\phi_1^{(k+1)}$ and $\phi_2^{(k+1)}$ fixed and minimizing Equation (87) with respect to \mathbf{p}_1 and \mathbf{p}_2 give the following:

$$\begin{aligned} (\mathbf{p}_1^{(k+1)}, \mathbf{p}_2^{(k+1)}) = \arg \min_{\mathbf{p}_1, \mathbf{p}_2} & \left(\mu |\mathbf{p}_1|_g + \mu |\mathbf{p}_2|_g + \frac{\alpha}{2} \|\mathbf{p}_1 - \nabla \phi_1^{(k+1)} - \mathbf{b}_1^{(k)}\|^2 \right. \\ & \left. + \frac{\alpha}{2} \|\mathbf{p}_2 - \nabla \phi_2^{(k+1)} - \mathbf{b}_2^{(k)}\|^2 \right). \end{aligned} \quad (90)$$

Theorem 4. For fixed $\mathbf{b}_1^{(k)}$ and $\mathbf{b}_2^{(k)}$, the minimizer $(\phi_1^{(k+1)}, \phi_2^{(k+1)})$ of the minimization problem (89) will satisfy the following equations:

$$\Delta \phi_1^{(k+1)} = \frac{1}{\alpha} \bar{r}_1^{(k)} + \nabla \cdot (\mathbf{p}_1^{(k)} - \mathbf{b}_1^{(k)}) \quad 0 \leq \phi_1^{(k+1)} \leq 1. \quad (91)$$

$$\Delta \phi_2^{(k+1)} = \frac{1}{\alpha} \bar{r}_2^{(k)} + \nabla \cdot (\mathbf{p}_2^{(k)} - \mathbf{b}_2^{(k)}) \quad 0 \leq \phi_2^{(k+1)} \leq 1. \quad (92)$$

These Laplace equations are solved by using Gauss-Seidel method and obtained the following relation for $\phi_\ell^{(k+1)}$:

$$\left\{ \begin{array}{l} \gamma_{\ell,i,j}^{(k)} = p_{\ell,x,i-1,j}^{(k)} - p_{\ell,x,i,j}^{(k)} + p_{\ell,y,i,j-1}^{(k)} - p_{\ell,y,i,j}^{(k)} - (b_{\ell,x,i-1,j}^{(k)} - b_{\ell,x,i,j}^{(k)} \\ \quad + b_{\ell,y,i,j-1}^{(k)} - b_{\ell,y,i,j}^{(k)}) \\ \tau_{\ell,i,j}^{(k)} = \frac{1}{4} \left(\phi_{\ell,i-1,j}^{(k)} + \phi_{\ell,i+1,j}^{(k)} + \phi_{\ell,i,j-1}^{(k)} + \phi_{\ell,i,j+1}^{(k)} - \frac{1}{\alpha} r_{\ell,i,j}^{(k)} + \gamma_{\ell,i,j}^{(k)} \right) \\ \phi_{\ell,i,j}^{(k+1)} = \max \left\{ \min \left\{ \tau_{\ell,i,j}^{(k)}, 1 \right\}, 0 \right\} \end{array} \right. \quad (93)$$

where $\ell = 1, 2$.

Now to find \mathbf{b}_1 and \mathbf{b}_2 , the following theorem is very useful to note:

Theorem 5. For fixed $\phi_1^{(k+1)}$ and $\phi_2^{(k+1)}$, the minimizer $(\mathbf{p}_1^{(k+1)}, \mathbf{p}_2^{(k+1)})$ of the minimization problem given in Equation (90) will satisfy the following vector shrinkage operator:

$$\mathbf{p}_1^{(k+1)} = \text{shrinkage}_g \left(\mathbf{b}_1^{(k)} + \nabla \phi_1^{(k+1)}, \frac{1}{\alpha} \right) = \text{shrinkage} \left(\mathbf{b}_1^{(k)} + \nabla \phi_1^{(k+1)}, \frac{\rho}{\alpha} \right) \quad (94)$$

$$p_2^{(k+1)} = shrinkage_g \left(b_2^{(k)} + \nabla \phi_2^{(k+1)}, \frac{1}{\alpha} \right) = shrinkage \left(b_2^{(k)} + \nabla \phi_2^{(k+1)}, \frac{\rho}{\alpha} \right) \tag{95}$$

where the vector shrinkage operator is given by the following:

$$shrinkage(x, \xi) = \begin{cases} \frac{x}{|x|} \max(|x| - \xi, 0), & x \neq 0 \\ 0, & x = 0 \end{cases} \tag{96}$$

For further details and experimental results of the proposed model and method, see Yang et al. (2014). In Fig. 1, the proposed method is tested on an artificial image. In Fig. 2, results of the proposed model on a real MRI image are given.

A Three-Stage Approach for Multiphase Segmentation Degraded Color Images

In 2017, Cai et al. proposed a smoothing, lifting, and thresholding method with three stages for multiphase segmentation of color images corrupted by different degradations: noise, information loss, and blur. The proposed method works in the following steps: in step one, a smooth restored image is obtained by applying the convex models Cai et al. (2013) and Chan et al. (2014) on each channel of original color image space. In the second stage, the smooth color image is transformed to

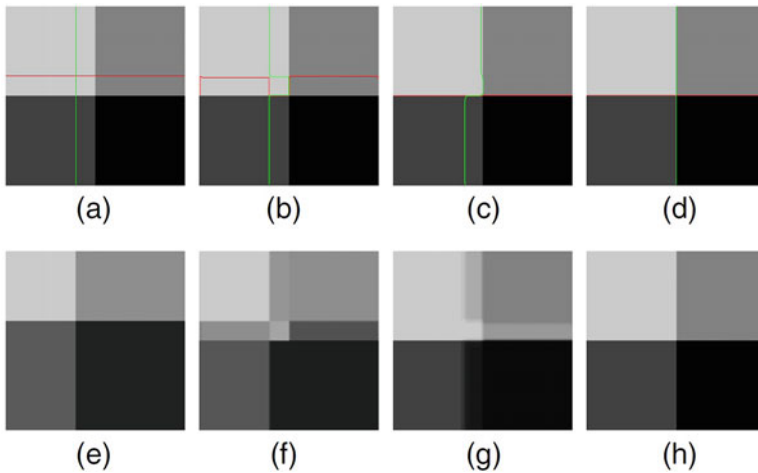


Fig. 1 Application of the proposed model to a simple synthetic image. (a)–(d): The active contour evolving process from the initial contour to the final contour. (e)–(h): The corresponding fitting images z at different iterations

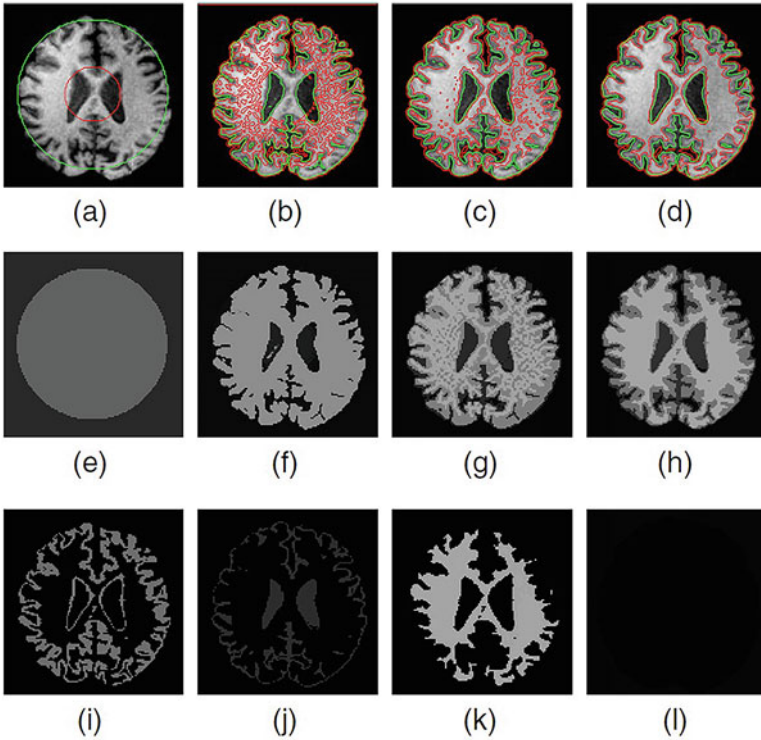


Fig. 2 Application of the proposed model to a brain MR image. (a)–(d): The active contour evolving process. (e)–(h): The evolution process of the fitting image z . (i)–(l): The final four segments with four averages $c_{11} = 113.1278$, $c_{10} = 48.3514$, $c_{01} = 167.2793$, and $c_{00} = 4.0692$

a secondary color space, which provides complementary information, and then a new vector-valued image is formed by using all channels from both color spaces (original and transformed). In stage 3, a multichannel thresholding is applied on the obtained vector-valued image to get segmented image.

Stage 1: Restoration and Smoothing of Given Image

Let z be a color image with d channels say z_i for $i = 1, 2, \dots, d$;, the following energy functional is considered:

$$E(z_i) = \frac{\lambda}{2} \int_{\Omega} \Psi_i(z_i - Ku_i)^2 dx + \frac{\mu}{2} \int_{\Omega} |\nabla u_i|^2 dx + \int_{\Omega} |\nabla u_i| dx, \quad i = 1, 2, \dots, d, \tag{97}$$

where $\Psi_i(\cdot)$ is the characteristic function and is a region descriptor. For existence and uniqueness of the minimizer of the above functional, see Cai et al. (2017).

The above model (97) is considered in discrete setting and is solved for the unique minimizer \bar{u}_i for each channel i by using different methods such as primal-dual method (Chambolle and Pock 2011; Chen et al. 2014), alternating direction method (Boyd et al. 2010), and split Bregman method (Goldstein and Osher 2009; Bregman 1967). Once \bar{u}_i is found, it is rescaled onto $[0, 1]$ and hence $\{\bar{u}_i\}_{i=1}^d \in [0, 1]^d$.

Stage 2: Dimension Lifting with Secondary Color Space

In first stage, a restored smooth image \bar{u}_i is obtained, whereas in this stage, the dimension lifting is performed on \bar{u}_i to extract more additional information from a different color space that help the segmentation in the later stage. Popular choices for other less correlated color spaces are the HSV (hue, saturation, and value), the CB (chromaticity-brightness), HSI (hue, saturation, and intensity), and the Lab (perceived lightness, red-green, and yellow-blue). Note that the Lab is a better color space than RGB, HSV, and HSI for segmentation. In this stage the authors created the Lab color space with the aim to be perceptually uniform in the sense that the numerical difference between two colors is proportional to perceived color difference. Here, the Lab is used as the additional color space, where the L channel correlates to perceived lightness, while the a and b channels correlate approximately with red-green and yellow-blue, respectively.

Let \bar{u}' denote Lab transform of \bar{u} , rescaling all the channels of \bar{u}' on the interval $[0, 1]$ to yield an image denoted by $\bar{u}' \in [0, 1]^3$. Introduce a new image \bar{u}^* by stacking \bar{u} and \bar{u}' having six channels as follows:

$$\bar{u}^* = (\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}'_1, \bar{u}'_2, \bar{u}'_3).$$

This image will be used for segmentation in the next stage.

Stage 3: Segmentation

Segmentation of the vector-valued image \bar{u}^* , obtained from the second stage in K segments, is done by using thresholding. This is based on the K-means algorithm (Kanungo et al. 2002) because of its simplicity and good asymptotic properties. According to the value of K , the algorithm clusters all points of $\{\bar{u}^*(x) : x \in \Omega\}$ into K Voronoi-shaped cells, say $\Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_K = \Omega$. The mean vector $c_k \in \Omega^6$ on each cell Ω_k by the following:

$$c_k = \frac{\int_{\Omega_k} \bar{u}^*(x) dx}{\int_{\Omega_k} dx}, \quad k = 1, 2, \dots, K. \quad (98)$$

Recall that each entry $c_k[i]$ for $i = 1, 2, \dots, 6$ is a value belonging to $\{R, G, B, L, a, b\}$, respectively. Using $c_k[i]$, \bar{u}^* can be divided into K phases by the following:

$$S_k := \{x \in \Omega : \|\bar{u} * -c_k\|_2 = \min_{1 \leq j \leq K} \|\bar{u} * -c_j\|_2\}, \quad k = 1, 2, \dots, K. \quad (99)$$

Clearly $\cup_{k=1}^K S_k = \Omega$ and $\cap_{k=1}^K S_k = \cdot$. For further details see Cai et al. (2013, 2017).

Selective Segmentation Models

Usually, two types of image segmentation problems are discussed in image processing: one is global segmentation, in which the complete image is segmented into all possible segments/regions, and the other one is the selective segmentation, in which a region of interest is segmented in an image. In previous sections, all discussions were about global segmentation. Another possible name used for selective segmentation in literature is interactive segmentation. This section is mainly devoted to selective segmentation.

Image Segmentation Under Geometrical Conditions

A model which is used for selective segmentation based on some geometrical constraints (like a set of points near the region of interest ROI) is proposed by Guyader and Gout (2008). The proposed model is based on the geodesic active contour model (Caselles et al. 1997) and geometrical constraints. Let $B = \{(x_i^*, y_i^*) \in \Omega, 1 \leq i \leq n_p\} \subset \Omega$ be the set of n_p distinct points near the boundary of the region of interest in the given image $z(x, y)$. The aim is to find an optimal contour $\Gamma \subset \Omega$ that best approaches the points from the set B while detecting the desired object in an image. The model works in the following way: let g be an edge detector function defined as follows:

$$g(w) = \frac{1}{1 + w^2}.$$

It must be noted that $g(|\nabla z(x, y)|)$ approaches to zero near edges in an image as discussed earlier. The purpose of the edge detector function g is to stop the evolving curve on edges/boundaries of the objects (ROI). A function $d(x, y)$ (distance metric) is introduced to stop the evolving curve near the geometrical points given in set B . This function $d(x, y)$ can be defined in the following way (Guyader and Gout 2008):

$$\forall (x, y) \in \Omega, \quad d(x, y) = \prod_{i=1}^{n_p} \left(1 - e^{-\frac{(x - x_i^*)^2}{2\sigma^2}} e^{-\frac{(y - y_i^*)^2}{2\sigma^2}} \right). \quad (100)$$

There exist other distance metrics d as well like the following:

$$d(x, y) = \text{distance}((x, y), B) = \min_{(x_i^*, y_i^*) \in B} \left| (x, y) - (x_i^*, y_i^*) \right|$$

for all $(x, y) \in \Omega$ and $i = 1, 2, \dots, n_p$; see for others (Gout et al. 2005). Clearly $d(x, y)$ acts locally and will be approximately 0 in the neighborhood of points of set B . The aim is to find a contour Γ along which either $d \simeq 0$ or $g \simeq 0$. The following energy functional is proposed:

$$F(\Gamma) = \int_{\Gamma} d(x, y)g(|\nabla z(x, y)|)ds. \quad (101)$$

The contour Γ will stop at local minima where $d \simeq 0$ (in the neighborhood of points for B) or $g \simeq 0$ (near object boundaries). By introducing level set function ϕ , functional given in Equation (101) becomes the following:

$$F_{\epsilon}(\phi(x, y)) = \int_{\Omega} d(x, y)g(|\nabla z(x, y)|)\delta_{\epsilon}(\phi)|\nabla\phi(x, y)|dxdy, \quad (102)$$

where $\delta_{\epsilon}(\phi)$ is the regularized delta function. The functional $F_{\epsilon}(\phi(x, y))$ will be minimized with respect to $\phi(x, y)$, by considering the following minimization problem:

$$\min_{\phi(x, y)} F_{\epsilon}(\phi(x, y)), \quad (103)$$

where $F_{\epsilon}(\phi(x, y))$ is given in Equation (102). First variation of the functional given in Equation (103) leads to the following Euler-Lagrange's equation:

$$-\delta_{\epsilon}(\phi(x, y))\nabla \cdot \left(d(x, y)g(|\nabla z(x, y)|)\frac{\nabla\phi(x, y)}{|\nabla\phi(x, y)|} \right) = 0.$$

Guyader and Gout (2008) solved the following evolution equation by introducing artificial time step t :

$$\frac{\partial\phi(x, y)}{\partial t} = \delta_{\epsilon}(\phi(x, y))\nabla \cdot \left(d(x, y)g(|\nabla z(x, y)|)\frac{\nabla\phi(x, y)}{|\nabla\phi(x, y)|} \right) \quad (104)$$

with the boundary condition:

$$\frac{\partial\phi(x, y)}{\partial \vec{n}} = 0,$$

where \vec{n} is the outward unit normal to the boundary $\partial\Omega$. Clearly the quantity $\frac{\partial\phi(x, y)}{\partial t}$ tends to 0 when a local minimum is achieved. In other words, if the model converges, the curve will not evolve any more since a steady state has been reached. A rescaling can be made so that the motion is applied to all level sets by replacing $\delta_{\epsilon}(\phi(x, y))$ by $|\nabla\phi(x, y)|$. Furthermore, it makes the flow independent of

the scaling of ϕ (Alvarez et al. 1992; Zhao et al. 2000). Thus they considered the following evolution problem:

$$\begin{aligned}\phi(x, y, 0) &= \phi_0(x, y), \\ \frac{\partial \phi(x, y)}{\partial t} &= |\nabla \phi(x, y)| \nabla \cdot \left(d(x, y) g(|\nabla z(x, y)|) \frac{\nabla \phi(x, y)}{|\nabla \phi(x, y)|} \right), \\ \frac{\partial \phi(x, y)}{\partial \vec{n}} &= 0 \quad \text{on } \partial \Omega,\end{aligned}\quad (105)$$

where $\phi_0(x, y)$ is the initial value of $\phi(x, y)$. To avoid the evolving curve to stuck at local minima, an extra term known as ‘‘balloon term’’ is given by $\alpha d(x, y) g(|\nabla z(x, y)|)$, where $\alpha > 0$. Thus the following evolution problem is considered for solution:

$$\begin{aligned}\phi(x, y, 0) &= \phi_0(x, y) \\ \frac{\partial \phi(x, y)}{\partial t} &= |\nabla \phi(x, y)| \nabla \cdot \left(d(x, y) g(|\nabla z(x, y)|) \frac{\nabla \phi(x, y)}{|\nabla \phi(x, y)|} \right) \\ &\quad + \alpha d(x, y) g(|\nabla z(x, y)|) |\nabla \phi(x, y)| \\ \frac{\partial \phi(x, y)}{\partial n} &= 0 \quad \text{on } \partial \Omega.\end{aligned}\quad (106)$$

After some manipulations:

$$\begin{aligned}\frac{\partial \phi(x, y)}{\partial t} &= |\nabla \phi(x, y)| d(x, y) g(|\nabla z(x, y)|) \nabla \cdot \left(\frac{\nabla \phi(x, y)}{|\nabla \phi(x, y)|} \right) \\ &\quad + \nabla (d(x, y) g(|\nabla z(x, y)|)) \cdot \nabla \phi + \alpha d(x, y) g(|\nabla z(x, y)|) |\nabla \phi(x, y)|.\end{aligned}\quad (107)$$

This elliptic-type partial differential equation can be solved by using any time marching scheme. One of the best among those is the additive operator splitting (AOS) method (Weickert et al. 1997), which is discussed earlier.

Active Contour-Based Image Selective Model

Badshah and Chen in (2010) proposed a model for selective segmentation of gray images, which is the extension of Gout model (Guyader and Gout 2008) by using region information of the image combined with geodesic contour model. The following minimization problem was proposed:

$$\min_{\phi(x, y), c_1, c_2} F(\phi(x, y), c_1, c_2), \quad (108)$$

where:

$$\begin{aligned}
 F(\Gamma, c_1, c_2) &= \mu \int_{\Gamma} d(x, y)g(|\nabla z(x, y)|)ds + \lambda_1 \int_{\text{inside}(\Gamma)} |z(x, y) - c_1|^2 dx dy \\
 &+ \lambda_2 \int_{\text{outside}(\Gamma)} |z(x, y) - c_2|^2 dx dy,
 \end{aligned}
 \tag{109}$$

where μ is a positive parameter. Clearly if $\lambda_1 = \lambda_2 = 0$ and $\mu = 1$, then minimization problem (109) reduces to minimization problem (101).

Using level set function and introducing regularized Heaviside function, the energy functional given in Equation (109) becomes the following:

$$\min_{\phi(x,y), c_1, c_2} F_{\epsilon}(\phi(x, y), c_1, c_2),
 \tag{110}$$

where:

$$\begin{aligned}
 &F_{\epsilon}(\phi(x, y), c_1, c_2) \\
 &= \mu \int_{\Omega} d(x, y)g(|\nabla z(x, y)|)\delta_{\epsilon}(\phi(x, y))|\nabla \phi(x, y)| dx dy \\
 &+ \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_{\epsilon}(\phi(x, y)) dx dy + \lambda_2 \int_{\Omega} |z(x, y) \\
 &- c_2|^2 (1 - H_{\epsilon}(\phi(x, y))) dx dy.
 \end{aligned}
 \tag{111}$$

Note that c_1 and c_2 are the average intensities as discussed earlier. Introducing $G(x, y) = d(x, y)g(|\nabla z(x, y)|)$ and then taking first variation of the proposed functional with respect to ϕ through Gâteaux derivatives lead to the following Euler-Lagrange’s equation:

$$\begin{aligned}
 &\delta_{\epsilon}(\phi)\mu \nabla \cdot \left(G(x, y) \frac{\nabla \phi}{|\nabla \phi|} \right) \\
 &- \delta_{\epsilon}(\phi)(\lambda_1(z(x, y) - c_1)^2 - \lambda_2(z(x, y) - c_2)^2) = 0, \quad \text{on } \Omega \\
 &G(x, y) \frac{\delta_{\epsilon}(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial \bar{n}} = 0, \quad \text{on } \partial \Omega.
 \end{aligned}
 \tag{112}$$

Solution of this elliptic PDE is the steady-state solution of the following evolution equation (parabolic PDE):

$$\begin{aligned}
 \frac{\partial \phi}{\partial t} &= \delta_{\epsilon}(\phi)\mu \nabla \cdot \left(G(x, y) \frac{\nabla \phi}{|\nabla \phi|} \right) \\
 &- \delta_{\epsilon}(\phi)(\lambda_1(z(x, y) - c_1)^2 - \lambda_2(z(x, y) - c_2)^2)
 \end{aligned}
 \tag{113}$$

with the boundary condition:

$$G(x, y) \frac{\delta_\epsilon(\phi)}{|\nabla\phi|} \frac{\partial\phi}{\partial\vec{n}} \Big|_{\partial\Omega} = 0,$$

where \vec{n} is the unit normal vector to the boundary of Ω . At steady state $\frac{\partial\phi}{\partial t} = 0$, which means the local minimum has been reached. After some manipulation, the above equation becomes the following:

$$\begin{cases} \phi(x, y, 0) = \phi_0(x, y) \\ \frac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi(x, y))\nabla \cdot \left(G(x, y) \frac{\nabla\phi}{|\nabla\phi|} \right) \\ -\delta_\epsilon(\phi)(\lambda_1(z(x, y) - c_1)^2 - \lambda_2(z(x, y) - c_2)^2), \\ G(x, y) \frac{\delta_\epsilon(\phi)}{|\nabla\phi|} \frac{\partial\phi}{\partial n} \Big|_{\partial\Omega} = 0. \end{cases} \quad (114)$$

A term $\alpha G(x, y)|\nabla\phi|$ (known as a balloon term) could be added to speed up the convergence of the evolution equation as discussed in the previous section, where α is a positive constant. This term prevents the curve from stopping on a nonsignificant local minimum and is also of importance when initializing the process with a curve inside the object to be detected (Guyader and Gout 2008). Thus Equation (114) with balloon term can be written as follows:

$$\begin{cases} \phi(x, y, 0) = \phi_0(x, y) \\ \frac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi(x, y))\nabla \cdot \left(G(x, y) \frac{\nabla\phi}{|\nabla\phi|} \right) \\ -\delta_\epsilon(\phi)(\lambda_1(z(x, y) - c_1)^2 - \lambda_2(z(x, y) - c_2)^2) + \alpha G(x, y)|\nabla\phi|, \\ G(x, y) \frac{\delta_\epsilon(\phi)}{|\nabla\phi|} \frac{\partial\phi}{\partial n} \Big|_{\partial\Omega} = 0, \end{cases} \quad (115)$$

after some manipulation leads to the following:

$$\begin{cases} \phi(x, y, 0) = \phi_0(x, y) \\ \frac{\partial\phi}{\partial t} = \mu\delta_\epsilon(\phi(x, y))G(x, y)\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) + \mu\delta_\epsilon(\phi(x, y))\nabla G(x, y) \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \\ -\delta_\epsilon(\phi)(\lambda_1(z - c_1)^2 - \lambda_2(z - c_2)^2) + \alpha G(x, y)|\nabla\phi|, \\ G(x, y) \frac{\delta_\epsilon(\phi)}{|\nabla\phi|} \frac{\partial\phi}{\partial n} \Big|_{\partial\Omega} = 0. \end{cases} \quad (116)$$

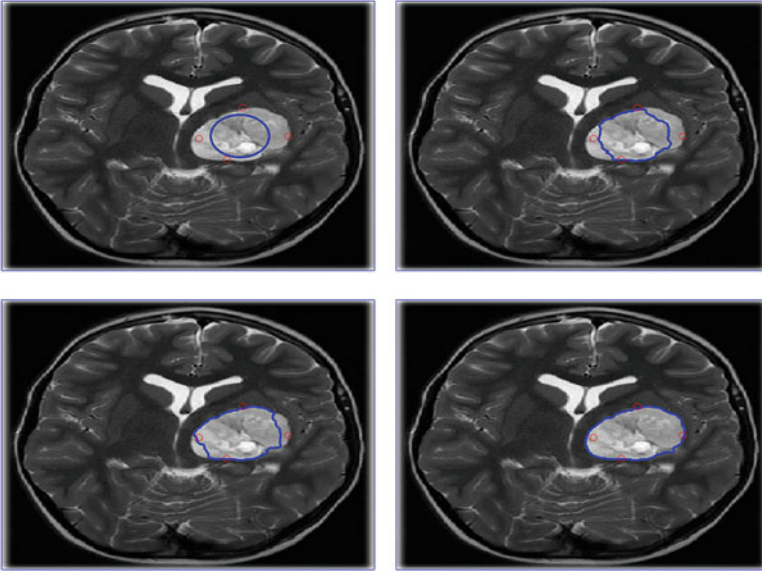


Fig. 3 To detect a tumor in a real brain MRI image with 4 markers with initial guess, $\phi_0 = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r_0$, where x_0 and y_0 are the average of x , y -components of the markers, respectively. $\mu = (\text{size of } z)^2/10$, $\lambda_1 = 0.0001$, $\lambda_2 = 0.0001$, $\alpha = -1.51 \times 10^{-2}$, and $\sigma = 4$

Existence and uniqueness of the solution can be proven along similar lines to Guyader and Gout (2008). This Equation (116) is solved by using time marching scheme like semi-implicit and additive operator splitting methods, which are discussed in the previous sections.

In Fig. 3, the proposed model is tested on a real brain MRI image to detect a tumor by taking four marker points near tumor in brain MR image. The initial condition is $\phi_0 = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r_0$, where x_0 and y_0 are the average of x , y -components of the markers, respectively. The other parameters used are $\mu = (\text{size of } z)^2/10$, $\lambda_1 = 0.0001$, $\lambda_2 = 0.0001$, $\alpha = -1.51 \times 10^{-2}$, and $\sigma = 4$. Top left figure is the original image with initial data, and top right figure is the result after 10 iterations. Bottom left figure is the result after 40 iterations, and bottom right figure is the final result after 200 iterations.

Parameter's selection. Initialization of the level set $\phi_0 = \sqrt{(x - x_0)^2 + (y - y_0)^2} - r_0$ is done automatically by taking x_0 and y_0 as the average of x , y -components of the marker's points, and r_0 is the minimum distance of the center from all marker's points. In most of the cases, $\lambda_1 = \lambda_2$ and may be taken small values of them. α controls the expanding of contour near edges of the object region whose values are near to zero and can be positive or negative. And μ is usually taken as multiple of the size of the given image, and this parameter must be chosen very carefully as the model is very sensitive with the selection of this parameter.

Dual-Level Set Selective Segmentation Model

In 2012, Rada and Chen proposed a selective segmentation model, in which two level sets (global and local) are constructed. Global-level set ϕ_G carries out global segmentation and local-level set ϕ_L carries out local selective segmentation. Introduce the following:

$$\begin{cases} \Gamma_L = \partial\Omega_L = \{(x, y) \in \Omega_L \mid \phi_L(x, y) = 0\} \\ \text{inside } (\Gamma_L) = \Omega_L = \{(x, y) \in \Omega_L \mid \phi_L(x, y) > 0\} \\ \text{outside } (\Gamma_L) = \Omega \setminus \overline{\Omega_L} = \{(x, y) \in \Omega_L \mid \phi_L(x, y) < 0\} \end{cases} \quad (117)$$

$$\begin{cases} \Gamma_G = \partial\Omega_G = \{(x, y) \in \Omega \mid \phi_G(x, y) = 0\} \\ \text{inside } (\Gamma_G) = \Omega_G = \{(x, y) \in \Omega \mid \phi_G(x, y) > 0\} \\ \text{outside } (\Gamma_G) = \Omega \setminus \overline{\Omega_G} = \{(x, y) \in \Omega \mid \phi_G(x, y) < 0\} \end{cases} \quad (118)$$

Note that $\Omega_L \subset \Omega_G \subset \Omega$. To look for all features Ω_G in the whole image domain Ω and the selective features Ω_L in the local domain Ω_G , they proposed the following energy functional by using regularized Heaviside function:

$$\begin{aligned} & \min_{\phi_L(x,y), \phi_G(x,y), c_1, c_2} F_\epsilon(\phi_L(x, y), \phi_G(x, y), c_1, c_2) \\ &= \mu_1 \int_{\Omega} d(x, y) g(|\nabla z(x, y)|) \delta_\epsilon(\phi_L(x, y)) |\nabla \phi_L(x, y)| H_\epsilon(\phi_G(x, y) + \gamma) dx dy \\ &+ \frac{\mu_L}{2} \int_{\Omega} (|\nabla \phi_L(x, y)| - 1)^2 dx dy \\ &+ \mu_2 \int_{\Omega} g(|\nabla z(x, y)|) \delta_\epsilon(\phi_G(x, y)) |\nabla \phi_G(x, y)| dx dy \\ &+ \frac{\mu_G}{2} \int_{\Omega} (|\nabla \phi_G(x, y)| - 1)^2 dx dy + \lambda_{1G} \int_{\Omega} |z(x, y) - c_1|^2 H_\epsilon(\phi_G(x, y)) dx dy \\ &+ \lambda_{2G} \int_{\Omega} |z(x, y) - c_2|^2 (1 - H_\epsilon(\phi_G(x, y))) dx dy \\ &+ \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_\epsilon(\phi_L(x, y)) dx dy \\ &+ \lambda_2 \int_{\Omega} |z(x, y) - c_1|^2 (1 - H_\epsilon(\phi_L(x, y))) H(\phi_G(x, y)) dx dy \\ &+ \lambda_3 \int_{\Omega} |z(x, y) - c_2|^2 (1 - H_\epsilon(\phi_L(x, y))) (1 - H_\epsilon(\phi_G(x, y))) dx dy \end{aligned} \quad (119)$$

Here μ_L, μ_G are positive. Keeping ϕ fixed and minimizing with respect to c_1 and c_2 lead the following:

$$c_1 = \frac{\lambda_{1G} \int_{\Omega} z H_{\epsilon}(\phi_G) dx dy + \lambda_1 \int_{\Omega} z H_{\epsilon}(\phi_L) dx dy + \lambda_2 \int_{\Omega} z (1 - H_{\epsilon}(\phi_L)) H_{\epsilon}(\phi_G) dx dy}{\lambda_{1G} \int_{\Omega} H_{\epsilon}(\phi_G) dx dy + \lambda_1 \int_{\Omega} H_{\epsilon}(\phi_L) dx dy + \lambda_2 \int_{\Omega} (1 - H_{\epsilon}(\phi_L)) H_{\epsilon}(\phi_G) dx dy}$$

$$c_2 = \frac{\lambda_{2G} \int_{\Omega} z (1 - H_{\epsilon}(\phi_G)) dx dy + \lambda_3 \int_{\Omega} z (1 - H_{\epsilon}(\phi_L)) (1 - H_{\epsilon}(\phi_G)) dx dy}{\lambda_{2G} \int_{\Omega} (1 - H_{\epsilon}(\phi_G)) dx dy + \lambda_3 \int_{\Omega} (1 - H_{\epsilon}(\phi_L)) (1 - H_{\epsilon}(\phi_G)) dx dy}$$

First variation of the functional given in Equation (119) with respect to ϕ_L and letting $G(x, y) = d(x, y)g(|\nabla z(x, y)|)$ lead the following:

$$\left\{ \begin{aligned} &\mu_1 \delta_{\epsilon}(\phi_L) \nabla \cdot \left(G(x, y) H_{\epsilon}(\phi_G + \gamma) \frac{\nabla \phi_L}{|\nabla \phi_L|} \right) + \mu_L \nabla \cdot \left(\left(1 - \frac{1}{|\nabla \phi_L|} \right) \nabla \phi_L \right) \\ &+ \delta_{\epsilon}(\phi_L) \left(-\lambda_1 (z(x, y) - c_1)^2 + \lambda_2 (z(x, y) - c_1)^2 H_{\epsilon}(\phi_G) \right. \\ &\quad \left. + \lambda_3 (z(x, y) - c_2)^2 (1 - H_{\epsilon}(\phi_G)) \right) = 0, \quad \text{in } \Omega \\ &\frac{\partial \phi_L}{\partial n} = 0, \quad \text{on } \partial \Omega \end{aligned} \right. \tag{120}$$

with Neumann boundary conditions. In similar way, Euler-Lagrange’s equation can be derived for ϕ_G . Introducing balloon terms as discussed earlier leads to the following equations:

$$\left\{ \begin{aligned} &\mu_1 \delta_{\epsilon}(\phi_L) \nabla \cdot \left(G(x, y) H_{\epsilon}(\phi_G + \gamma) \frac{\nabla \phi_L}{|\nabla \phi_L|} \right) + \mu_L \nabla \cdot \left(\left(1 - \frac{1}{|\nabla \phi_L|} \right) \nabla \phi_L \right) \\ &+ \delta_{\epsilon}(\phi_L) \left(-\lambda_1 (z(x, y) - c_1)^2 + \lambda_2 (z(x, y) - c_1)^2 H_{\epsilon}(\phi_G) \right. \\ &\quad \left. + \lambda_3 (z(x, y) - c_2)^2 (1 - H_{\epsilon}(\phi_G)) \right) + \alpha G(x, y) |\nabla \phi_L| = 0, \quad \text{in } \Omega \\ &\frac{\partial \phi_L}{\partial n} = 0, \quad \text{on } \partial \Omega \end{aligned} \right. \tag{121}$$

and:

$$\left\{ \begin{aligned} &\mu_2 \delta_{\epsilon}(\phi_G) \nabla \cdot \left(g(x, y) \frac{\nabla \phi_G}{|\nabla \phi_G|} \right) + \mu_G \nabla \cdot \left(\left(1 - \frac{1}{|\nabla \phi_G|} \right) \nabla \phi_G \right) \\ &+ \delta_{\epsilon}(\phi_G + \gamma) \left(-\mu_1 G(x, y) |\nabla H_{\epsilon}(\phi_L)| \right) + \delta_{\epsilon}(\phi_G) \left(-\lambda_{1G} (z(x, y) - c_1)^2 \right. \\ &\quad \left. + \lambda_{2G} (z(x, y) - c_2)^2 - \lambda_2 (z(x, y) - c_1)^2 (1 - H(\phi_L)) \right. \\ &\quad \left. + \lambda_3 (z(x, y) - c_2)^2 (1 - H(\phi_L)) + \alpha g(x, y) |\nabla \phi_G| \right) = 0, \quad \text{in } \Omega \\ &\frac{\partial \phi_G}{\partial n} = 0, \quad \text{on } \partial \Omega \end{aligned} \right. \tag{122}$$

An additive operator splitting method (time marching scheme) is used to the respective parabolic partial differential equation:

$$\left\{ \begin{array}{l} \frac{\partial \phi_G}{\partial t} = \mu_1 \delta_\epsilon (\phi_L) \nabla \cdot \left(G(x, y) H_\epsilon (\phi_G + \gamma) \frac{\nabla \phi_L}{|\nabla \phi_L|} \right) \\ \quad + \mu_L \nabla \cdot \left(\left(1 - \frac{1}{|\nabla \phi_L|} \right) \nabla \phi_L \right) \\ \quad + \delta_\epsilon (\phi_L) \left(-\lambda_1 (z(x, y) - c_1)^2 + \lambda_2 (z(x, y) - c_1)^2 H_\epsilon (\phi_G) \right. \\ \quad \left. + \lambda_3 (z(x, y) - c_2)^2 (1 - H_\epsilon (\phi_G)) \right) + \alpha G(x, y) |\nabla \phi_L|, \quad \text{in } \Omega \\ \frac{\partial \phi_L}{\partial n} = 0, \quad \text{on } \partial \Omega \end{array} \right. \quad (123)$$

and:

$$\left\{ \begin{array}{l} \frac{\partial \phi_G}{\partial t} = \mu_2 \delta_\epsilon (\phi_G) \nabla \cdot \left(g(x, y) \frac{\nabla \phi_G}{|\nabla \phi_G|} \right) + \mu_G \nabla \cdot \left(\left(1 - \frac{1}{|\nabla \phi_G|} \right) \nabla \phi_G \right) \\ \quad + \delta_\epsilon (\phi_G + \gamma) \left(-\mu_1 G(x, y) |\nabla H_\epsilon (\phi_L)| \right) + \delta_\epsilon (\phi_G) \left(-\lambda_{1G} (z(x, y) - c_1)^2 \right. \\ \quad \left. + \lambda_{2G} (z(x, y) - c_2)^2 - \lambda_2 (z(x, y) - c_1)^2 (1 - H (\phi_L)) \right) \\ \quad + \lambda_3 (z(x, y) - c_2)^2 (1 - H (\phi_L)) + \alpha g(x, y) |\nabla \phi_G|, \quad \text{in } \Omega \\ \frac{\partial \phi_G}{\partial n} = 0, \quad \text{on } \partial \Omega \end{array} \right. \quad (124)$$

For further solution steps and experimental results, see Rada and Chen (2012). The model produces good and accurate results in hard images and images having overlapped regions but has high computational cost due to solution of system of PDEs for updating two level sets.

One-Level Selective Segmentation Model

In Rada and Chen (2013), proposed a one-level selective segmentation model. Consider the set of some geometrical points in the image domain as discussed earlier. They proposed the following energy functional:

$$\begin{aligned} \min_{\Gamma, c_2} F(\Gamma, c_2) = \min_{\Gamma, c_2} & \left\{ \mu \int_{\Gamma} g(|\nabla z(x, y)|) dx dy \right. \\ & \left. + \lambda_1 \int_{\text{inside}(\Gamma)} |z(x, y) - c_1|^2 dx dy \right\} \end{aligned}$$

$$\begin{aligned}
 & + \lambda_2 \int_{\text{outside}(\Gamma)} |z(x, y) - c_2|^2 dx dy \\
 & + \nu \left\{ \left(\int_{\text{inside}(\Gamma)} dx dy - A_1 \right)^2 + \left(\int_{\text{outside}(\Gamma)} dx dy - A_2 \right)^2 \right\}, \quad (125)
 \end{aligned}$$

where $\lambda_1, \lambda_2, \mu, \nu$ are positive constants and g is the edge detector function which was defined earlier. Note that c_1 is known, which is the average intensity of the polygon constructed in the image by using the marker points. c_2 and Γ are unknown and need to found by minimizing the functional in (125). A_1 and A_2 are the areas of the region inside and outside polygon constructed from the marker points. Using level set function and regularized Heaviside function, the functional given in (125) takes the following form:

$$\begin{aligned}
 \min_{\phi(x,y),c_2} F_\epsilon(\phi(x, y), c_2) &= \mu \int_{\Omega} g(|\nabla z(x, y)|) \delta_\epsilon(\phi(x, y)) |\nabla(\phi(x, y))| \\
 &+ dx dy + \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_\epsilon(\phi(x, y)) dx dy \\
 &+ \lambda_2 \int_{\Omega} |z(x, y) - c_2|^2 (1 - H_\epsilon(\phi(x, y))) dx dy \\
 &+ \nu \left\{ \left(\int_{\Omega} H_\epsilon(\phi(x, y)) dx dy - A_1 \right)^2 \right. \\
 &\left. + \left(\int_{\Omega} (1 - H_\epsilon(\phi(x, y))) dx dy - A_2 \right)^2 \right\} dx dy. \quad (126)
 \end{aligned}$$

Keeping ϕ fixed and minimizing this functional with respect to c_2 give the following:

$$c_2(\phi(x, y)) = \frac{\int_{\Omega} z(x, y) (1 - H_\epsilon(\phi(x, y))) dx dy}{\int_{\Omega} (1 - H_\epsilon(\phi(x, y))) dx dy}$$

and keeping c_2 fixed and if the marker points are not near to the boundary of the region of interest. Thus first variation with respect to ϕ gives the following Euler-Lagrange's equation:

$$\begin{aligned}
 \delta_\epsilon(\phi) &\left\{ \mu \nabla \cdot \left(g(|\nabla z(x, y)|) \frac{\nabla \phi}{|\nabla \phi|} \right) - \left[\lambda_1 (z(x, y) - c_1)^2 - \lambda_2 (z(x, y) - c_2)^2 \right] \right. \\
 &\left. - \nu \left[\left(\int_{\Omega} H dx dy - A_1 \right) - \left(\int_{\Omega} (1 - H) dx dy - A_2 \right) \right] \right\} = 0 \text{ in } \Omega, \quad (127)
 \end{aligned}$$

with Neumann boundary condition. If the marker points are near the boundary of the ROI, then Equation (127) becomes after introducing balloon term the following:

$$\begin{aligned} & \delta_\epsilon(\phi) \left\{ \mu \nabla \cdot \left(d(x, y) g(|\nabla z(x, y)|) \frac{\nabla \phi}{|\nabla \phi|} \right) \right. \\ & \quad - \left[\lambda_1 (z(x, y) - c_1)^2 - \lambda_2 (z(x, y) - c_2)^2 \right] \\ & \quad \left. - v \left[\left(\int_{\Omega} H dx dy - A_1 \right) - \left(\int_{\Omega} (1 - H) dx dy - A_2 \right) \right] \right\} \\ & - \alpha d(x, y) g(x, y) |\nabla \phi| = 0. \end{aligned} \quad (128)$$

Corresponding unsteady partial differential equation is solved by using additive operator splitting method which is discussed earlier; for reference see Badshah and Chen (2010) and Rada and Chen (2012, 2013). For experimental results of the model, see Rada and Chen (2013).

Reproducible Kernel Hilbert Space-Based Image Segmentation

One of the basic problems in image segmentation is to handle low contrast and missing edge information. This problem is addressed in many papers. One of that is given in Burrows et al. (2021), in which Burrows et al. proposed methods for segmentation of images having objects with low contrast by making weak edges more prominent. To make the unclear/weak edges more prominent, the authors used reproducible kernel Hilbert space (RKHS) and approximated Heaviside functions.

Deng et al. in (2016) used RKHS and approximated Heaviside functions for another type of imaging problem, namely, image super resolution. RKHS models the smooth parts of an image, while edges may be represented by a set of approximated Heaviside functions. For details about RKHS and approximated Heaviside functions, see Deng et al. (2016) and Burrows et al. (2021).

Global Segmentation Model

This is a two-stage model for segmentation of images with low contrast and noise. In the first stage, RKHS-based model is used to get clean approximation of the original noisy image, and then edge components are separated from the smooth components. In the second stage, a suitable segmentation model is used on the clean image. The following model is proposed for separating edge features and removing noise:

$$\min_{d, \beta} \frac{1}{2} \|z - (Kd + \Psi\beta)\|^2 + p_1 d^T Kd + p_2 \|\beta\|_1 + p_3 g^T |\nabla(Kd + \Psi\beta)|, \quad (129)$$

where Ψ collects values of the variation $\psi(v \cdot x + c)$ with v as the orientation at position c . ψ is the one-dimensional approximated Heaviside function:

$$\psi(t) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{t}{\delta}\right).$$

β is a vector of all weights used for computing the edge part of an image, which is modeled from the set of $\psi(t)$. K is a $\ell \times N$ matrix with $K_{j,k} = K(x_j, x_k)$; g is the edge detector function based on $\Psi\beta$, performing better than a gradient-based one. The final term encourages the contrast to be low in homogeneous regions and high near edges.

The model given in Eq. 129 is solved by introducing auxiliary variables say $\theta = \beta$, $W = Kd + \Psi\beta$, and $v = \nabla W$, to have the following scheme:

$$\begin{aligned} \min_{d, \beta, \theta, W, v} \frac{1}{2} \|z - (Kd + \Psi\beta)\|^2 + p_1 d^T Kd + p_2 \|\theta\|_1 + p_3 g^T |v| + \frac{\rho_1}{2} \|\theta \\ - \beta + b_1\|^2 + \frac{\rho_2}{2} \|W - (Kd + \Psi\beta) + b_2\|^2 + \frac{\rho_3}{2} \|v - \nabla W + b_3\|^2. \end{aligned} \quad (130)$$

To implement a block coordinate descent scheme, take the following initial approximations: $d^{(0)}$, $\beta^{(0)}$, $\theta^{(0)}$, $W^{(0)}$, $v^{(0)}$, and update them alternatively and iteratively as follows:

The d problem in proximal form:

$$\begin{aligned} d^{(k)} = \arg \min d \frac{1}{2} \|z - (Kd + \Psi\beta^{(k-1)})\|^2 + p_1 d^T Kd + \frac{\zeta_1}{2} \|d - d^{(k-1)}\|^2 \\ + \frac{\rho_2}{2} \|W^{(k-1)} - (Kd + \Psi\beta^{(k-1)}) + b_2^{(k-1)}\|^2, \end{aligned} \quad (131)$$

solution of this problem is obtained after some manipulation as follows:

$$d^{(k-1)} = A^{-1} \left(K^T z - (1 + \rho_2) K^T \Psi\beta^{(k-1)} + \zeta_1 d^{(k-1)} + \rho_2 K^T (W^{(k-1)} + b_2^{(k-1)}) \right), \quad (132)$$

Where:

$$A = (1 + \rho_2) K^T K + 2p_1 K + \zeta_1 I, \quad (I \text{ is the identity matrix}).$$

Linearizing β problem and solving give the following proximal linear form:

$$\beta^{(k)} = \arg \min_{\beta} \langle \hat{p}^{(k)}, \beta - \hat{\beta}^{(k-1)} \rangle + \frac{\rho}{2} \|\theta^{(k-1)} - \beta + b_1^{(k-1)}\|^2 + \frac{\zeta_2}{2} \|\beta - \hat{\beta}^{(k-1)}\|^2, \quad (133)$$

where $\hat{\beta}^{(k-1)} = \beta^{(k-1)} + \omega^{(k-1)}(\beta^{(k-1)} - \beta^{(k-2)})$ and $\hat{p}^{(k)} = \nabla f(\hat{\beta}^{(k-1)})$, with:

$$\begin{aligned} f(\hat{\beta}^{(k-1)}) = \frac{1}{2} \|z - (Kd^{(k)} + \Psi\hat{\beta}^{(k-1)})\|^2 + \mu g^T |v^{(k-1)}| \\ + \frac{\rho_2}{2} \|W^{(k-1)} - (Kd^{(k)} + \Psi\hat{\beta}^{(k-1)}) + b_2^{(k-1)}\|^2, \end{aligned} \quad (134)$$

$$\beta^{(k)} = \frac{1}{(\rho_1 + \zeta_2)} (\rho_1 (\theta^{(k-1)} + b_1^{(k-1)}) + \zeta_2 \hat{\beta}^{(k-1)} - \hat{p}^{(k)}). \quad (135)$$

Subproblems for θ , W and v , are given as follows:

$$\begin{aligned} \theta^{(k)} &= \arg \min_{\theta} \|\theta\|_1 + \frac{\rho_1}{2} \left\| \theta - \beta^{(k)} + b_1^{(k-1)} \right\|_2^2, \\ W^{(k)} &= \arg \min_W \frac{\rho_2}{2} \left\| W - \left(Kd^{(k)} + \Psi\beta^{(k)} \right) + b_2^{(k-1)} \right\|_2^2 \\ &\quad + \frac{\rho_3}{2} \left\| \mathbf{v}^{(k-1)} - \nabla W + \mathbf{b}_3^{(k-1)} \right\|_2^2, \\ \mathbf{v}^{(k)} &= \arg \min_{\mathbf{v}} \nu g^\top |\mathbf{v}| + \frac{\rho_3}{2} \left\| \mathbf{v} - \nabla W^{(k)} + \mathbf{b}_3^{(k-1)} \right\|_2^2 \end{aligned} \quad (136)$$

Corresponding solutions are given by the following:

$$\theta^{(k)} = \text{shrink} \left(\beta^{(k)} - b_1^{(k-1)}, \frac{\rho_1}{\rho_1} \right), \quad (137)$$

$$W^{(k)} = \mathfrak{H} \left[\mathcal{F}^* \left(\frac{\rho_3 \mathcal{F} \left(\nabla^* \left(\mathbf{v}^{(k-1)} + \mathbf{b}_3^{(k-1)} \right) \right) + \rho_2 \mathcal{F} \left(Kd^{(k)} + \Psi\beta^{(k)} - b_2^{(k)} \right)}{\rho_2 + \rho_3 \mathcal{F}(\nabla^2)} \right) \right] \quad (138)$$

$$\mathbf{v}^{(k)} = \text{shrink} \left(\nabla W^{(k)} - \mathbf{b}_3^{(k-1)}, \frac{\nu}{\rho_3} \cdot g \right) \quad (139)$$

Bregman parameters are updated as follows:

$$b_1^{(k)} = b_1^{(k-1)} + \theta^{(k)} - \beta^{(k)} \quad (140)$$

$$b_2^{(k)} = b_2^{(k-1)} + W^{(k)} - \left(Kd^{(k)} + \Psi\beta^{(k)} \right) \quad (141)$$

$$\mathbf{b}_3^{(k)} = \mathbf{b}_3^{(k-1)} + \mathbf{v}^{(k)} - \nabla W^{(k)}. \quad (142)$$

The first-stage model given in Eq. 129 gives us separation edges from the rest and gives us a clean image say $M = Kd + \Psi\beta$. This clean image M is used in the next stage as an input in the segmentation model (Chan et al. 2006) and is given by the following:

$$\begin{aligned}
 F(u) = & \int_{\Omega} g(|\Psi\beta|)|\nabla u|d\mathbf{x} + \lambda_1 \int_{\Omega} (M - c_1)^2 u d\mathbf{x} \\
 & + \lambda_2 \int_{\Omega} (M - c_2)^2 (1 - u)d\mathbf{x} + \xi \int_{\Omega} v(u)d\mathbf{x}
 \end{aligned}
 \tag{143}$$

Using similar framework, the authors proposed a combined model which combines RKHS model with convex CV model. As a result the following model is proposed:

$$\min_{d, \beta, 0 \leq u \leq 1, c_1, c_2} F(d, \beta, u, c_1, c_2)
 \tag{144}$$

where:

$$\begin{aligned}
 F(d, \beta, u, c_1, c_2) = & \frac{1}{2} \|z - (Kd + \Psi\beta)\|^2 + \gamma d^T Kd + \alpha \|\beta\|_1 + \mu g^T |\nabla u| \\
 & + \lambda [(Kd - \Psi\beta - c_1)^2 u + (Kd - \Psi\beta - c_2)^2 (1 - u)].
 \end{aligned}
 \tag{145}$$

To avoid non-differentiability of ℓ_1 norm, the following auxiliary variables are done before, $\theta = \beta$ and $w = (w_1, w_2) = \nabla u$. Thus the minimization problem becomes the following:

$$\min_{d, \beta, \theta, w, 0 \leq u \leq 1, c_1, c_2} F(d, \beta, \theta, w, u, c_1, c_2)
 \tag{146}$$

where

$$\begin{aligned}
 F(d, \beta, \theta, w, u, c_1, c_2) = & \frac{1}{2} \|z - (Kd + \Psi\beta)\|^2 + \gamma d^T Kd + \alpha \|\beta\|_1 + \mu g^T |w| \\
 & + \lambda [(Kd - \Psi\beta - c_1)^2 u + (Kd - \Psi\beta - c_2)^2 (1 - u)] \\
 & + \frac{\rho_1}{2} \|\theta - \beta + b_1\|_2^2 + \frac{\rho_2}{2} \|w - \nabla u + b_2\|_2^2.
 \end{aligned}
 \tag{147}$$

This equation leads to subproblems for $d, \beta, \theta, c_1, c_2, u, w$, for the solution BCD scheme is used as follows:

Subproblem 1.

$$\begin{aligned}
 d^{(k)} = & \arg \min_d \frac{1}{2} \|z - (Kd + \Psi\beta^{(k-1)})\|^2 + \gamma (d)^\top Kd + \frac{\xi_1}{2} \|d - d^{(k-1)}\|^2 \\
 & + \lambda \left[\left(u^{(k-1)} \right)^\top \left(Kd + \Psi\beta^{(k-1)} - c_1^{(k-1)} \right)^2 \right. \\
 & \left. + \left(1 - u^{(k-1)} \right)^\top \left(Kd + \Psi\beta^{(k-1)} - c_2^{(k-1)} \right)^2 \right].
 \end{aligned}
 \tag{148}$$

The solution is given by the following:

$$d^{(k)} = A^{-1} \left(K^\top z - (1 + 2\lambda) K^\top \Psi \beta^{(k-1)} + \zeta_1 d^{(k-1)} + 2\lambda K^\top \left[c_1^{(k-1)} u^{(k-1)} + c_2^{(k-1)} (1 - u^{(k-1)}) \right] \right) \quad (149)$$

where $A = (1 + 2\lambda) K^\top K + 2\gamma K + \zeta_1 I$.

Subproblem 2. To get optimal value of β , the following subproblem will be solved:

$$\beta^{(k)} = \arg \min_{\beta} \langle \hat{p}^{(k)}, \beta - \hat{\beta}^{(k-1)} \rangle + \rho_1 \text{over} 2 \|\theta^{(k-1)} - \beta + b_1^{(k-1)}\| + \frac{\zeta}{2} \|\beta - \hat{\beta}^{(k-1)}\|^2, \quad (150)$$

where $\hat{\beta}^{(k-1)} = \beta^{(k-1)} + \omega^{(k-1)} (\beta^{(k-1)} - \beta^{(k-2)})$ and $\hat{p}^{(k)} = \nabla f (\hat{\beta}^{(k-1)})$, where f is given by the following:

$$f (\hat{\beta}^{(k-1)}) = \frac{1}{2} \|z - (Kd^{(k)} + \Psi \hat{\beta}^{(k-1)})\|^2 + \mu g^\top |\mathbf{w}^{(k-1)}| + \lambda \left[\left(u^{(k-1)} \right)^\top \left(Kd^{(k)} + \Psi \hat{\beta}^{(k-1)} - c_1^{(k-1)} \right)^2 + \left(1 - u^{(k-1)} \right)^\top \left(Kd^{(k)} + \Psi \hat{\beta}^{(k-1)} - c_2^{(k-1)} \right)^2 \right] \quad (151)$$

$$\begin{aligned} \nabla f (\hat{\beta}^{(k-1)}) &= -\Psi^\top \left(z - (Kd^{(k)} + \Psi \hat{\beta}^{(k-1)}) \right) \\ &- 2\mu \Psi^\top \left(|\mathbf{w}^{(k-1)}| \odot \left(g (\Psi \hat{\beta}^{(k-1)}) \right) \odot (\Psi \hat{\beta}^{(k-1)}) \right) \\ &+ 2\lambda \Psi^\top \left[Kd^{(k)} + \Psi \hat{\beta}^{(k-1)} - c_1^{(k-1)} u^{(k-1)} - c_2^{(k-1)} (1 - u^{(k-1)}) \right], \end{aligned} \quad (152)$$

where \odot denotes the Hadamard product between vectors (component-wise multiplication). Thus the β update is given as follows:

$$\beta^{(k)} = \frac{1}{(\rho_1 + L_2)} \left(\rho_1 \left(\theta^{(k-1)} + b_1^{(k-1)} \right) + \zeta_2 \hat{\beta}^{(k-1)} - \hat{p}^{(k)} \right).$$

Subproblem 3. For the optimal solution of θ , the following minimization subproblem is solved:

$$\theta^{(k)} = \arg \min_{\theta} \alpha \|\theta\|_1 + \frac{\rho_1}{2} \|\theta - \beta^{(k)} + b_1^{(k-1)}\|^2, \quad (153)$$

whose solution is given by the following:

$$\theta^{(k)} = \text{shrink}(\beta^{(k)} - b_1^{(k-1)}, \frac{\alpha}{\rho_1}), \quad (154)$$

and the Bregman parameter is updated in the following way:

$$b_1^{(k)} = b_1^{(k-1)} + \theta^{(k)} - \beta^{(k)} \quad (155)$$

Subproblem 4. This subproblem is solved for finding c_1 and c_2 , for which the following minimization problem is solved:

$$c_1^{(k)} = \arg \min_{c_1} \lambda \left(u^{(k-1)} \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} - c_1 \right)^2 + \frac{\xi_3}{2} \left\| c_1 - c_1^{(k-1)} \right\|^2, \quad (156)$$

$$c_2^{(k)} = \arg \min_{c_2} \lambda \left(1 - u^{(k-1)} \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} - c_2 \right)^2 + \frac{\xi_4}{2} \left\| c_2 - c_2^{(k-1)} \right\|^2, \quad (157)$$

and the solutions are given by the following:

$$c_1^{(k)} = \frac{\xi_3 c_1^{(k-1)} + 2\lambda \left(u^{(k-1)} \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} \right)}{\xi_3 + 2\lambda \left(u^{(k-1)} \right)^\top I}, \quad (158)$$

$$c_2^{(k)} = \frac{\xi_4 c_2^{(k-1)} + 2\lambda \left(1 - u^{(k)} \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} \right)}{\xi_4 + 2\lambda \left(1 - u^{(k)} \right)^\top I}. \quad (159)$$

Subproblem 5. In this subproblem, the following minimization problem is solved for optimal value of u :

$$u^{(k)} = \arg \min_{u \in [0,1]} \frac{\rho_2}{2} \left\| \mathbf{w}^{(k-1)} - \nabla u + \mathbf{b}_2^{(k-1)} \right\|_2^2 + \frac{\xi_5}{2} \left\| u - u^{(k-1)} \right\|^2 \\ + \lambda \left[\left(u \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} - c_1^{(k)} \right)^2 \right. \\ \left. + \left(1 - u \right)^\top \left(Kd^{(k)} + \Psi\beta^{(k)} - c_2^{(k)} \right)^2 \right]. \quad (160)$$

The solution to this is given by the following:

$$u^{(k)} = \Re \left[\mathcal{F}^* \left(\frac{\left(\rho_2 \mathcal{F} \left(\nabla^* \left(\mathbf{w}^{(k-1)} + \mathbf{b}_2^{(k-1)} \right) \right) - \lambda \mathcal{F} \left(r^{(k)} \right) + \xi_5 \mathcal{F} \left(u^{(k-1)} \right) \right)}{\xi_5 + \rho_2 \mathcal{F} \left(\nabla^2 \right)} \right) \right], \quad (161)$$

where $r^{(k)} = \left(Kd^{(k)} + \Psi\beta^{(k)} - c_1^{(k)}\right)^2 - \left(Kd^{(k)} + \Psi\beta^{(k)} - c_2^{(k)}\right)^2$ and \mathcal{F} is the fast Fourier transform operator and \mathcal{F}^* is its inverse.

Subproblem 6. In this subproblem, the following minimization problem is solved to update w :

$$w^{(k)} = \arg \min_w \mu g^\top |w| + \frac{\rho_2}{2} \|w - \nabla u^{(k)} + b_2^{(k-1)}\|_2^2. \tag{162}$$

Solving this minimization problem leads to the following solution:

$$w^{(k)} = \text{shrink}(\nabla u^{(k)} - b_2^{(k-1)}, \frac{\mu}{\rho_2} \cdot g), \tag{163}$$

the Bregman parameter is updated as follows:

$$b_2^{(k)} = b_2^{(k-1)} + w^{(k)} - \nabla u^{(k)}. \tag{164}$$

For experimental results, comparison, and extension of the model to selective segmentation, the readers are advised to see Burrows et al. (2021).

An Optimization-Based Multilevel Algorithm for Selective Image Segmentation Models

In 2017, Jumaat and Chen proposed a multilevel method for solution of Badshah-Chen selective segmentation model discussed in section “Active Contour-Based Image Selective Model” and Rada-Chen selective segmentation discussed in section “One-Level Selective Segmentation Model”.

Multilevel Algorithm for Badshah-Chen (BC) Model

Consider energy functional of the Badshah-Chen model given in Equation (110):

$$\begin{aligned} \min_{\phi(x,y), c_1, c_2} F_\epsilon(\phi(x, y), c_1, c_2) &= \mu \int_{\Omega} G(x, y) |\nabla H_\epsilon(\phi(x, y))| dx dy \\ &+ \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_\epsilon(\phi(x, y)) dx dy \\ &+ \lambda_2 \int_{\Omega} |z(x, y) - c_2|^2 (1 - H_\epsilon(\phi(x, y))) dx dy. \end{aligned}$$

where $G(x, y) = d(x, y)g(|\nabla z(x, y)|)$ and $|\nabla H_\epsilon(\phi(x, y))| = \delta_\epsilon(\phi)|\nabla\phi(x, y)|$. Suppose that the average intensities c_1 and c_2 are found at the start by using (13), and to update ϕ , the following minimization problem will be considered:

$$\begin{aligned}
 \min_{\phi(x,y)} F_\epsilon(\phi(x, y)) &= \mu \int_{\Omega} G(x, y)\delta_\epsilon(\phi)|\nabla\phi(x, y)|dxdy \\
 &+ \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_\epsilon(\phi(x, y))dxdy \quad (165) \\
 &+ \lambda_2 \int_{\Omega} |z(x, y) - c_2|^2(1 - H_\epsilon(\phi(x, y)))dxdy.
 \end{aligned}$$

Here assume that given image $z(x, y)$ has size $n \times n$ where $n = 2^L$. The standard coarsening defines $L + 1$ levels where $k = 1$ (finest level), $2, \dots, L, L + 1$ (coarsest level); furthermore, k -th level has $\tau_k \times \tau_k$ “superpixels,” and each “superpixel” has $b_k \times b_k$ pixels where $\tau_k = \frac{n}{b_k}$ and $b_k = 2^{k-1}$. By using discrete form of TV $|\nabla\phi|$, Equation (165) can be written as follows:

$$\begin{aligned}
 &\min_{\{\phi_{i,j}\}'s} F(\phi_{1,1}, \phi_{2,1}, \dots, \phi_{m_1-1,m_2}, \phi_{m_1,m_2}) \\
 &= \mu \sum_{i=1}^{m_1-1} \sum_{j=1}^{m_2-1} G_{i,j} \sqrt{\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{h}\right)^2 + \left(\frac{\phi_{i,j+1} - \phi_{i,j}}{h}\right)^2} \cdot \delta_\epsilon(\phi_{i,j})h^2 \\
 &+ \sum_{i=1}^{m_1-1} \sum_{j=1}^{m_2-1} [\lambda_1(z_{i,j} - c_1)^2 H_\epsilon(\phi_{i,j}) + \lambda_2(z_{i,j} - c_2)^2(1 - H_\epsilon(\phi_{i,j}))].h^2. \\
 &= \underline{\mu} \sum_{i=1}^{m_1-1} \sum_{j=1}^{m_2-1} G_{i,j} \sqrt{\left(\phi_{i+1,j} - \phi_{i,j}\right)^2 + \left(\phi_{i,j+1} - \phi_{i,j}\right)^2} \cdot \delta_\epsilon(\phi_{i,j}) \quad (166) \\
 &+ \sum_{i=1}^{m_1-1} \sum_{j=1}^{m_2-1} \underbrace{[\lambda_2(z_{i,j} - c_1)^2 - \lambda_2(z_{i,j} - c_2)^2]}_{r(x,y)} H_\epsilon(\phi_{i,j}) + \text{terms independent of } \phi,
 \end{aligned}$$

where $\underline{\mu} = \mu/h$ and the minimization is done with respect to ϕ , so the last term will not be considered from here onward. Consider fine-level local minimization first, which is done by using coordinate descent method.

The Finest-Level Local Minimization ($k = 1$)

Let $\tilde{\phi}$ be the current iterate. Then our idea is to solve a series of subproblems of the following form:

$$\min_C F_\epsilon(\tilde{\phi} + C)$$

where C is a local and piecewise constant function. Consider a particular pixel (i, j) . Clearly if only $\phi_{i,j}$ is allowed to vary, we simply consider the local subproblem:

$$\begin{aligned} \min_{\phi_{i,j}} F^{\text{local}}(\phi_{i,j}) = & \underline{\mu} \left[G_{ij} \sqrt{(\phi_{ij} - \tilde{\phi}_{i+1,j})^2 + (\phi_{ij} - \tilde{\phi}_{i,j+1})^2} \delta_\epsilon(\phi_{i,j}) \right. \\ & + G_{i-1,j} \sqrt{(\phi_{ij} - \tilde{\phi}_{i-1,j})^2 + (\tilde{\phi}_{i-1,j} - \tilde{\phi}_{i-1,j+1})^2} \delta_\epsilon(\tilde{\phi}_{i-1,j}) \\ & + G_{i,j-1} \sqrt{(\phi_{ij} - \tilde{\phi}_{i,j-1})^2 + (\tilde{\phi}_{i,j-1} - \tilde{\phi}_{i+1,j-1})^2} \delta_\epsilon(\tilde{\phi}_{i,j-1}) \left. \right] \\ & + r_{ij} H(\tilde{\phi}_{ij}), \end{aligned}$$

where $r_{i,j} = \lambda_1(z_{i,j} - c_1)^2 - \lambda_2(z_{i,j} - c_2)^2$. Starting from $\phi_{i,j}^{\text{old}} = \tilde{\phi}_{i,j}$, we can iterate the following (Richardson type) scheme to obtain an approximation for $\phi_{i,j}$:

$$\phi_{i,j}^{\text{new}} = RHS/LHS, \quad (167)$$

where:

$$\begin{aligned} RHS = & \underline{\mu} \left[G_{ij} \frac{(\tilde{\phi}_{i+1,j} + \tilde{\phi}_{i,j+1})}{L_1} \delta_\epsilon(\phi_{i,j}^{\text{old}}) + G_{i-1,j} \frac{\tilde{\phi}_{i-1,j} \cdot \delta_\epsilon(\tilde{\phi}_{i-1,j})}{L_2} \right. \\ & \left. + G_{i,j-1} \frac{\tilde{\phi}_{i,j-1} \cdot \delta_\epsilon(\tilde{\phi}_{i,j-1})}{L_3} \right] + r_{i,j} \delta_\epsilon(\tilde{\phi}_{i,j}), \end{aligned}$$

$$LHS = \underline{\mu} \left[\frac{2\delta_\epsilon(\phi_{i,j}^{\text{old}})}{L_1} + \frac{2\epsilon L_1}{\pi(\epsilon^2 + \phi_{i,j}^{\text{old}2})^2} + \frac{\delta_\epsilon(\tilde{\phi}_{i-1,j})}{L_2} + \frac{\delta_\epsilon(\tilde{\phi}_{i,j-1})}{L_3} \right]$$

and

$$L_1 = \sqrt{(\phi_{ij}^{\text{old}} - \tilde{\phi}_{i+1,j})^2 + (\phi_{ij}^{\text{old}} - \tilde{\phi}_{i,j+1})^2} + \beta$$

$$L_2 = \sqrt{(\phi_{ij}^{\text{old}} - \tilde{\phi}_{i-1,j})^2 + (\tilde{\phi}_{i-1,j} - \tilde{\phi}_{i-1,j+1})^2} + \beta$$

$$L_3 = \sqrt{(\phi_{ij}^{\text{old}} - \tilde{\phi}_{i,j-1})^2 + (\tilde{\phi}_{i,j-1} - \tilde{\phi}_{i+1,j-1})^2} + \beta,$$

and $\gamma > 0$ is a regularizing parameter. Equation (167) is usually done for few steps only to update $\tilde{\phi}_{i,j}$.

The General-Level k Local Minimization ($1 < k \leq L$)

On a general-level k , consider the following minimization subproblem:

$$\min_C F(\tilde{\phi} + C), \quad (168)$$

where C is a local and piecewise constant function of support $\tau_k \times \tau_k = 2^{k-1} \times 2^{k-1}$ at each block (i, j) of pixels. On k th level, the subproblem may be taken as follows:

$$\hat{c} = \arg \min_{c \in \mathbb{R}^{\tau_k \times \tau_k}} F(\tilde{\phi} + I_k B_k c), \quad C_k = I_k B_k \hat{c}, \quad (169)$$

where $B_k : \mathbb{R} \rightarrow \mathbb{R}^{\tau_k \times \tau_k}$ duplicates a constant to a block of constants and $I_k : \mathbb{R}^{\tau_k \times \tau_k} \rightarrow \mathbb{R}^{n \times n}$ is the interpolation operator so $C_k \in \mathbb{R}^{n \times n}$. Here we may illustrate $C_k = I_k B_k \hat{c}$ as follows (Chan and Chen 2006):

$$C_k = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & c & \dots & c & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & c & \dots & c & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \text{ to approximate } \begin{bmatrix} c_{11} & \dots & \dots & \dots & c_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ c_{i1} & \dots & c_{ii} & \dots & c_{ij} & \dots & c_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{j1} & \dots & c_{ji} & \dots & c_{jj} & \dots & c_{jn} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{n1} & \dots & \dots & \dots & \dots & \dots & c_{nn} \end{bmatrix}.$$

Details of solving the local minimization subproblem (169) are here. Set on level k , $b = \tau_k = 2^{k-1}$, $k_1 = (i - 1)b + 1$, $k_2 = ib$, $\ell_1 = (j - 1)b + 1$, $\ell_2 = jb$. Firstly, note that on level k , there are only $m_1/\tau_k \times m_2/\tau_k$ subproblems each of which is essentially one dimensional (mimicking a coarse grid of a geometric multigrid method). Secondly, introduce the Richardson-type iterative method adopted for each subproblem.

At each block (i, j) of pixels, solve (169) for $c_{i,j}$. Observe that each TV term $|\nabla\phi|$ does not change within the interior pixels of each block on level k because of the following:

$$\begin{aligned} & \sqrt{[(c_{i,j} + \tilde{\phi}_{k,\ell}) - (c_{i,j} + \tilde{\phi}_{k+1,\ell})]^2 + [(c_{i,j} + \tilde{\phi}_{k,\ell}) - (c_{i,j} + \tilde{\phi}_{k,\ell+1})]^2} \\ & = \sqrt{[\tilde{\phi}_{k,\ell} - \tilde{\phi}_{k+1,\ell}]^2 + [\tilde{\phi}_{k,\ell} - \tilde{\phi}_{k,\ell+1}]^2} \equiv T_{k,\ell}. \end{aligned}$$

So it remains to consider the contribution to the TV term stemming from the boundary pixels (of the block) and the contribution of all interior pixels to the δ_ϵ term. Thus solving (169) is equivalent to solving the following (i, j) block local minimization problem:

$$\begin{aligned} & \min_{c_{i,j}} F(\tilde{\phi}_{i,j} + I_k B_k c_{i,j}) \\ & = \underline{\mu} \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \sqrt{[c_{i,j} - (\tilde{\phi}_{k_1-1,\ell} - \tilde{\phi}_{k_1,\ell})]^2 + [\tilde{\phi}_{k_1-1,\ell} - \tilde{\phi}_{k_1-1,\ell+1}]^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_1-1,\ell}) \\ & \quad + \underline{\mu} \sum_{k=k_1}^{k_2-1} G_{k,\ell_2} \sqrt{[c_{i,j} - (\tilde{\phi}_{k,\ell_2+1} - \tilde{\phi}_{k,\ell_2})]^2 + [\tilde{\phi}_{k,\ell_2} - \tilde{\phi}_{k+1,\ell_2}]^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k,\ell_2}) \\ & \quad + \underline{\mu} G_{k_2,\ell_2} \sqrt{[c_{i,j} - (\tilde{\phi}_{k_2,\ell_2+1} - \tilde{\phi}_{k_2,\ell_2})]^2 + [c_{i,j} - (\tilde{\phi}_{k_2+1,\ell_2} - \tilde{\phi}_{k_2,\ell_2})]^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_2,\ell_2}) \end{aligned}$$

$$\begin{aligned}
& + \underline{\mu} G_{k_2, \ell} \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{[c_{i,j} - (\tilde{\phi}_{k_2+1, \ell} - \tilde{\phi}_{k_2, \ell})]^2 + [\tilde{\phi}_{k_2, \ell} - \tilde{\phi}_{k_2, \ell+1}]^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_2, \ell}) \\
& + \underline{\mu} G_{k, \ell_1} \sum_{k=k_1}^{k_2} \sqrt{[c_{i,j} - (\tilde{\phi}_{k, \ell_1-1} - \tilde{\phi}_{k, \ell_1})]^2 + [\tilde{\phi}_{k, \ell_1-1} - \tilde{\phi}_{k+1, \ell_1-1}]^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell_1}) \\
& + \sum_{k=k_1+1}^{k_2-1} \sum_{\ell=\ell_1+1}^{\ell_2-1} T_{k, \ell} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell}) + \sum_{\ell=\ell_1}^{\ell_2} \sum_{k=k_1}^{k_2} r(k, \ell) H_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell}). \tag{170}
\end{aligned}$$

To simplify the formulae, let:

$$\Phi_{k, \ell} = \tilde{\phi}_{k, \ell+1} - \tilde{\phi}_{k, \ell}, \quad \Theta_{k, \ell} = \tilde{\phi}_{k+1, \ell} - \tilde{\phi}_{k, \ell},$$

and:

$$P_{k, \ell} = \frac{\Phi_{k, \ell} + \Theta_{k, \ell}}{2}, \quad Q_{k, \ell} = \frac{\Phi_{k, \ell} - \Theta_{k, \ell}}{2}.$$

Using the identity:

$$\sqrt{(c-a)^2 + (c-b)^2} = \sqrt{2} \sqrt{\left(c - \frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2},$$

we may rewrite (170) as the following problem:

$$\begin{aligned}
\mathcal{F}(c_{i,j}) &= \underline{\mu} G_{k_1-1, \ell} \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{i,j} - \Theta_{k_1-1, \ell})^2 + \Phi_{k_1-1, \ell}^2} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_1-1, \ell}) \\
& + \underline{\mu} G_{l, \ell_2} \sum_{k=k_1}^{k_2-1} \sqrt{(c_{i,j} - \Phi_{k, \ell_2})^2 + \Theta_{k, \ell_2}^2} \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell_2}) \\
& + \underline{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2, \ell} \sqrt{(c_{i,j} - \Theta_{k_2, \ell})^2 + \Phi_{k_2, \ell}^2} \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_2, \ell}) \\
& + \underline{\mu} \sum_{k=k_1}^{k_2} G_{k, \ell_1} \sqrt{(c_{i,j} - \Phi_{k, \ell_1})^2 + \Theta_{k, \ell_1}^2} \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell_1}) \\
& + \underline{\mu} G_{k_2, \ell_2} \sqrt{2} \sqrt{(c_{i,j} - P_{k_2, \ell_2})^2 + (Q_{k_2, \ell_2})^2} \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k_2, \ell_2}) \\
& + \underline{\mu} \sum_{k=k_1+1}^{k_2-1} \sum_{\ell=\ell_1+1}^{\ell_2-1} T_{k, \ell} \cdot \delta_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell}) + \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} r_{k, \ell} H_\epsilon(c_{i,j} + \tilde{\phi}_{k, \ell}).
\end{aligned}$$

The first-order condition for $\mathcal{F}(c_{i,j}) = 0$ and doing some manipulations, the following iterative scheme for c_{ij} will be achieved:

$$c_{i,j}^{\text{new}} = RHS^{\text{old}} / LHS^{\text{old}}, \tag{171}$$

starting from $c_{i,j}^{\text{old}} = 0$:

$$\begin{aligned} RHS^{\text{old}} = & 2\mu \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \frac{\tilde{\phi}_{k_1-1,\ell} \sqrt{(c_{i,j}^{\text{old}} - \Theta_{k_1-1,\ell})^2 + \Phi_{k_1-1,\ell}^2}}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k_1-1,\ell})^2)^2} \\ & + \mu \sum_{\ell=\ell_1}^{\ell_2} \frac{\Theta_{k_1-1,\ell}}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k_1-1,\ell})^2)^2 \sqrt{(c_{i,j}^{\text{old}} - \Theta_{k_1-1,\ell})^2 + \Phi_{k_1-1,\ell}^2}} \\ & + \dots + 2\mu \sum_{k=k_1+1}^{k_2-1} \sum_{\ell=\ell_1+1}^{\ell_2-1} T_{k,\ell} \cdot \frac{\tilde{\phi}_{k,\ell}}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k,\ell})^2)^2} \\ & - \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} r_{k,\ell} \cdot \left[\frac{2c_{i,j}^{\text{old}} \tilde{\phi}_{k,\ell}}{(\epsilon^2 + \phi_{k,\ell}^2)^2} + \frac{1}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k,\ell})^2)^2} \right], \end{aligned}$$

and:

$$\begin{aligned} LHS^{\text{old}} = & -2\mu \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \frac{\sqrt{(c_{i,j}^{\text{old}} - \Theta_{k_1-1,\ell})^2 + \Phi_{k_1-1,\ell}^2}}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k_1-1,\ell})^2)^2} \\ & + \mu \sum_{\ell=\ell_1}^{\ell_2} \frac{1}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k_1-1,\ell})^2)^2 \sqrt{(c_{i,j}^{\text{old}} - \Theta_{k_1-1,\ell})^2 + \Phi_{k_1-1,\ell}^2}} \\ & + \dots - 2\mu \sum_{k=k_1+1}^{k_2-1} \sum_{\ell=\ell_1+1}^{\ell_2-1} \frac{T_{k,\ell}}{(\epsilon^2 + (c_{i,j}^{\text{old}} + \tilde{\phi}_{k,\ell})^2)^2} \\ & - 2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} r_{k,\ell} \frac{\tilde{\phi}_{k,\ell}}{(\epsilon^2 + \tilde{\phi}_{k,\ell}^2)^2}. \end{aligned}$$

Once $c_{i,j}$ is obtained, $\tilde{\phi}_{k,\ell}$ is updated as follows:

$$\phi_{k,l} = \tilde{\phi}_{k,\ell} + c_{i,j}$$

to the full (i, j) block.

The Coarsest-Level Minimization ($k = L + 1$)

On the coarsest level, the whole image is considered to be a single block, so contribution for updating the constant will only come from the delta function term $\delta_\epsilon(\phi)$, i.e., no contribution from the TV term. Thus consider the following local minimization problem on the coarsest level:

$$\min_c F(\tilde{\phi} + I_k B_k c) = \min_c \mu \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} T_{i,j} \delta_\epsilon(\tilde{\phi}_{i,j} + c) + \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} r_{i,j} H_\epsilon(\tilde{\phi}_{i,j} + c).$$

Taking variation with respect to c and equating to 0 leads to the following:

$$\begin{aligned} & - \mu \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} G_{ij} G_{i,j} T_{i,j} \frac{\tilde{\phi}_{i,j} + c^{\text{new}}}{(\epsilon^2 + (\tilde{\phi}_{i,j} + c)^2)^2} \\ & + \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} r_{i,j} \left[\frac{2c^{\text{old}} \tilde{\phi}_{i,j}}{(\epsilon^2 + \tilde{\phi}_{i,j}^2)^2} + \frac{1}{(\epsilon^2 + (c^{\text{old}} + \tilde{\phi}_{i,j})^2)} \right. \\ & \left. - \frac{2c^{\text{new}} \tilde{\phi}_{i,j}}{(\epsilon^2 + \tilde{\phi}_{i,j}^2)^2} \right] = 0. \end{aligned} \quad (172)$$

Linearizing and solving this equation for c^{new} and then updating $\tilde{\phi}$ will be similarly done as above.

Here is the algorithm for multilevel method for BC model:

Algorithm 6 2D multilevel algorithm (ML1)

$[\phi, c_1, c_2] \leftarrow \text{Opt Multilevel1}(\phi, z)$ Given the image z and an initial guess $\phi = \tilde{\phi}$ with $L + 1$ levels, our multilevel algorithm proceeds as follows:

Start

set $\phi_0 = \tilde{\phi}$ and compute c_1, c_2 .

for level $k = 1, 2, \dots, L + 1$.

If $k = 1$, for finest level

solve (167).

Elseif $k = L + 1$ i.e. on coarsest level.

solve (172) to find c

Else on all other levels

solve (171).

Update $\phi = \tilde{\phi} + I_k B_k c$.

end

Go to Start with $\tilde{\phi} = \phi$ unless $\|\phi - \phi_0\| < \text{tol}$.

Exactly in same lines, multilevel method for RC model can be derived, and it is left as an exercise for the reader. For comparison and experimental results, see Jumaat and Chen (2017).

Machine/Deep Learning Techniques for Image Segmentation

In this section, a survey of deep/machine learning techniques for image segmentation is given.

Machine Learning with Region-Based Active Contour Models in Medical Image Segmentation

In 2017, Pratando et al. proposed an architecture which integrates machine learning with a region-based active contour model.

Proposed Framework

The proposed framework can be constructed from any algorithm used for classification, which is combined with a region-based model with a level set method. The matrix of classifier probability scores is generated by using KNN and support vector machine (SVM). The matrix is then regularized and combined with Chan-Vese (CV) active contour model (Chan and Vese 2001) which is discussed in section “[Chan-Vese Model](#)” in detail.

Classifier Probability Scores

For a given image z , a matrix of classifier probability scores is generated from the classification algorithms. Here two classification algorithms, namely, KNN and SVM, are investigated.

KNN. KNN provides scores in the range $[0, 1]$ which can be implemented easily using the fuzzy KNN rule. This rule is derived from the fuzzy set and the KNN classifier in machine learning. For a reference set $X_R = \{x_i : 1 \leq i \leq m_R\}$ and a set of l -dimensional vectors $W = \{w_i : 1 \leq i \leq m_R\}$, $w_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l})$, l and m_R are the number of classes and the number of elements in the reference set X_R , respectively. Due to fuzziness of the vectors, the following condition must be satisfied:

$$\sum_{j=1}^l w_{i,j} = 1, \quad 0 \leq w_{i,j} \leq 1. \quad (173)$$

The value of $w_{i,j}$ for $1 \leq i \leq m_R$ and $1 \leq j \leq l$ is the membership value of the i -th object to class j . For a particular x to be classified, the set K of indices corresponding to the classes of k -nearest neighbors of x in X_R is obtained. The fuzzy decision vector v in the fuzzy KNN is computed in the following way:

$$v = \frac{1}{k} \sum_{s \in K} w_s. \quad (174)$$

The maximum v_j , $1 \leq j \leq l$ where $v = (v_1, v_2, \dots, v_l)$ is used to define the object class in the original KNN.

Support vector machine SVM. In support vector machine, the given data is divided into two classes by finding a hyperplane between the classes with largest margin. This is done by using a sign function $class(x) = sgn(h(x))$, where $h(x)$ is the separating hyperplane for the two classes and is given by the following:

$$h(x) = \mathbf{w}_0^T \mathbf{x} + b_0 \quad (175)$$

where \mathbf{w}_0 is a d -dimensional optimal weight vector, \mathbf{x} is the given data, and b_0 is the optimal bias. Since it may be difficult to separate the data in the original input space, a transformation of the data into a higher dimensional space through function φ is introduced. The $h(x)$ takes the following form:

$$h(x) = \mathbf{w}_0^T \varphi(\mathbf{x}) + b_0. \quad (176)$$

It is still hard to find φ explicitly, so a kernel $K(\mathbf{x}, \mathbf{x}_i)$ is introduced and thus (176) may be written as follows:

$$h(x) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b_0, \quad (177)$$

where α_i is the estimated SVM parameter and $y_i \in \{1, -1\}$ is the desired class for the corresponding \mathbf{x}_i . The value of $h(x)$ is the SVM evaluation score and the sign is the predicted class. Note that the scores of KNN falls in the range $[0, 1]$ and that of SVM in the range $(-\infty, \infty)$, which can be converted to a prior probability score.

Regularization for Classifier Probability Score

Classifiers generate binary results by applying a hard limiter function to the probability scores. Let $s \in [0, 1]$ be a probability score and ρ be a regularization function that maps s to a real value in $[0, 1]$. The traditional classifiers generate binary results by the following:

$$\rho_1(s) = \begin{cases} 1 & \text{if } s \geq \frac{1}{2} \\ 0 & \text{if } s < \frac{1}{2} \end{cases} \quad (178)$$

Instead of refining these binary scores using machine learning algorithms. To retain the probability scores which are processed further by applying any region-based active contour model. This aims to find an optimal solution where the function $\rho(s)$ can be simply expressed by the following:

$$\rho_2(s) = s. \quad (179)$$

A nonlinear function ρ approximately lying under ρ_2 for $s > 0.5$ and above ρ_2 for $s < 0.5$ leads to better results. The regularization function in general should satisfy the following conditions:

1. The domain and the range, ρ , should be $[0, 1]$.
2. It should be increasing.
3. The following equations must hold:

$$\lim_{s \rightarrow 0} \rho(s) = 0 \quad (180)$$

$$\lim_{s \rightarrow 0.5} \rho(s) = 0.5 \quad (181)$$

$$\lim_{s \rightarrow 1} \rho(s) = 1. \quad (182)$$

4. It should be close to 0.5 when s is in the vicinity of 0.5.

There are some more options for taking regularization functions $\rho(s)$; for details see Pratondo et al. (2017).

The map of ρ is then fed to a region-based active contour model. Through energy minimization using the level set method, the optimum solution for the desired region can be obtained. For experimental results, data set utilization, and comparisons, see Pratondo et al. (2017).

ResBCU-Net: Deep Learning Approach for Segmentation of Skin Images

In 2022, Badshah and Ahmad proposed a new architecture based on CNNs, namely, ResBCU-Net for segmentation of skin images/medical images. The network, ResBCU-Net, is an extension of the U-Net which utilizes residual blocks, batch normalization, and bidirectional ConvLSTM. In addition, we present an extended form of ResBCU-Net, ResBCU-Net(d=3), which takes advantage of densely connected layers in its bottleneck section.

Proposed Work

Based on U-Net (Olaf et al. 2015) and inspired by residual blocks (He et al. 2016), batch normalization (Ioffe and Szegedy 2015), and bidirectional convolutional long-short-term memory (BConvLSTM) network (Song et al. 2018), a neural network, named as ResBCU-Net, shown in the Fig. 4 was proposed for segmentation of skin/medical images. The authors have made changes in the encoding path and decoding path of the classical U-Net, which is explained here in detail by considering encoding and decoding separately.

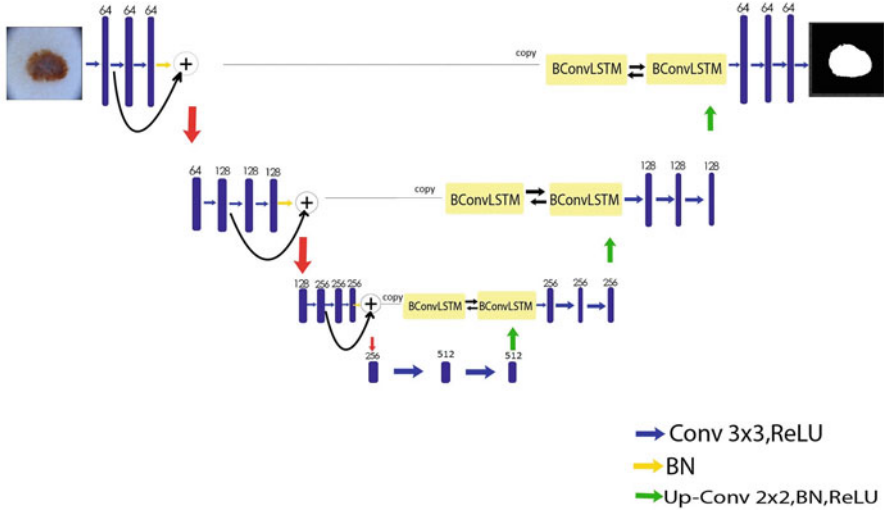


Fig. 4 ResBCU-Net architecture with residual blocks in the encoding path and BCConvLSTM in the decoding path. The numbers on top of the rectangles show number of channels

Encoding

Unlike to the U-Net (Olaf et al. 2015), encoding/contracting path of ResBCU-Net consists of residual blocks (He et al. 2016) and batch normalization layers (Ioffe and Szegedy 2015) with nine convolution layers. The path consists of three blocks; each block contains three convolution layer followed by a batch normalization layer. The output of first convolution layer in each block is added with the output of the batch normalization layer, which is then followed by a max pooling layer. At the same time, before the max pooling layer, the output of each block is passed for concatenation with the corresponding output of the decoding/expanding path.

Residual Blocks

Successive sequences of convolution layers lead to learning of different features; in some cases it may also lead to learning of redundant features; and adding more layers lead to higher training error. To solve this problem in such deeper models, residual blocks are introduced in He et al. (2016). An input to some convolution layers is added to the output of the layers; the resultant is again fed to the successive convolution layers; an example of residual block is shown in the Fig. 5.

The authors utilized this approach for ResBCU-Net encoding path. Instead of blocks of two convolution layers in the encoding path, three convolutions blocks each followed by a batch normalization layer are introduced. Each block is then converted to residual blocks by adding the output of the first convolution layer to the output of the batch normalization layer in the block, as shown in the Fig. 6.

Fig. 5 ResNet residual block

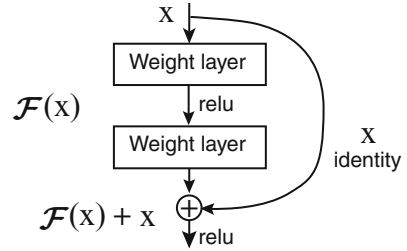
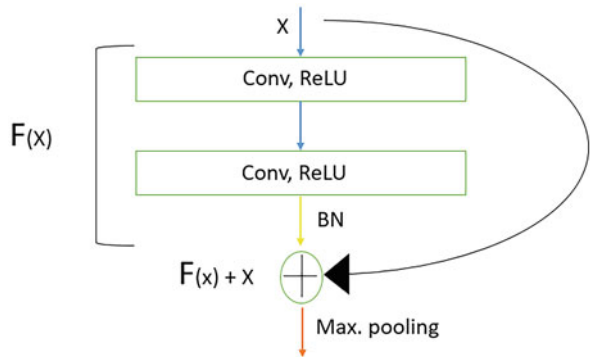


Fig. 6 ResBCU-Net residual block



Batch Normalization

To avoid over-fitting and for acceleration of the training process, batch normalization layers (Ioffe and Szegedy 2015) in the encoding and decoding path of ResBCU-Net are included. The batch normalization layer controls variation in distribution by calculating mean and standard deviation values of the data set as a whole by adjusting the mean to 0 and variance to 1; the equation for batch normalization (BN) is given below:

$$BN = \gamma_c \left[\frac{I_{n,c,h,w} - \mu_c}{\sqrt{\sigma_c^2 - \epsilon}} \right] + \beta_c$$

where $I_{n,c,h,w}$ represents n-number of images provided to a neural network at a time with c channels, h heights, and w widths. μ_c and σ_c^2 are channel-wise global mean and variance of the images, respectively. β_c and γ_c are learnable mean and standard deviation, respectively, while ϵ is kept constant as 0.00001.

Batch normalization layers in each block of the encoding and decoding path are introduced. In the encoding path, BN layers are used at the end of each block just before max pooling layer. After max pooling layer of the third block of convolution layers, bottleneck section of the network starts, where only two convolution layers

each followed by an activation function, ReLU, are used. While in the decoding path, the batch normalization layers after each up-sampling layer are used, which are then followed by activation functions, ReLU, before proceeding to the next block.

Decoding

The decoding/expanding path of ResBCU-Net, inspired by BCDU-Net (Guo et al. 2019), contains convolution layers, up-sampling layers, batch normalization layers (Ioffe and Szegedy 2015), and bidirectional LSTM convolutions (BConvLSTM) (Song et al. 2018). Right after the bottleneck portion of the network, an up-sampling convolution with 2×2 filter, followed by a batch normalization layer, is used which is then followed by two convolution layer blocks. Features from the corresponding blocks in the encoding path are passed into the BConvLSTM after concatenation with the outputs of the corresponding block of the decoding path. In each block, outputs of BConvLSTMs are passed into two convolutional layers. At the end of the decoding path, we use a convolution layer with 1×1 filter followed by a sigmoid function as an activation function.

Bidirectional Long-Short-Term Memory Convolutions (BConvLSTM)

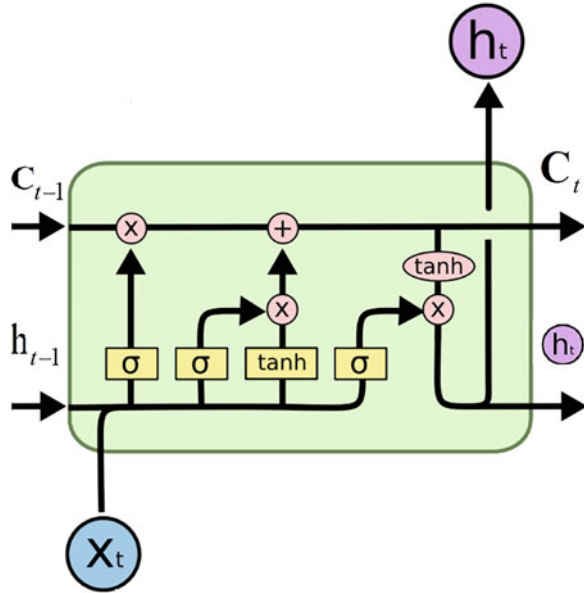
In the decoding path, the convolutional long-short-term memory (ConvLSTM) networks for ResBCU-Net are inspired by Azad et al. (2019) and Guo et al. (2019). The ConvLSTM to process features into two directions is used: forward and backward, known as BConvLSTM (Song et al. 2018). BConvLSTM has been implemented successively to enhance performance of neural networks (Cui et al. 1801; Guo et al. 2019). LSTMs are enhanced version of recurrent neural networks (RNNs) (Jordan 1990; Cleeremans et al. 1989; Pearlmutter 1989), which have been developed to overcome the gradient vanishing issue in long dependence of neural network in training (Fig. 7).

A single block of ConvLSTM consists of input gate, i_t ; forget gate, f_t ; and output gate, O_t . If $\chi_1, \chi_2, \dots, \chi_t$ are inputs, C_1, C_2, \dots, C_t are cell-state outputs, and h_1, h_2, \dots, h_t represent hidden states; then the function of a single ConvLSTM can be represented by the following equations:

$$\begin{aligned} i_t &= \sigma(\omega_{xi} * \chi_t + \omega_{hi} * h_{t-1} + \omega_{ci} o C_{t-1} + b_i) \\ f_t &= \sigma(\omega_{xf} * \chi_t + \omega_{hf} * h_{t-1} + \omega_{cf} o C_{t-1} + b_f) \\ C_t &= f_t o C_{t-1} + i_t o \tanh(\omega_{xc} * \chi_t + \omega_{hc} * h_{t-1} + b_c) \\ O_t &= \sigma(\omega_{xo} * \chi_t + \omega_{ho} * h_{t-1} + \omega_{co} o C_t + b_o) \\ h_t &= O_t o \tanh(C_t). \end{aligned}$$

Here, $*$ and o are convolution operator and Hadamard product, respectively. The h_t is the hidden state, output, of the single ConvLSTM block. Now, in case of bidirectional ConvLSTM (BConvLSTM), the output can be represented as follows:

Fig. 7 A single block of ConvLSTM network (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)



$$Y_t = \tanh(\omega_y^{\vec{h}} * \vec{h}_t + \omega_y^{\overleftarrow{h}} * \overleftarrow{h}_t + b).$$

Here, \vec{h}_t and \overleftarrow{h}_t are output states of forward and backward direction feature process; and the Y_t is the final output of a BConvLSTM block.

The copied features from encoding path are concatenated with corresponding outputs from decoding path and are then passed into BConvLSTM blocks. The output of these blocks then proceeds forward to the two convolution layer blocks. For training, testing, and comparison, see Badshah and Ahmad (2022) and references there in.

Conclusion

Some of the well-known active contour models for image segmentation are presented. Here both types of segmentations (global and selective) are discussed. In this chapter two-phase and multiphase segmentation models are discussed in detail. Minimization techniques for finding the optimal values and discussion about the fast numerical methods for solution of partial differential equations arising from the minimization of the models were key points of discussion in this chapter.

References

- Allen, A.M., Cahn, J.W.: A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metall.* **27**, 1085–1095 (1979)
- Alvarez, L., Lions, P.-L., Morel, J.M.: Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.* **29**(3), 845–866 (1992)
- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer, New York (2002)
- Azad, R., Asadi-Aghbolaghi, M., Fathy, M., Escalera, S.: Bi-directional convlstm U-Net with densley connected convolutions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019)
- Badshah, N., Ahmad, A.: ResBCU-Net: deep learning approach for segmentation of skin images. *Biomed. Sig. Process. Control* **71**, 103137 (2022)
- Badshah, N., Chen, K.: Multigrid method for the Chan-Vese model in variational segmentation. *Commun. Comput. Phys.* **4**(2), 294–316 (2008)
- Badshah, N., Chen, K.: On two multigrid algorithms for modeling variational multiphase image segmentation. *IEEE Trans. Image Process.* **18**(5), 1097–1106 (2009)
- Badshah, N., Chen, K.: Image selective segmentation under geometrical constraints using an active contour approach. *Commun. Comput. Phys.* **7**(4), 759–778 (2010)
- Barash, D., Schlick, T., Israeli, M., Kimmel, R.: Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps. *J. Math. Imaging Vis.* **19**, 33–48 (2003)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2010)
- Brandt, A.: Multi-level adaptive solutions to boundary-value problems. *Math. Comput.* **31**(2), 333–390 (1977)
- Bregman, L.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.* **7**(3), 200–217 (1967)
- Briggs, W.L.: *A Multigrid Tutorial*. (1999)
- Burrows, L., Guo, W., Chen, K., Torella, F.: Reproducible kernel Hilbert space based global and local image segmentation. *Inverse Probl. Imaging* **15**(1), 1–25 (2021)
- Cai, X., Chan, R., Zeng, T.: A two-stage image segmentation method using a convex variant of the Mumford-Shah model and thresholding. *SIAM J. Imaging Sci.* **6**(1), 368–390 (2013)
- Cai, X., Chan, R., Nikolova, M., Zeng, T.: A three-stage approach for segmenting degraded color images: smoothing, lifting and thresholding (SLaT). *J. Sci. Comput.* **72**, 1313–1332 (2017)
- Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vis.* **22**, 61–79 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011)
- Chan, T.F., Chen, K.: An optimization based multilevel algorithm for total variation image denoising. *SIAM J. Multiscale Model. Simul. (MMS)* **5**(2), 615–645 (2006)
- Chan, T.F., Vese, L.A.: Active Contours without Edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006)
- Chan, R., Yang, H., Zeng, T.: A two-stage image segmentation method for blurry images with poisson or multiplicative gamma noise. *SIAM J. Imaging Sci.* **7**(1), 98–127 (2014)
- Chen, K.: *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, Cambridge (2005)

- Chen, Y., Lan, G., Ouyang, Y.: Optimal primal-dual methods for a class of saddle point problems. *SIAM J. Optim.* **24**(4), 1779–1814 (2014)
- Cleeremans, A., Servan-Schreiber, D., McClelland, J.: Finite state automata and simple recurrent networks. *Neural Comput.* **1**(3), 372–381 (1989). MIT Press
- Cui, Z., Ke, R., Pu, Z., Wang, Y.: Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint, 1801.02143 (2018)
- Deng, L.-J., Guo, W., Huang, T.-Z.: Single-image super-resolution via an iterative reproducing kernel hilbert space method. *IEEE Trans. Circuits Syst. Video Technol.* **26**, 2001–2014 (2016)
- Geiser, J., Bartecki, K.: Additive, multiplicative and iterative splitting methods for Maxwell equations: algorithms and applications. In: International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2017)
- Goldstein, T., Osher, S.: The split bregman algorithm for l_1 regularized problems. *SIAM J. Imaging Sci.* **2**, 323–343 (2009)
- Gout, C., Guyader, C.L., Vese, L.: Segmentation under geometrical conditions with geodesic active contour and interpolation using level set methods. *Numer. Algorithms* **39**, 155–173 (2005)
- Guo, Y., Stein, J., Wu, G., Krishnamurthy, A.: SAU-Net: a universal deep network for cell counting. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 299–306 (2019)
- Guyader, C.L., Gout, C.: Geodesic active contour under geometrical conditions theory and 3D applications. *Numer. Algorithms* **48**, 105–133 (2008)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint, 1502.03167 (2015)
- Jeon, M., Alexander, M., Pedrycz, W., Pizzi, N.: Unsupervised hierarchical image segmentation with level set and additive operator splitting. *Pattern Recogn. Lett.* **26**(10), 1461–1469 (2005)
- Jordan, M.I.: Attractor dynamics and parallelism in a connectionist sequential machine. *Artif. Neural Netw.: Concept Learn.* 112–127 (1990)
- Jumaat, A.K., Chen, K.: An optimization based multilevel algorithm for variational image segmentation models. *Electron. Trans. Numer. Anal.* **46**, 474–504 (2017)
- Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., Wu, A.: An efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 881–892 (2002)
- Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comput. Vis.* **6**(4), 321–331 (1988)
- Lu, T., Neittaanmaki, P., Tai, X.-C.: A parallel splitting up method for partial differential equations and its application to navier-stokes equations. *RAIRO Math. Model. Numer. Anal.* **26**(6), 673–708 (1992)
- Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**, 577–685 (1989)
- Olaf, R., Philipp, F., Thomas, B.: U-Net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241 (2015)
- Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
- Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.* **4**(2), 460–489 (2005)
- Pearlmutter, B.: Learning state space trajectories in recurrent neural networks. *Neural Comput.* **1**(2), 263–269 (1989). MIT Press
- Pratondo, A., Chee-Kong, C., Sim-Heng, O.: Integrating machine learning with region-based active contour models in medical image segmentation. *J. Vis. Commun. Image R.* **43**, 1–9 (2017)
- Rada, L., Chen, K.: A new variational model with dual level set functions for selective segmentation. *Commun. Comput. Phys.* **12**(1), 261–283 (2012)

- Rada, L., Chen, K.: Improved selective segmentation model using one level set. *J. Algorithms Comput. Technol.* **7**(4), 509–541 (2013)
- Roberts, M., Chen, K., Li, J., Irion, K.: On an effective multigrid solver for solving a class of variational problems with application to image segmentation. *Int. J. Comput. Math.* **97**(10), 1–21 (2019)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithm. *Physica D* **60**(1–4), 259–268 (1992)
- Savage, J., Chen, K.: An improved and accelerated non-linear multigrid method for total-variation denoising. *Int. J. Comput. Math.* **82**(8), 1001–1015 (2005)
- Song, H., Wang, W., Zhao, S., Shen, J., Lam, K.: Pyramid dilated deeper ConvLSTM for video salient object detection. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 715–731 (2018)
- Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**, 146–159 (1994)
- Trottenberg, U., Schuller, A.: *Multigrid*. Academic, Orlando (2001)
- Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.* **50**, 271–293 (2002)
- Weickert, J., Kühne, G.: Fast methods for implicit active contours models, preprint 61. Universität des Saarlandes, Saarbrücken (2002)
- Weickert, J., ter Haar Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonlinear diffusion filtering. *Scale-space theory in computer vision. Lect. Notes Comput. Sci.* **1252**, 260–271 (1997)
- Yang, Y., Zhao, Y., Wu, B., Wang, H.: A fast multiphase image segmentation model for gray images. *Comput. Math. Appl.* **67**, 1559–1581 (2014)
- Yuan, Y., He, C.: Variational level set methods for image segmentation based on both L^2 and Sobolev gradients. *Non Linear Anal. Real World Appl.* **13**, 959–966 (2012)
- Zhao, H.-K., Osher, S., Merriman, B., Kang, M.: Implicit and non parametric shape reconstruction from unorganized data using a variational level set method. *Comput. Vis. Image Underst.* **80**(3), 295–314 (2000)



On Variable Splitting and Augmented Lagrangian Method for Total Variation-Related Image Restoration Models

13

Zhifang Liu, Yuping Duan, Chunlin Wu, and Xue-Cheng Tai

Contents

Introduction	504
Basic Notation	507
Augmented Lagrangian Method for Total Variation-Related Image Restoration Models	508
Augmented Lagrangian Method for TV- L^2 Restoration	510
Augmented Lagrangian Method for TV- L^2 Restoration with Box Constraint	516
Augmented Lagrangian Method for TV Restoration with Non-quadratic Fidelity	519
Extension to Multichannel Image Restoration	524
The Multichannel TV Restoration Model	524
Augmented Lagrangian Method for Multichannel TV Restoration	526
Extension to High-Order Models	528
Augmented Lagrangian Method for Second-Order Total Variation Model	528
Augmented Lagrangian Method for Total Generalized Variation Model	531
Augmented Lagrangian Method for Euler Elastic-Based Model	536
Augmented Lagrangian Method for Mean Curvature-Based Model	539
Numerical Experiments	541
Conclusions	546
References	546

Z. Liu

School of Mathematical Sciences, Tianjin Normal University, Tianjin, China

e-mail: matlzhf@tjnu.edu.cn

Y. Duan

Center for Applied Mathematics, Tianjin University, Tianjin, China

e-mail: yuping.duan@tju.edu.cn

C. Wu (✉)

School of Mathematical Sciences, Nankai University, Tianjin, China

e-mail: wucl@nankai.edu.cn

X.-C. Tai

Hong Kong Center for Cerebro-cardiovascular Health Engineering (COCHE), Shatin, Hong Kong

e-mail: xtai@hkcoche.org

Abstract

Variable splitting and augmented Lagrangian method are widely used in image processing. This chapter briefly reviews its applications for solving the total variation (TV) related image restoration problems. Due to the nonsmoothness of TV, related models and variants are nonsmooth convex or nonconvex minimization problems. Variable splitting and augmented Lagrangian method can benefit from the separable structure and efficient subsolvers, and has convergence guarantee in convex cases. We present this approach for a number of TV minimization models including TV- L^2 , TV- L^1 , TV with nonquadratic fidelity term, multichannel TV, high-order TV, and curvature minimization models.

Keywords

Variable splitting · Augmented lagrangian method · Total variation · Image restoration · Box constraint

Introduction

This short survey provides a brief review of the variable splitting and augmented Lagrangian method for total variation (TV)-related image restoration models. We will focus on this computational problem closely, and do not plan to touch other related topics like theoretical model analysis and algorithmic connections, which can be referred to, e.g., Aubert and Kornprobst (2010) and Glowinski et al. (2016) and references therein. Also, to keep the context as compact as possible, we would not expand all the details, although there are definitely lots of excellent works in the literature.

Total variation, which is a semi-norm of the space of functions of bounded variation, was first proposed for image denoising by Rudin, Osher, and Fatemi (ROF) in Rudin et al. (1992). In the discrete setting, it is essentially the L_1 norm of gradients and can maintain the sparse discontinuities. Therefore, it is appropriate to preserve image edges that are usually the most important features for images to recover. Owing to its edge-preserving property and convexity, total variation has been demonstrated very successful and become popular in image restoration like image denoising (Rudin et al. 1992; Le et al. 2007), image deblurring (Chan and Wong 1998; Wu and Tai 2010) and image inpainting (Bertalmio et al. 2003) and also various other types of image processing tasks including image decomposition (Vese and Osher 2003), image segmentation (Chan and Vese 2001), CT reconstruction (Persson et al. 2001), phase retrieval (Chang et al. 2016) and so on.

The total variation model has been generalized in many ways for different purposes. The original total variation regularization was proposed for gray image restoration (Rudin et al. 1992), which is the single channel case. To restore multichannel data, such as color images with RGB channels, people extended it to color TV and vectorial TV regularizations (Blomgren and Chan 1998; Sapiro

and Ringach 1996). It is well-known that images recovered by total variation regularized models have the undesired staircase effect. To prevent the total variation oversharpening, there are several remarkable methods to improve the total variation regularization. These include the variable exponent TV models (Chen et al. 2006) and a wide class of high-order models, such as inf-convolution model (Chambolle and Lions 1997), second-order total variation model (Lysaker et al. 2003), bounded Hessian model (Hinterberger and Scherzer 2006), total generalized variation model (Bredies et al. 2010), and total fractional-order variation model (Zhang and Chen 2015) etc. By co-area formula, the total variation is the integral of lengths of all level curves of the intensity function. One natural extension way is thus to introduce curve curvature term for regularization. For example, Euler's elastica which contains both lengths and curvatures was proposed for image inpainting (Chan et al. 2002; Yashtini and Kang 2016), denoising (Tai et al. 2011; Duan et al. 2013), zooming (Tai et al. 2011; Duan et al. 2013), illusory contour (Kang et al. 2014), image decomposition (Liu et al. 2018), and image reconstruction (Yan and Duan 2020). Such regularity can provide strong priors for the continuity of edges. Another total variation-related geometric regularization technique we would like to mention is mean curvature minimization (Zhu and Chan 2012), which considers the image or graph in a high-dimensional space and transfers the image minimization problems to the corresponding surface minimization problems. From the viewpoint of image domain, total variation regularization was also extended to implicit surfaces, triangulated meshes and even general manifolds for image and data processing on curved spaces (Lai and Chan 2011; Wu et al. 2012) and normal vector filtering for surface denoising (Zhang et al. 2015). By exploiting the spatial interactions in images, total variation regularization was also generalized to nonlocal TV (Lou et al. 2010). By using non-convex penalty functions instead of the L_1 norm, non-convex TV regularizations got more and more attentions in recent years; see Chen et al. (2012), Hintermüller and Wu (2013), Wu et al. (2018), and Selesnick et al. (2020) and the references therein. They have been shown capable to generate good results with neat edges, as indicated by the interesting lower bound theory Nikolova (2005); Chen et al. (2012); Zeng and Wu (2018); Feng et al. (2018).

However, the non-smoothness of the total variation semi-norm gives rise to a challenge of its minimization. To overcome this problem, the common way is replacing total variation by its smoothed versions in image restoration model. Therefore, one can solve the new associated Euler-Lagrangian equation and obtain an approximate solution of the original model (Acar and Vogel 1994). For solving this Euler-Lagrangian equation, Rudin, Osher and Fatemi proposed a gradient flow method (Rudin et al. 1992). This method is slow due to strict constraints on the time step size and many methods have been proposed to improve on it. Some efficient methods are dual methods (Chambolle 2004; Chambolle and Pock 2011), the split Bregman method (Goldstein and Osher 2009) and splitting-and-penalty based methods (Wang et al. 2008), proximity algorithms (Micchelli et al. 2011), alternating direction method of multipliers (Chan et al. 2013) and augmented Lagrangian methods (Tai and Wu 2009; Wu and Tai 2010; Wu et al. 2011, 2012).

The augmented Lagrangian method was originally introduced by Hestenes and Powell for solving constrained optimization problem and further systematically studied by many researchers, such as Rockafellar (1974) and Bertsekas (1996 (firstly published in 1982)). It was also widely applied to optimize unconstrained minimization problem with the aid of operator-splitting technique (Glowinski and Tallec 1989) by which one can transform the unconstrained optimization problem to its equivalent constrained versions. One of the special and very useful instance of augmented Lagrangian methods is the alternating direction method of multipliers (ADMM) (Boyd 2010; He and Yuan 2012), which is famous in optimization and statistics community and has broad applications. ADMM has been extensively studied in recent decades and has many practical variants, such as linearized ADMM (Wang and Yuan 2012), preconditioned ADMM (Deng and Yin 2016), proximal ADMM (Fazel et al. 2013), accelerated ADMM (Ouyang et al. 2015), stochastic ADMM (Chen et al. 2018) and non-convex ADMM (Li and Pong 2015; Wang et al. 2019).

Indeed, the variable splitting and augmented Lagrangian method gained great successes in solving nonlinear variational problems that arise from physics, mechanics, economics, etc. (Glowinski and Tallec 1989). The variable splitting step helps to transform a complicated problem into a constrained optimization with more variables, then an iteration based on augmented Lagrangian method is performed with several easier subproblems. Inspired by this, the method was proposed by Tai and Wu to optimize the total variation-based image restoration model in Tai and Wu (2009) and Wu and Tai (2010). As expected, augmented Lagrangian methods benefit from the periodic boundary condition which is commonly assumed for image processing problems and the L_1 norm which is included in the total variation seminorm. The augmented Lagrangian method for TV-based image restoration model has two subproblems. The periodic boundary condition allows us to solve one of the subproblems via Fourier transformation with FFT implementation in the case of deconvolution case. Meanwhile, the other subproblem with the L^1 norm has closed form solution. Despite the fact that the image processing problems are naturally in large scale, these two advantages of the augmented Lagrangian method make it efficient in minimizing the objective functionals related with the non-smooth total variation for various image processing tasks. Since Tai and Wu (2009); Wu and Tai (2010), the variable splitting and augmented Lagrangian method has been widely applied to total variation-related minimizations like the single channel case (Wu and Tai 2010; Tai and Wu 2009; Wu et al. 2011), the multichannel case (Wu and Tai 2010), high-order models (Wu and Tai 2010), TV-Stokes model (Hahn et al. 2012), Euler's elastica image restoration model (Tai et al. 2011; Duan et al. 2013; Yashtini and Kang 2016), mean curvature image denoising (Zhu et al. 2013; Myllykoski et al. 2015), total variation minimization in curved spaces for either data processing (Lai and Chan 2011; Wu et al. 2012) or normal vector-filtering based surface denoising (Zhang et al. 2015) and even more in Ramani and Fessler (2011) and Güven et al. (2016). Therein for some complicated non-convex models like Euler's elastica or mean curvature based, how to introduce the auxiliary variables is tricky and important to get stable and efficient

algorithms. There are some close connections between the augmented Lagrangian method and other approaches such as split Bregman method (Goldstein and Osher 2009) and Chambolle’s projection method (Chambolle 2004), and some works for improving classical augmented Lagrangian method can be found in Li et al. (2013), etc.

The content included here are organized as follows. In section “[Basic Notation](#)”, we present some basic notations. In section “[Augmented Lagrangian Method for Total Variation-Related Image Restoration Models](#)”, we present augmented Lagrangian methods TV restoration models with L^2 fidelity term and TV restoration models with non-quadratic fidelity. In Section “[Extension to Multichannel Image Restoration](#)”, we present augmented Lagrangian methods for multichannel TV restoration. In Section “[Extension to High-Order Models](#)”, we present augmented Lagrangian methods for high-order models, including second-order total variation model, total generalized variation model, Euler’s elastica model, and mean curvature model. In Section “[Numerical Experiments](#)”, we show some numerical experiments. We conclude this paper in Section “[Conclusions](#)”.

Basic Notation

We follow Wu and Tai (2010) for most notations. As a gray image is a 2D array, we represent it by an $N \times N$ matrix, without the loss of generality. It is useful to denote the Euclidean space $\mathbb{R}^{N \times N}$ as \mathcal{X} and write $\mathcal{Y} = \mathcal{X} \times \mathcal{X}$. We recall the discrete gradient operator

$$\begin{aligned} \nabla : \mathcal{X} &\rightarrow \mathcal{Y} \\ x &\rightarrow \nabla x, \end{aligned}$$

where ∇x is given by

$$(\nabla x)_{i,j} = ((\mathring{D}_1^+ x)_{i,j}, (\mathring{D}_2^+ x)_{i,j}), i, j = 1, \dots, N,$$

with

$$\begin{aligned} (\mathring{D}_1^+ x)_{i,j} &= \begin{cases} x_{i,j+1} - x_{i,j}, & 1 \leq j \leq N-1, \\ x_{i,1} - x_{i,N}, & j = N, \end{cases} \\ (\mathring{D}_2^+ x)_{i,j} &= \begin{cases} x_{i+1,j} - x_{i,j}, & 1 \leq i \leq N-1, \\ x_{1,j} - x_{N,j}, & i = N. \end{cases} \end{aligned}$$

Here \mathring{D}_1^+ and \mathring{D}_2^+ are used to denote forward difference operators with periodic boundary condition for FFT algorithm implementation. We mention that other boundary conditions with corresponding implementation tricks can also be adopted.

The usual inner products and L^2 norms in the spaces \mathcal{X} and \mathcal{Y} are as follows. We denote

$$\langle x, z \rangle = \sum_{1 \leq i, j \leq N} x_{i,j} z_{i,j} \text{ and } \|x\| = \sqrt{\langle x, x \rangle},$$

for $x, z \in \mathcal{X}$; and

$$\langle w, y \rangle = \langle w^1, y^1 \rangle + \langle w^2, y^2 \rangle, \text{ and } \|y\| = \sqrt{\langle y, y \rangle},$$

for $y = (y^1, y^2) \in \mathcal{Y}$ and $w = (w^1, w^2) \in \mathcal{Y}$. At each pixel (i, j) , we define

$$|y_{i,j}| = |(y_{i,j}^1, y_{i,j}^2)| = \sqrt{(y_{i,j}^1)^2 + (y_{i,j}^2)^2}$$

as the usual Euclidean norm in \mathbb{R}^2 . We mention that $\|x\|_{L^p}$ is used to denote the general L^p norm of $x \in \mathcal{X}$.

By using the inner products of \mathcal{X} and \mathcal{Y} , it is clear that the discrete divergence operator, as the adjoint operator of $-\nabla$, is as follows

$$\begin{aligned} \text{div} : \mathcal{Y} &\rightarrow \mathcal{X} \\ y = (y^1, y^2) &\rightarrow \text{div } y, \end{aligned}$$

where

$$(\text{div } y)_{i,j} = y_{i,j}^1 - y_{i,j-1}^1 + y_{i,j}^2 - y_{i-1,j}^2 = (\mathring{D}_1^- y^1)_{i,j} + (\mathring{D}_2^- y^2)_{i,j},$$

with backward difference operators \mathring{D}_1^- and \mathring{D}_2^- and periodic boundary conditions $y_{i,0}^1 = y_{i,N}^1$ and $y_{0,j}^2 = y_{N,j}^2$.

Augmented Lagrangian Method for Total Variation-Related Image Restoration Models

We assume $d \in \mathcal{X}$ to be an observed image. As usual, we model the degradation procedure as

$$\underline{x} \xrightarrow{\text{linear transformation}} K \underline{x} \xrightarrow{\text{noise}} d, \quad (1)$$

where $\underline{x} \in \mathcal{X}$ is the ground truth image and $K : \mathcal{X} \rightarrow \mathcal{X}$ is a linear operator like a blur. In other cases, such as when K is a Radon transform or a subsampling, the dimensions of the observed data d and the ground truth data \underline{x} may be different. However, there is no essential difficulty, and the method framework here also

applies. Here the noise is not necessarily to be additive and could be Gaussian, impulsive, Poisson, or even others. The task of image restoration is to recover \underline{x} from d . In this survey we only consider the case where the linear operator K is given. Even so, we usually cannot directly solve \underline{x} from (1), because this is a typical inverse problem. Both the random measurement noise and the bad condition number of K bring computational difficulties. Regularization on the solution should be considered to overcome the ill-posedness.

Although the classical Tikhonov regularization has achieved great successes in lots of general inverse problems, it turns out to over smooth image edges, the most important image structure. Indeed, one of the most basic and successful image restoration models is based on total variation regularization, which reads

$$\min_{x \in \mathcal{X}} \{E(x) = F(Kx) + R(\nabla x) + B(x)\}, \quad (2)$$

where $F(Kx)$ is a fidelity term, $R(\nabla x)$ is the total variation of x (Rudin et al. 1992) defined by

$$R(\nabla x) = \text{TV}(x) = \sum_{1 \leq i, j \leq N} |(\nabla x)_{i,j}|, \quad (3)$$

and $B(x)$ is an indicator function of box constraints defined as follows

$$B(x) = \begin{cases} 0, & \underline{b} \leq x_{i,j} \leq \bar{b}, \forall i, j, \\ +\infty, & \text{otherwise.} \end{cases}$$

Lots of researches (Le et al. 2007; Beck and Teboulle 2009; Chan et al. 2013) show that to involve this kind of constraints is useful, when the intensity range is clear. Otherwise, one can just let the box parameters \underline{b} be $-\infty$ or \bar{b} be $+\infty$. This model includes numerous particular cases studied in the literatures.

For further analysis and interpretation, we make the following assumptions:

- Assumption 1. $\text{Null}(\nabla) \cap \text{Null}(K) = \{0\}$.
- Assumption 2. $\text{dom}(R \circ \nabla) \cap \text{dom}(F \circ K) \cap \text{dom}(B) \neq \emptyset$.
- Assumption 3. $F(z)$ is convex, proper, coercive, and lower semi-continuous.
- Assumption 4. $\text{dom}(F)$ is open.

where $\text{Null}(\cdot)$ is the null space of \cdot ; $\text{dom}(F) = \{z \in \mathcal{X} : F(z) < +\infty\}$ is the domain of F ; and $\text{dom}(R \circ \nabla)$, $\text{dom}(B)$, $\text{dom}(F \circ K)$ are similar. Here we have some comments on these assumptions, which are relatively quite natural. Since most linear operators K 's like blur kernels correspond essentially to averaging operations, Assumption 1 is reasonable. Moreover, although the fidelity terms $F(\cdot)$'s are diverse by the statistics of the noise models, many of them meet all of those Assumption 3 and 4, like the following typical ones:

1. The squared L^2 fidelity (corresponding to Gaussian noise):

$$F(Kx) = \frac{\alpha}{2} \|Kx - d\|^2,$$

2. The L^1 fidelity (Nikolova 2004) (corresponding to impulsive noise):

$$F(Kx) = \alpha \|Kx - d\|_{L^1},$$

3. The Kullback-Leibler (KL) divergence fidelity (corresponding to Poisson noise, assuming $d_{i,j} > 0, \forall i, j$, as in Le et al. (2007)):

$$F(Kx) = \begin{cases} \alpha \sum_{1 \leq i, j \leq N} ((Kx)_{i,j} - d_{i,j} \log(Kx)_{i,j}), & (Kx)_{i,j} > 0, \forall i, j, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\alpha > 0$ is a parameter. Note for Poisson noise, we use the definition of the fidelity on the whole space for analysis convenience, compared to Le et al. (2007) (where $K = I$) and (Brune et al. 2009).

Under the Assumptions 1, 2, 3, and 4, it is not difficult to see that the functional $E(x)$ in (2) is convex, proper, coercive, and lower semi-continuous. Thus we have the following existence and uniqueness result, by the generalized Weierstrass theorem and Fermat's rule (Glowinski and Tallec 1989; Rockafellar and Wets 1998).

Theorem 1. *The minimization problem (2) has at least one solution x , which satisfies*

$$0 \in K^* \partial F(Kx) - \text{div } \partial R(\nabla x) + \partial B(x), \quad (4)$$

with $\partial F(Kx)$ and $\partial R(\nabla x)$ being the sub-differentials (Rockafellar and Wets 1998) of F at Kx and R at ∇x , respectively. Moreover, if $F \circ K(x)$ is strictly convex, the minimizer is unique.

Next, we present to use the augmented Lagrangian method for TV regularization-based image restoration models (2) which satisfy our assumptions.

Augmented Lagrangian Method for TV- L^2 Restoration

In this section, we review the augmented Lagrangian method proposed for the TV restoration model with L^2 fidelity term (Tai and Wu 2009; Wu and Tai 2010)

$$\min_{x \in \mathcal{X}} \left\{ E_{\text{TV}}(x) = \frac{\alpha}{2} \|Kx - d\|^2 + R(\nabla x) \right\}, \quad (5)$$

where $\alpha > 0$ and $R(\nabla x)$ is defined as in (3). This model is a special case of model (2), where $F(Kx) = \frac{\alpha}{2} \|Kx - d\|^2$ and the box constraint vanishes. In the literatures, people commonly call model (5) as TV- L^2 model.

The TV- L^2 model is a fundamental model in image restoration, which is usually applied for removing Gaussian-type noise and the linear degradation like blur in image restoration problems (Rudin et al. 1992; Acar and Vogel 1994). By standard Bayesian estimation, the L^2 fidelity term is deduced from the statistical distribution of the i.i.d Gaussian noise, which guarantees that the recovered image resembles the underly truth image closely. Meanwhile, the total variation regularization preserves the sharp edges.

As we mentioned before, the total variation term is non-smooth and is a compound of the L^1 norm and the gradient operator. There is a basic idea that decouples the total variation term and treats the L^1 norm and the gradient operator separately. By combining this with variable splitting technique, the augmented Lagrangian method demonstrates this idea.

First, we introduce an auxiliary variable $y \in \mathcal{Y}$ for ∇x and convert the minimization problem (5) to an equivalent constrained optimization problem

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \left\{ G_{\text{TV}}(x, y) = \frac{\alpha}{2} \|Kx - d\|^2 + R(y) \right\}, \\ \text{s.t. } y = \nabla x. \end{aligned} \quad (6)$$

Then, we define the following augmented Lagrangian function for the constrained optimization problem (6)

$$\mathcal{L}_{\text{TV}}(x, y; \lambda) = \frac{\alpha}{2} \|Kx - d\|^2 + R(y) + \langle \lambda, y - \nabla x \rangle + \frac{\beta}{2} \|y - \nabla x\|^2, \quad (7)$$

with the Lagrange multiplier $\lambda \in \mathcal{Y}$ and a positive penalty parameter β . The augmented Lagrangian method for the problem (6) is to seek a saddle-point of the augmented Lagrangian function (7):

$$\begin{aligned} \text{Find } (x^*, y^*, \lambda^*) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}, \\ \text{s.t. } \mathcal{L}_{\text{TV}}(x^*, y^*; \lambda) \leq \mathcal{L}_{\text{TV}}(x^*, y^*; \lambda^*) \leq \mathcal{L}_{\text{TV}}(x, y; \lambda^*), \\ \forall (x, y, \lambda) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}, \end{aligned} \quad (8)$$

The following theorem (Glowinski and Tallec 1989; Wu and Tai 2010) reveals the relation between the solution of problem (5) and the saddle-point of problem (8).

Theorem 2. $x^* \in \mathcal{X}$ is a solution of problem (5) if and only if there exist $y^* \in \mathcal{Y}$ and $\lambda^* \in \mathcal{Y}$ such that $(x^*, y^*; \lambda^*)$ is a saddle-point of problem (8).

Finally, we employ an alternating direction iterative procedure in the augmented Lagrangian method to seek a saddle-point of problem (8); see Algorithm 1.

Algorithm 1 Augmented Lagrangian method for TV- L^2 model

Initialization: $x^{-1} = 0, y^{-1} = 0, \lambda^0 = 0$.

Iteration: For $k = 0, 1, \dots$:

1. Compute (x^k, y^k) as an (approximate) minimizer of the augmented Lagrangian function (7) with the Lagrange multiplier λ^k , i.e.,

$$(x^k, y^k) \approx \arg \min_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \mathcal{L}_{\text{TV}}(x, y; \lambda^k), \tag{9}$$

where $\mathcal{L}_{\text{TV}}(x, y; \lambda^k)$ is defined as (7).

2. Update

$$\lambda^{k+1} = \lambda^k + \beta(y^k - \nabla x^k).$$

We can see that the minimization problem (9) still can not be solved directly and exactly. Our strategy is separating the problem (9) into two subproblems with respect to x and y and minimizing them alternatively.

The Solution to Sub-problem w.r.t. x

Given y , the minimization problem (9) with respect to x is

$$\min_{x \in \mathcal{X}} \left\{ \frac{\alpha}{2} \|Kx - d\|^2 - \langle \lambda^k, \nabla x \rangle + \frac{\beta}{2} \|y - \nabla x\|^2 \right\}.$$

It is a quadratic optimization problem, whose first-order optimality condition gives a linear equation

$$(\alpha K^* K - \beta \Delta)x = \alpha K^* d - \text{div}(\lambda^k + \beta y). \tag{10}$$

If K is a convolution operator like a convolution blur, the above equation under periodic boundary condition can be efficiently solved via Fourier transform with fast Fourier transform (FFT) implementation (Wang et al. 2008; Wu and Tai 2010). One can obtain its solution by

$$x = \mathcal{F}^{-1} \left(\frac{\alpha \mathcal{F}(K^*) \mathcal{F}(d) - \mathcal{F}(\text{div}) \mathcal{F}(\lambda^k + \beta y)}{\alpha \mathcal{F}(K^*) \mathcal{F}(K) - \beta \mathcal{F}(\Delta)} \right),$$

where \mathcal{F} and \mathcal{F}^{-1} denote the Fourier transform and the inverse Fourier transform. Fourier transforms of operators K^* , K , div , and Δ mean the transforms of the corresponding convolution kernels. If K is not a convolution operator, such as a Radon transform or a subsampling, we can solve the above equation (10) by other well-developed linear solvers like conjugate gradient (CG) method.

The Solution to Sub-problem w.r.t. y

Given x , the minimization problem (9) with respect to y is

$$\min_{y \in \mathcal{Y}} \left\{ R(y) + \langle \lambda^k, y \rangle + \frac{\beta}{2} \|y - \nabla x\|^2 \right\}. \quad (11)$$

According to the definition of $R(y)$, we can rewrite (11) as

$$\min_{y \in \mathcal{Y}} \left\{ \sum_{1 \leq i, j \leq N} |y_{i,j}| + \frac{\beta}{2} \sum_{1 \leq i, j \leq N} \left| y_{i,j} - \left(\nabla x - \frac{\lambda^k}{\beta} \right)_{i,j} \right|^2 \right\}, \quad (12)$$

whose solution is in closed form as follows

$$y_{i,j} = \max \left(0, 1 - \frac{1}{\beta |\eta_{i,j}|} \right) \eta_{ij}, \quad (13)$$

where $\eta = \nabla x - \lambda^k/\beta \in \mathcal{Y}$. This solution can be derived from the first-order optimality condition via the subdifferential theory (Wang et al. 2008) or the geometric explanation of the minimizer (Wu et al. 2011). We remark that the geometric method can be easily extended to higher (>2) dimensional case (Wu and Tai 2010; Wu et al. 2011) (see, e.g., multichannel image restoration and high-order models in later sections) or the case where $R(\cdot)$ is non-convex (Wu et al. 2018).

Here, we review the geometric interpretation of the formula (13) given in Wu et al. (2011). As one can see, the problem (12) is separable, and at each pixel (i, j) , we can reduce it to a simple form

$$\min_{u \in \mathbb{R}^2} \left\{ |u| + \frac{\beta}{2} |u - v|^2 \right\}, \quad (14)$$

where $v \in \mathbb{R}^2$; see Fig. 1.

In fact, the minimizer of (14) locates in the same quadrant of v and inside of the solid circle with O as center and $|v|$ as radius; see Fig. 1. Without loss of generality, we consider the points inside the solid circle at the first quadrant, e.g., u . We draw a dotted circle with O as center and $|u|$ as radius, which intersects the line segment Ov at a point u^* . By the triangle inequality, we have

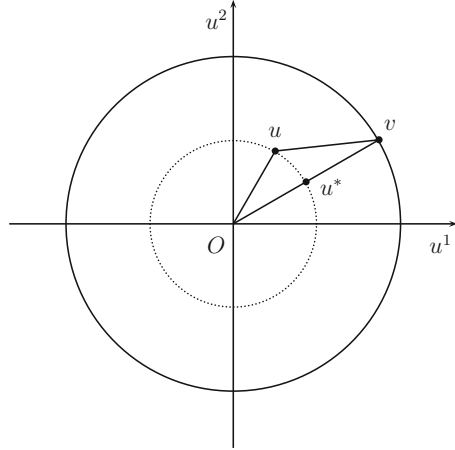
$$|u| + |u - v| \geq |v| = |u^*| + |u^* - v|.$$

Since $|u| = |u^*|$, we obtain

$$|u - v| \geq |u^* - v|,$$

which indicates

Fig. 1 A geometric interpretation of the formula (13)



$$|u| + \frac{\beta}{2}|u - v|^2 \geq |u^*| + \frac{\beta}{2}|u^* - v|^2.$$

The above equality implies that the solution of (14) locates on the line segment Ov . Therefore, we let $u = \gamma v$ with $0 \leq \gamma \leq 1$ and simplify the problem (14) into an univariate optimization problem

$$\min_{0 \leq \gamma \leq 1} \left\{ \gamma|v| + \frac{\beta}{2}(\gamma - 1)^2|v|^2 \right\}. \tag{15}$$

The above problem (15) can be solved exactly and has a closed form solution

$$\gamma^* = \max \left(0, 1 - \frac{1}{\beta|v|} \right).$$

According to (10) and (13), we can solve (9) by an alternating minimization procedure; see Algorithm 2.

Algorithm 2 Augmented Lagrangian method for TV- L^2 model – solve the minimization problem (9)

Initialization: $x^{k,0} = x^{k-1}, y^{k,0} = y^{k-1}$.

Iteration: For $l = 0, 1, \dots, L - 1$:

- Compute $x^{k,l+1}$ by solving (10) for $y = y^{k,l}$;
- Compute $y^{k,l+1}$ from (13) for $x = x^{k,l+1}$.

Output: $x^k = x^{k,L}, y^k = y^{k,L}$.

Here L can be chosen using some convergence test techniques. In fact, setting $L = 1$ is sufficient to establish the convergence of the sequence (Wu and Tai 2010) generated by Algorithm 1. In this case, the augmented Lagrangian method is well-known as the alternating direction method of multipliers (Boyd 2010).

Convergence Analysis

In this section, we present some convergence results of Algorithm 1. Actually, we can verify that Algorithm 1 is convergence in two cases, i.e., when the minimization problem (9) is exactly solved in each iteration and the problem (9) is roughly solved in each iteration (Glowinski and Tallec 1989; Wu and Tai 2010). We comment that the convergence proof in Wu and Tai (2010) is based on Glowinski and Tallec (1989) but reduces the uniform convexity assumption of $R(\cdot)$. Here, we just take the main convergence results from Wu and Tai (2010) and omit the details.

In the first case, we should set $L \rightarrow \infty$ in Algorithm 2, and the inner iteration is guaranteed to converge.

Theorem 3. *The sequence $\{(x^{k,l}, y^{k,l}) : l = 0, 1, 2, \dots\}$ generated by Algorithm 2 converges to a solution of the problem (9).*

Theorem 4. *Assume that $(x^*, y^*; \lambda^*)$ is a saddle-point of $\mathcal{L}_{\text{TV}}(x, y; \lambda)$. Suppose that the minimization problem (9) is exactly solved in each iteration; i.e., $L \rightarrow \infty$ in Algorithm 2. Then the sequence $(x^k, y^k; \lambda^k)$ generated by Algorithm 1 satisfies*

$$\begin{cases} \lim_{k \rightarrow \infty} G_{\text{TV}}(x^k, y^k) = G_{\text{TV}}(x^*, y^*), \\ \lim_{k \rightarrow \infty} \|y^k - \nabla x^k\| = 0. \end{cases} \quad (16)$$

Since $R(y)$ is continuous, (16) indicates that x^k is a minimizing sequence of $E_{\text{TV}}(\cdot)$. If we further have $\text{Null}(K) = \{0\}$, then

$$\begin{cases} \lim_{k \rightarrow \infty} x^k = x^*, \\ \lim_{k \rightarrow \infty} y^k = y^*. \end{cases}$$

In the second case, we set $L = 1$ in Algorithm 2.

Theorem 5. *Assume that $(x^*, y^*; \lambda^*)$ is a saddle-point of $\mathcal{L}_{\text{TV}}(x, y; \lambda)$. Suppose that the minimization problem (9) is roughly solved in each iteration, i.e., with $L = 1$ in Algorithm 2. Then the sequence $(x^k, y^k; \lambda^k)$ generated by Algorithm 1 satisfies*

$$\begin{cases} \lim_{k \rightarrow \infty} G_{\text{TV}}(x^k, y^k) = G_{\text{TV}}(x^*, y^*), \\ \lim_{k \rightarrow \infty} \|y^k - \nabla x^k\| = 0. \end{cases} \quad (17)$$

Since $R(y)$ is continuous, (17) indicates that x^k is a minimizing sequence of $E_{\text{TV}}(\cdot)$. If we further have $\text{Null}(K) = \{0\}$, then

$$\begin{cases} \lim_{k \rightarrow \infty} x^k = x^*, \\ \lim_{k \rightarrow \infty} y^k = y^*. \end{cases}$$

Augmented Lagrangian Method for TV- L^2 Restoration with Box Constraint

In this section, we review the augmented Lagrangian method for the TV restoration model with the L^2 fidelity term and the box constraint (Chan et al. 2013), which reads

$$\min_{x \in \mathcal{X}} \left\{ E_{\text{TVB}}(x) = \frac{\alpha}{2} \|Kx - d\|^2 + R(\nabla x) + B(x) \right\}, \quad (18)$$

where $\alpha > 0$, $R(\nabla x)$ is defined as (3), and we have $-\infty < \underline{b} \leq \bar{b} < +\infty$ in $B(x)$. This model is also a special case of model (2), where $F(Kx) = \frac{\alpha}{2} \|Kx - d\|^2$.

The box constraint is inherent in digital image processing. The nature image is stored as discrete numerical arrays in some digital media. The typical used ranges are $[0, 1]$ and $[0, 255]$. It has been shown that adding the box constraint in image restoration can improve the quality of the recovered image (Beck and Teboulle 2009; Chan et al. 2013).

The original method proposed in Chan et al. (2013) is under the framework of the alternating direction method of multipliers, which is a special case of the augmented Lagrangian method. For the sake of clarity, we reformulate it in our notations and styles.

Compared with the TV- L^2 model (5), this model has one more non-differentiability term $B(x)$. Thus, we need another variable to eliminate the nondifferentiation for x . We introduce two auxiliary variables $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$ and rewrite the problem (18) to be the following constrained optimization problem

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}} & \left\{ G_{\text{TVB}}(x, y, z) = \frac{\alpha}{2} \|Kx - d\|^2 + R(y) + B(z) \right\} \\ \text{s.t.} & \quad \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x, \end{aligned} \quad (19)$$

where $\mathcal{I}_1 : \mathcal{X} \rightarrow \mathcal{X}$ is the identity operator.

We define the augmented Lagrangian function for the problem (19) as follows

$$\begin{aligned} \mathcal{L}_{\text{TVB}}(x, y, z; \lambda_y, \lambda_z) &= \frac{\alpha}{2} \|Kx - d\|^2 + R(y) + B(z) \\ &+ \left\langle \begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x \right\rangle \\ &+ \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x \right\|_{\mathcal{S}}^2, \end{aligned} \quad (20)$$

where $\begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}$ is the Lagrangian multiplier and $\mathcal{S} = \begin{pmatrix} \beta_y \mathcal{I}_2 & \\ & \beta_z \mathcal{I}_1 \end{pmatrix}$ with the identity operator $\mathcal{I}_2 : \mathcal{Y} \rightarrow \mathcal{Y}$ and positive parameters β_y, β_z . Here $\|u\|_{\mathcal{S}}$ denotes the \mathcal{S} -norm, defined by $\|u\|_{\mathcal{S}} = \sqrt{\langle u, \mathcal{S}u \rangle}$.

For the augmented Lagrangian method, we consider the saddle-point problem

$$\begin{aligned} &\text{Find } (x^*, y^*, z^*, \lambda_y^*, \lambda_z^*) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{X}, \\ &\text{s.t. } \mathcal{L}_{\text{TVB}}(x^*, y^*, z^*; \lambda_y, \lambda_z) \leq \mathcal{L}_{\text{TVB}}(x^*, y^*, z^*; \lambda_y^*, \lambda_z^*) \leq \mathcal{L}_{\text{TVB}}(x, y, z; \lambda_y^*, \lambda_z^*), \\ &\quad \forall (x, y, z, \lambda_y, \lambda_z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{X}. \end{aligned} \quad (21)$$

Finally, we use an alternating direction iterative scheme in the augmented Lagrangian method to solve the saddle-point problem (21); see Algorithm 3.

Algorithm 3 Augmented Lagrangian method for TV- L^2 model with box constraint

Initialization: $x^{-1} = 0, \begin{pmatrix} y^{-1} \\ z^{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \lambda_y^0 \\ \lambda_z^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Iteration: For $k = 0, 1, \dots$:

1. Compute (x^k, y^k, z^k) as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier $\begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}$, i.e.,

$$(x^k, y^k, z^k) \approx \arg \min_{(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{X}} \mathcal{L}_{\text{TVB}}(x, y, z; \lambda_y^k, \lambda_z^k), \quad (22)$$

where $\mathcal{L}_{\text{TVB}}(x, y, z; \lambda_y^k, \lambda_z^k)$ is as in (20).

2. Update

$$\begin{pmatrix} \lambda_y^{k+1} \\ \lambda_z^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix} + \begin{pmatrix} \beta_y (y^k - \nabla x^k) \\ \beta_z (z^k - x^k) \end{pmatrix}.$$

To solve the minimization problem (22), we separate it into two subproblems respect to x and $\begin{pmatrix} y \\ z \end{pmatrix}$ and employ an alternative minimization procedure.

The Solution to Sub-problem w.r.t. x

Given $\begin{pmatrix} y \\ z \end{pmatrix}$, the minimization problem (22) with respect to x reads

$$\min_{x \in \mathcal{X}} \left\{ \frac{\alpha}{2} \|Kx - d\|^2 - \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x \right\rangle + \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x \right\|_{\mathcal{S}}^2 \right\}, \quad (23)$$

whose first-order optimization condition gives a linear equation

$$(\alpha K^* K - \beta_y \Delta + \beta_z \mathcal{I}_1) x = \alpha K^* d - \operatorname{div}(\lambda_y^k + \beta_y y) + \lambda_z^k + \beta_z z. \quad (24)$$

Similar to the equation (10), the above equation can be efficiently solved by fast linear solvers such as FFT and CG.

The Solution to Sub-problem w.r.t. (y, z)

Given x , the minimization problem (22) with respect to $\begin{pmatrix} y \\ z \end{pmatrix}$ reads

$$\min_{(y,z) \in \mathcal{Y} \times \mathcal{Z}} \left\{ R(y) + B(z) + \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} \right\rangle + \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ \mathcal{I}_1 \end{pmatrix} x \right\|_{\mathcal{S}}^2 \right\}, \quad (25)$$

which can be separated into two independent minimization problems:

- y -subproblem:

$$\min_{y \in \mathcal{Y}} \left\{ R(y) + (\lambda_y^k, y) + \frac{\beta_y}{2} \|y - \nabla x\|^2 \right\}, \quad (26)$$

- z -subproblem:

$$\min_{z \in \mathcal{Z}} \left\{ B(z) + (\lambda_z^k, z) + \frac{\beta_z}{2} \|z - x\|^2 \right\}. \quad (27)$$

We can obtain the minimizer of (26) from (13) and the minimizer of (27) as follows

$$z_{i,j} = \mathcal{P}_{[\underline{b}, \bar{b}] }(\xi_{i,j}), \quad \forall i, j, \quad (28)$$

where $\mathcal{P}_{[b, \bar{b}]}$ (\cdot) is the projection onto the interval $[b, \bar{b}]$ and

$$\xi = x - \frac{\lambda_z^k}{\beta_z} \in \mathcal{X}.$$

After knowing the solutions of the subproblems (23) and (25), we use the following alternative minimization procedure to solve (22); see Algorithm 4.

Algorithm 4 Augmented Lagrangian method for TV- L^2 model with box constraint – solve the minimization problem (22)

Initialization: $x^{k,0} = x^{k-1}$, $\begin{pmatrix} y^{k,0} \\ z^{k,0} \end{pmatrix} = \begin{pmatrix} y^{k-1} \\ z^{k-1} \end{pmatrix}$.

Iteration: For $l = 0, 1, 2, \dots, L - 1$:

- Compute $x^{k,l+1}$ by solving (24) for $\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} y^{k,l} \\ z^{k,l} \end{pmatrix}$;
- Compute $\begin{pmatrix} y^{k,l+1} \\ z^{k,l+1} \end{pmatrix}$ from (13) and (28) for $x = x^{k,l+1}$.

Output: $x^k = x^{k,L}$, $\begin{pmatrix} y^k \\ z^k \end{pmatrix} = \begin{pmatrix} y^{k,L} \\ z^{k,L} \end{pmatrix}$.

The convergence results of Algorithms 3 and 4 are similar to the convergence results proposed in the previous section, one can refer to Chan et al. (2013) for details.

Augmented Lagrangian Method for TV Restoration with Non-quadratic Fidelity

In this section, we review the augmented Lagrangian method proposed in Wu et al. (2011) for the TV restoration model with non-quadratic fidelity, which reads

$$\min_{x \in \mathcal{X}} \{E_{\text{TVNQ}}(x) = R(\nabla x) + F(Kx)\}. \quad (29)$$

where $R(\nabla x)$ is defined as in (3). Here, we consider the non-quadratic fidelity $F(Kx)$ which arises for removing non-Gaussian-type noises, such as impulsive noise and Poisson noise. For impulsive noise removal, we usually use the L^1 fidelity (Nikolova 2004)

$$F(Kx) = \alpha \|Kx - d\|_{L^1}, \quad (30)$$

and for Poisson noise removal, we commonly choose the Kullback-Leibler (KL) divergence fidelity (Le et al. 2007; Brune et al. 2009)

$$F(Kx) = \begin{cases} \alpha \sum_{1 \leq i, j \leq N} ((Kx)_{i,j} - d_{i,j} \log(Kx)_{i,j}), & (Kx)_{i,j} > 0, \forall i, j, \\ +\infty, & \text{otherwise.} \end{cases} \quad (31)$$

In this section, we focus on the augmented Lagrangian method for image restoration with these two non-quadratic fidelities. For other non-quadratic fidelities, one can extend our method accordingly.

The non-quadratic fidelities (30) and (31) are non-smooth. Adopting the idea to cope with total variation term, we require one more auxiliary variable to remove the nonlinearity arising from $F(Kx)$. We first introduce two auxiliary variables y and z and reformulate (29) to an equivalent constrained optimization problem

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}} \{ & G_{\text{TVNQ}}(y, z) = R(y) + F(z) \} \\ \text{s.t.} \quad & \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} \nabla \\ K \end{pmatrix} x. \end{aligned} \quad (32)$$

We then define the augmented Lagrangian function for (32) as

$$\begin{aligned} \mathcal{L}_{\text{TVNQ}}(x, y, z; \lambda_y, \lambda_z) = & R(y) + F(z) \\ & + \left\langle \begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ K \end{pmatrix} x \right\rangle + \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ K \end{pmatrix} x \right\|_{\mathcal{S}}^2 \end{aligned} \quad (33)$$

with Lagrange multiplier $\begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}$ and $\mathcal{S} = \begin{pmatrix} \beta_y \mathcal{I}_2 & \\ & \beta_z \mathcal{I}_1 \end{pmatrix}$ and consider the saddle-point problem

$$\begin{aligned} \text{Find } (x^*, y^*, z^*, \lambda_y^*, \lambda_z^*) \in & \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \mathcal{X} \times \mathcal{X}, \\ \text{s.t. } \mathcal{L}_{\text{TVNQ}}(x^*, y^*, z^*; \lambda_y, \lambda_z) \leq & \mathcal{L}_{\text{TVNQ}}(x^*, y^*, z^*; \lambda_y^*, \lambda_z^*) \\ \leq \mathcal{L}_{\text{TVNQ}}(x, y, z; \lambda_y^*, \lambda_z^*), & \\ \forall (x, y, z, \lambda_y, \lambda_z) \in & \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \mathcal{X} \times \mathcal{X}. \end{aligned} \quad (34)$$

Finally, we use the following iterative algorithm to solve the saddle-point problem (34); see Algorithm 5.

Algorithm 5 Augmented Lagrangian method for TV restoration with non-quadratic fidelity

Initialization: $x^{-1} = 0$, $\begin{pmatrix} y^{-1} \\ z^{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} \lambda_y^0 \\ \lambda_z^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Iteration: For $k = 0, 1, \dots$:

1. Compute (x^k, y^k, z^k) as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multipliers $\begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}$, i.e.,

$$(x^k, y^k, z^k) \approx \arg \min_{(x,y,z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}} \mathcal{L}_{\text{TVNQ}}(x, y, z; \lambda_y^k, \lambda_z^k), \quad (35)$$

where $\mathcal{L}_{\text{TVNQ}}(x, y, z; \lambda_z^k, \lambda_y^k)$ is as in (33).

2. Update

$$\begin{pmatrix} \lambda_y^{k+1} \\ \lambda_z^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix} + \begin{pmatrix} \beta_y (y^k - \nabla x^k) \\ \beta_z (z^k - K x^k) \end{pmatrix}.$$

We employ an alternating minimization procedure to solve the problem (35).

The Solution to Sub-problem w.r.t. x

Given $\begin{pmatrix} y \\ z \end{pmatrix}$, we have the subproblem of x as follows

$$\min_{x \in \mathcal{X}} \left\{ - \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} \nabla \\ K \end{pmatrix} x \right\rangle + \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ K \end{pmatrix} x \right\|_{\mathcal{S}}^2 \right\}, \quad (36)$$

which has the optimality condition

$$(\beta_z K^* K - \beta_y \Delta) x = K^* (\lambda_z^k + \beta_z z) - \text{div}(\lambda_y^k + \beta_y y). \quad (37)$$

We can use fast linear solvers to solve the above equation, such as FFT and CG.

The Solution to Sub-problem w.r.t. (y, z)

Given x , we have the subproblem of $\begin{pmatrix} y \\ z \end{pmatrix}$ as follows

$$\min_{(y,z) \in \mathcal{Y} \times \mathcal{Z}} \left\{ R(y) + F(z) + \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} \right\rangle + \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla \\ K \end{pmatrix} x \right\|_{\mathcal{S}}^2 \right\}. \quad (38)$$

We can split it into two distinct minimization problems with respect to y and z as follows

- y -subproblem:

$$\min_{y \in \mathcal{Y}} \left\{ R(y) + (\lambda_y^k, y) + \frac{\beta_y}{2} \|y - \nabla x\|^2 \right\}; \tag{39}$$

- z -subproblem:

$$\min_{z \in \mathcal{Z}} \left\{ F(z) + (\lambda_z^k, z) + \frac{\beta_z}{2} \|z - Kx\|^2 \right\}. \tag{40}$$

For the problem (39), it is the same as the problem (11) and can be solved via (13). For the problem (40), we next show its solution based on the choices of $F(\cdot)$.

For the L^1 fidelity (30), we can rewrite the z -subproblem (40) as

$$\min_{z \in \mathcal{Z}} \left\{ \alpha \|z - d\|_{L^1} + \frac{\beta_z}{2} \|z - \xi\|^2 \right\}$$

where

$$\xi = Kx - \frac{\lambda_z^k}{\beta_z}.$$

It has closed form solution (Wu et al. 2011)

$$z_{i,j} = d_{i,j} + \max \left(0, 1 - \frac{\alpha}{\beta_z |\xi_{i,j} - d_{i,j}|} \right) (\xi_{i,j} - d_{i,j}), \tag{41}$$

which is a one-dimensional case of (13). In this case, the alternating minimization procedure to solve the problem (35) is described in Algorithm 6.

For the KL divergence fidelity (31), we can rewrite the z -subproblem (40) as

$$\min_{\substack{z \in \mathcal{Z} \\ z_{i,j} > 0, \forall i,j}} \left\{ \alpha \sum_{1 \leq i,j \leq N} (z_{i,j} - d_{i,j} \log z_{i,j}) + \frac{\beta_z}{2} \sum_{1 \leq i,j \leq N} \left| z_{i,j} - \left(Kx - \frac{\lambda_z^k}{\beta_z} \right)_{i,j} \right|^2 \right\}.$$

It has closed form solution (Wu et al. 2011)

$$z_{i,j} = \frac{1}{2} \left(\sqrt{\left(\xi_{i,j} - \frac{\alpha}{\beta_z} \right)^2 + 4 \frac{\alpha}{\beta_z} d_{i,j}} + \left(\xi_{i,j} - \frac{\alpha}{\beta_z} \right) \right), \tag{42}$$

Algorithm 6 Augmented Lagrangian method for TV restoration with the L^1 fidelity – solve the minimization problem (35)

Initialization: $x^{k,0} = x^{k-1}, \begin{pmatrix} y^{k,0} \\ z^{k,0} \end{pmatrix} = \begin{pmatrix} y^{k-1} \\ z^{k-1} \end{pmatrix}$.

Iteration: For $l = 0, 1, 2, \dots, L - 1$:

- Compute $x^{k,l+1}$ by solving (37) for $\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} y^{k,l} \\ z^{k,l} \end{pmatrix}$;
- Compute $\begin{pmatrix} y^{k,l+1} \\ z^{k,l+1} \end{pmatrix}$ from (13) and (41) for $x = x^{k,l+1}$.

Output: $x^k = x^{k,L}, \begin{pmatrix} y^k \\ z^k \end{pmatrix} = \begin{pmatrix} y^{k,L} \\ z^{k,L} \end{pmatrix}$.

where

$$\xi = Kx - \frac{\lambda_z^k}{\beta_z}.$$

Now, the alternating minimization procedure to solve the problem (35) with the KL divergence fidelity (31) can be described in Algorithm 7.

Algorithm 7 Augmented Lagrangian method for TV restoration with the KL divergence fidelity – solve the minimization problem (35)

Initialization: $x^{k,0} = x^{k-1}, \begin{pmatrix} y^{k,0} \\ z^{k,0} \end{pmatrix} = \begin{pmatrix} y^{k-1} \\ z^{k-1} \end{pmatrix}$.

Iteration: For $l = 0, 1, 2, \dots, L - 1$:

- Compute $x^{k,l+1}$ from (37) for $\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} y^{k,l} \\ z^{k,l} \end{pmatrix}$;
- Compute $\begin{pmatrix} y^{k,l+1} \\ z^{k,l+1} \end{pmatrix}$ from (13) and (42) for $x = x^{k,l+1}$.

Output: $x^k = x^{k,L}, \begin{pmatrix} y^k \\ z^k \end{pmatrix} = \begin{pmatrix} y^{k,L} \\ z^{k,L} \end{pmatrix}$.

The convergence results of Algorithms 5, 6 and 7 are established in Wu et al. (2011), which are similar to convergence results presented previously for Algorithms 1 and 2.

Extension to Multichannel Image Restoration

In this section, we review the augmented Lagrangian method for the multichannel TV restoration (Wu and Tai 2010). The multichannel images are widely used, such as three-channel RGB color image.

The Multichannel TV Restoration Model

We denote an M -channel image by $\mathbf{x} = (x_1, x_2, \dots, x_M)$, where $x_m \in \mathcal{X}$, $\forall m = 1, 2, \dots, M$. We mention that, at each pixel (i, j) , the intensity of \mathbf{x} is vector-valued, i.e.,

$$\mathbf{x}_{i,j} = ((x_1)_{i,j}, (x_2)_{i,j}, \dots, (x_M)_{i,j}).$$

Let us define

$$\mathcal{X} = \underbrace{\mathcal{X} \times \mathcal{X} \times \dots \times \mathcal{X}}_M, \quad \mathcal{Y} = \underbrace{\mathcal{Y} \times \mathcal{Y} \times \dots \times \mathcal{Y}}_M.$$

Then we have $\mathbf{x} \in \mathcal{X}$ and

$$\nabla \mathbf{x} = (\nabla x_1, \nabla x_2, \dots, \nabla x_M) \in \mathcal{Y}.$$

The usual inner products and L^2 norms in the spaces \mathcal{X} and \mathcal{Y} are as follows. We denote

$$\begin{aligned} \langle \mathbf{x}, \mathbf{z} \rangle &= \sum_{1 \leq m \leq M} \langle x_m, z_m \rangle, \quad \|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}; \\ \langle \mathbf{y}, \mathbf{w} \rangle &= \sum_{1 \leq m \leq M} \langle y_m, w_m \rangle, \quad \|\mathbf{y}\| = \sqrt{\langle \mathbf{y}, \mathbf{y} \rangle}. \end{aligned}$$

for $\mathbf{x}, \mathbf{z} \in \mathcal{X}$ and $\mathbf{y}, \mathbf{w} \in \mathcal{Y}$. At each pixel (i, j) , we also define the following pixel-by-pixel norms

$$|\mathbf{x}_{i,j}| = \sqrt{\sum_{1 \leq m \leq M} (x_m)_{i,j}^2} \quad \text{and} \quad |\mathbf{y}_{i,j}| = \sqrt{\sum_{1 \leq m \leq M} |(y_m)_{i,j}|^2}.$$

for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$.

With reference to the degradation model (1) of the gray image, here we model the multichannel image degradation procedure as

$$\underline{\mathbf{x}} \xrightarrow{\text{linear transformation}} \mathbf{K} \underline{\mathbf{x}} \xrightarrow{\text{noise}} \mathbf{d},$$

where $\mathbf{d} \in \mathcal{X}$ is an observed image and $\mathbf{K} : \mathcal{X} \rightarrow \mathcal{X}$ is linear operator like a blur. Here the noise could be also Gaussian, impulsive, Poisson, or even others.

In this survey, we consider \mathbf{K} as the blur operator and the noise is Gaussian type. The operator \mathbf{K} has the form of

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1M} \\ K_{21} & K_{22} & \cdots & K_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ K_{M1} & K_{M2} & \cdots & K_{MM} \end{pmatrix},$$

where each K_{ij} is a convolution matrix. The diagonal elements of \mathbf{K} denote within-channel blurs, while the off-diagonal elements describe cross-channel blurs. To solve $\underline{\mathbf{x}}$, we consider the following multichannel image restoration model (Sapiro and Ringach 1996)

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ E_{\text{MTV}}(\mathbf{x}) = \frac{\alpha}{2} \|\mathbf{K}\mathbf{x} - \mathbf{d}\|^2 + R_{\text{MTV}}(\nabla\mathbf{x}) \right\}, \tag{43}$$

where

$$R_{\text{MTV}}(\nabla\mathbf{x}) = \text{TV}(\mathbf{x}) = \sum_{1 \leq i, j \leq N} \sqrt{\sum_{1 \leq m \leq M} |(\nabla x_m)_{i,j}|^2}$$

is the vectorial TV semi-norm (Sapiro and Ringach 1996) (see Blomgren and Chan 1998 for some other choices).

Similarly as for the single channel image restoration model, here we make the following assumption:

- $\text{Null}(\nabla) \cap \text{Null}(\mathbf{K}) = \{0\}$.

Under this assumption, one can verify that the functional $E_{\text{MTV}}(\mathbf{x})$ in (43) is convex, proper, coercive, and continuous. Hence, we have the following result (Wu and Tai 2010).

Theorem 6. *The problem (43) has at least one solution \mathbf{x} , which satisfies*

$$0 \in \alpha \mathbf{K}^*(\mathbf{K}\mathbf{x} - \mathbf{d}) - \text{div } \partial R_{\text{MTV}}(\nabla\mathbf{x}),$$

where $\partial R_{\text{MTV}}(\nabla\mathbf{x})$ is the subdifferential of R_{MTV} at $\nabla\mathbf{x}$. Moreover, if $\text{Null}(\mathbf{K}) = \{0\}$, the minimizer is unique.

Augmented Lagrangian Method for Multichannel TV Restoration

By introducing a new variable $\mathbf{y} = (y_1, y_2, \dots, y_M) \in \mathcal{Y}$, we first reformulate the minimization problem (43) to the following equivalent constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} & \left\{ G_{\text{MTV}}(\mathbf{x}, \mathbf{y}) = \frac{\alpha}{2} \|\mathbf{K}\mathbf{x} - \mathbf{d}\|^2 + R_{\text{MTV}}(\mathbf{y}) \right\} \\ \text{s.t.} & \quad \mathbf{y} = \nabla \mathbf{x}. \end{aligned} \quad (44)$$

We then define the augmented Lagrangian function as

$$\mathcal{L}_{\text{MTV}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}) = \frac{\alpha}{2} \|\mathbf{K}\mathbf{x} - \mathbf{d}\|^2 + R_{\text{MTV}}(\mathbf{y}) + \langle \boldsymbol{\lambda}, \mathbf{y} - \nabla \mathbf{x} \rangle + \frac{\beta}{2} \|\mathbf{y} - \nabla \mathbf{x}\|^2,$$

with the multiplier $\boldsymbol{\lambda} \in \mathcal{Y}$ and a positive constant β . The augmented Lagrangian method aims at seeking a saddle-point of the following problem:

$$\begin{aligned} \text{Find} & \quad (\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \\ \text{s.t.} & \quad \mathcal{L}_{\text{MTV}}(\mathbf{x}^*, \mathbf{y}^*; \boldsymbol{\lambda}) \leq \mathcal{L}_{\text{MTV}}(\mathbf{x}^*, \mathbf{y}^*; \boldsymbol{\lambda}^*) \leq \mathcal{L}_{\text{MTV}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}^*) \\ & \quad \forall (\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}. \end{aligned} \quad (45)$$

Finally, an iterative procedure to solve the problem (45) is described in Algorithm 8.

Algorithm 8 Augmented Lagrangian method for the multichannel TV model

Initialization: $\mathbf{x}^{-1} = 0, \mathbf{y}^{-1} = 0, \boldsymbol{\lambda}^0 = 0$.

Iteration: For $k = 0, 1, 2, \dots$:

1. Compute $(\mathbf{x}^k, \mathbf{y}^k)$ from

$$(\mathbf{x}^k, \mathbf{y}^k) \approx \arg \min_{(\mathbf{x}, \mathbf{y}) \in (\mathcal{X}, \mathcal{Y})} \mathcal{L}_{\text{MTV}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\lambda}^k). \quad (46)$$

2. Update

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta(\mathbf{y}^k - \nabla \mathbf{x}^k).$$

As for the minimization problem (46), we separate it into two subproblems with respect to \mathbf{x} and \mathbf{y} and minimize them alternatively.

The Solution to Sub-problem w.r.t. \mathbf{x}

For a given \mathbf{y} , there is the following minimization problem of variable \mathbf{x}

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \frac{\alpha}{2} \|\mathbf{K}\mathbf{x} - \mathbf{d}\|^2 - \langle \boldsymbol{\lambda}^k, \nabla \mathbf{x} \rangle + \frac{\beta}{2} \|\mathbf{y} - \nabla \mathbf{x}\|^2 \right\}. \quad (47)$$

Applying Fourier transforms to the optimality condition of the problem (47), we obtain

$$[\alpha \mathcal{F}(\mathbf{K}^*) \mathcal{F}(\mathbf{K}) - \beta \mathcal{F}(\Delta)] \mathcal{F}(\mathbf{x}) = \alpha \mathcal{F}(\mathbf{K}^*) \mathcal{F}(\mathbf{d}) - \mathcal{F}(\text{div}) \mathcal{F}(\boldsymbol{\lambda}^k + \beta \mathbf{y}), \quad (48)$$

from which $\mathcal{F}(\mathbf{x})$ can be found and then \mathbf{x} via an inverse Fourier transform (Yang et al. 2009; Wu and Tai 2010). Here applying Fourier transform to a block matrix is regarded as applying Fourier transform to each block.

The Solution to Sub-problem w.r.t. \mathbf{y}

For a given \mathbf{x} , there is the following minimization problem of variable \mathbf{y}

$$\min_{\mathbf{y} \in \mathcal{Y}} \{ R_{\text{MTV}}(\mathbf{y}) + \langle \boldsymbol{\lambda}^k, \mathbf{y} \rangle + \frac{\beta}{2} \|\mathbf{y} - \nabla \mathbf{x}\|^2 \}.$$

It has the following closed form solution (Yang et al. 2009; Wu and Tai 2010)

$$y_{i,j} = \max \left(1 - \frac{1}{\beta |\eta_{i,j}|}, 0 \right) \eta_{i,j}, \quad (49)$$

where $\boldsymbol{\eta} = \nabla \mathbf{x} - \frac{\boldsymbol{\lambda}^k}{\beta} \in \mathcal{Y}$. Indeed, this solution is a high-dimensional version of (13), which can be also derived from the geometric method.

According to (48) and (49), we then have an alternating minimization procedure to (46); see Algorithm 9.

Algorithm 9 Augmented Lagrangian method for the multichannel TV model – solve the minimization problem (46)

Initialization: $\mathbf{x}^{k,0} = \mathbf{x}^{k-1}$, $\mathbf{y}^{k,0} = \mathbf{y}^{k-1}$.

Iteration: For $l = 0, 1, 2, \dots, L - 1$:

- Compute $\mathbf{x}^{k,l+1}$ from (48) for $\mathbf{y} = \mathbf{y}^{k,l}$;
- Compute $\mathbf{y}^{k,l+1}$ from (49) for $\mathbf{x} = \mathbf{x}^{k,l+1}$.

Output: $\mathbf{x}^k = \mathbf{x}^{k,L}$, $\mathbf{y}^k = \mathbf{y}^{k,L}$.

We remark that the convergence results of Algorithms 3 and 4 can be directly extended for the Algorithms 8 and 9 (Wu and Tai 2010) and we omit the details.

Extension to High-Order Models

In this section, we review augmented Lagrangian methods for some high-order models, including the second-order total variation model (Lysaker et al. 2003), the total generalized variation model (Bredies et al. 2010), the Euler's elastic-based model (Chan et al. 2002; Tai et al. 2011), and the mean curvature model (Zhu and Chan 2012; Zhu et al. 2013).

Augmented Lagrangian Method for Second-Order Total Variation Model

To overcome the staircase effect, Lysaker, Lundervold, and Tai suggested regularizing the total variation of the gradient and proposed a model based on second-order derivatives (Lysaker et al. 2003). We begin with some notations to establish this second-order total variation (TV^2) model.

Let

$$\widehat{\mathcal{Y}} = \mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X}.$$

We define the discrete Hessian operator

$$\begin{aligned} H : \mathcal{X} &\rightarrow \widehat{\mathcal{Y}} \\ x &\rightarrow Hx, \end{aligned}$$

with

$$(Hx)_{i,j} = \begin{pmatrix} (\mathring{D}_{11}^{-+}x)_{i,j} & (\mathring{D}_{12}^{++}x)_{i,j} \\ (\mathring{D}_{21}^{++}x)_{i,j} & (\mathring{D}_{22}^{-+}x)_{i,j} \end{pmatrix},$$

where \mathring{D}_{11}^{-+} , \mathring{D}_{12}^{++} , \mathring{D}_{21}^{++} and \mathring{D}_{22}^{-+} are second-order difference operators and given by

$$\begin{aligned} (\mathring{D}_{11}^{-+}x)_{i,j} &:= (\mathring{D}_1^- (\mathring{D}_1^+ x))_{i,j}, \\ (\mathring{D}_{12}^{++}x)_{i,j} &:= (\mathring{D}_1^+ (\mathring{D}_2^+ x))_{i,j}, \\ (\mathring{D}_{21}^{++}x)_{i,j} &:= (\mathring{D}_2^+ (\mathring{D}_1^+ x))_{i,j}, \\ (\mathring{D}_{22}^{-+}x)_{i,j} &:= (\mathring{D}_2^- (\mathring{D}_2^+ x))_{i,j}. \end{aligned}$$

The usual inner product and L^2 norm in the space $\widehat{\mathcal{Y}}$ are as follows. We denote

$$\langle y, w \rangle = \langle y^1, w^1 \rangle + \langle y^2, w^2 \rangle + \langle y^3, w^3 \rangle + \langle y^4, w^4 \rangle \text{ and } \|y\| = \sqrt{\langle y, y \rangle},$$

for $y = \begin{pmatrix} y^1 & y^2 \\ y^3 & y^4 \end{pmatrix} \in \widehat{\mathcal{Y}}$ and $w = \begin{pmatrix} w^1 & w^2 \\ w^3 & w^4 \end{pmatrix} \in \widehat{\mathcal{W}}$. At each pixel (i, j) ,

$$|y_{i,j}| = \sqrt{(y^1)_{i,j}^2 + (y^2)_{i,j}^2 + (y^3)_{i,j}^2 + (y^4)_{i,j}^2}$$

is the usual Euclidean norm in \mathbb{R}^4 . By using the inner products of $\widehat{\mathcal{Y}}$ and \mathcal{X} and the definitions of the finite difference operators, the adjoint operator of H is as follows

$$H^* : \widehat{\mathcal{Y}} \rightarrow \mathcal{X}$$

$$y = \begin{pmatrix} y^1 & y^2 \\ y^3 & y^4 \end{pmatrix} \rightarrow H^*y,$$

where

$$(H^*y)_{i,j} = (\mathring{D}_{11}^{+-}y^1)_{i,j} + (\mathring{D}_{21}^{-}y^1)_{i,j} + (\mathring{D}_{12}^{-}y^3)_{i,j} + (\mathring{D}_{22}^{+-}y^4)_{i,j},$$

where \mathring{D}_{11}^{+-} , \mathring{D}_{12}^{-} , \mathring{D}_{21}^{-} , and \mathring{D}_{22}^{+-} are second-order difference operators.

By regularizing the norm of the discrete Hessian, the TV² model (Lysaker et al. 2003) reads

$$\min_{x \in \mathcal{X}} \left\{ E_{\text{TV}^2}(x) = \frac{\alpha}{2} \|Kx - d\|^2 + R_{\text{HO}}(Hx) \right\}, \tag{50}$$

where $\alpha > 0$, $d \in \mathcal{X}$ is the observed image, $K : \mathcal{X} \rightarrow \mathcal{X}$ is the blur operator and

$$R_{\text{HO}}(Hx) = \sum_{1 \leq i, j \leq N} |(Hx)_{i,j}|. \tag{51}$$

Similarly as for the total variation restoration model, we make the following assumption:

- $\text{Null}(H) \cap \text{Null}(K) = \{0\}$.

Under this assumption, the functional $E_{\text{TV}^2}(x)$ in (50) is convex, proper, coercive, and continuous. Hence, we have the following result.

Theorem 7. *The problem (50) has at least one solution x , which satisfies*

$$0 \in \alpha K^*(Kx - d) + H^* \partial R_{\text{HO}}(Hx),$$

where $\partial R_{\text{HO}}(Hx)$ is the subdifferential of R_{HO} at Hx . Moreover, if $\text{Null}(K) = \{0\}$, the minimizer is unique.

In the following we review the augmented Lagrangian method proposed in Wu and Tai (2010) to solve (50). We first introduce a new variable $\hat{y} \in \widehat{\mathcal{Y}}$ and reformulate (50) into a constrained optimization problem

$$\begin{aligned} \min_{x \in \mathcal{X}, \hat{y} \in \widehat{\mathcal{Y}}} & \left\{ G_{\text{TV}^2}(x, \hat{y}) = \frac{\alpha}{2} \|Kx - d\|^2 + R_{\text{HO}}(\hat{y}) \right\} \\ \text{s.t.} & \quad \hat{y} = Hx. \end{aligned} \quad (52)$$

To solve (52), we define the augmented Lagrangian functional as

$$\mathcal{L}_{\text{TV}^2}(x, \hat{y}; \lambda) = \frac{\alpha}{2} \|Kx - d\|^2 + R_{\text{HO}}(\hat{y}) + \langle \lambda, \hat{y} - Hx \rangle + \frac{\beta}{2} \|\hat{y} - Hx\|^2, \quad (53)$$

with the multiplier $\lambda \in \widehat{\mathcal{Y}}$ and a positive constant β , and consider the following saddle-point problem:

$$\begin{aligned} \text{Find } (x^*, \hat{y}^*, \lambda^*) & \in \mathcal{X} \times \widehat{\mathcal{Y}} \times \widehat{\mathcal{Y}} \\ \text{s.t. } \mathcal{L}_{\text{TV}^2}(x^*, \hat{y}^*; \lambda) & \leq \mathcal{L}_{\text{TV}^2}(x^*, \hat{y}^*; \lambda^*) \leq \mathcal{L}_{\text{TV}^2}(x, \hat{y}; \lambda^*) \\ \forall (x, \hat{y}; \lambda) & \in \mathcal{X} \times \widehat{\mathcal{Y}} \times \widehat{\mathcal{Y}}. \end{aligned} \quad (54)$$

We employ an iterative procedure to solve the saddle-point problem (54), which is described as Algorithm 10.

Algorithm 10 Augmented Lagrangian method for the TV² model

Initialization: $x^{-1} = 0, \hat{y}^{-1} = 0, \lambda^0 = 0$.

Iteration: For $k = 0, 1, 2, \dots$:

1. Compute (x^k, \hat{y}^k) from

$$(x^k, \hat{y}^k) \approx \arg \min_{(x, \hat{y}) \in (\mathcal{X}, \widehat{\mathcal{Y}})} \mathcal{L}_{\text{TV}^2}(x, \hat{y}; \lambda^k). \quad (55)$$

2. Update

$$\lambda^{k+1} = \lambda^k + \beta(\hat{y}^k - Hx^k).$$

The Solution to Sub-problem w.r.t. x

Given y , we are going to solve the following minimization problem

$$\min_{x \in \mathcal{X}} \left\{ \frac{\alpha}{2} \|Kx - d\|^2 - \langle \lambda^k, Hx \rangle + \frac{\beta}{2} \|\hat{y} - Hx\|^2 \right\}, \quad (56)$$

the first-order optimality condition of which gives us a linear equation as follows

$$(\alpha K^* K + \beta H^* H)x = \alpha K^* d + H^*(\lambda^k + \beta \hat{y}). \quad (57)$$

This equation can be solved by well-developed linear solvers such as FFT and CG.

The Solution to Sub-problem w.r.t. \hat{y}

Given x , we are going to solve the following minimization problem

$$\min_{\hat{y} \in \widehat{\mathcal{Y}}} \left\{ R_{\text{HO}}(\hat{y}) + (\lambda^k, \hat{y}) + \frac{\beta}{2} \|\hat{y} - Hx\|^2 \right\}, \quad (58)$$

the closed form solution of which is

$$\hat{y}_{i,j} = \max \left(0, 1 - \frac{1}{\beta |\eta_{i,j}|} \right) \eta_{i,j}, \quad (59)$$

where $\eta = Hx - \frac{\lambda^k}{\beta} \in \widehat{\mathcal{Y}}$. We mention that the solution (59) is a high-dimensional version of (13), which can be also derived from the geometric method.

According to (57) and (59), we then use an iterative procedure to alternatively calculate x and \hat{y} ; see Algorithm 11.

Algorithm 11 Augmented Lagrangian method for the TV^2 model – solve the minimization problem (55)

Initialization: $x^{k,0} = x^{k-1}$, $\hat{y}^{k,0} = \hat{y}^{k-1}$.

Iteration: For $l = 0, 1, 2, \dots, L - 1$:

- Compute $x^{k,l+1}$ by solving (57) for $\hat{y} = \hat{y}^{k,l}$;
- Compute $\hat{y}^{k,l+1}$ from (59) for $x = x^{k,l+1}$.

Output: $x^k = x^{k,L}$, $\hat{y}^k = \hat{y}^{k,L}$.

We mention that the convergence results of the augmented Lagrangian method for the TV^2 model are straightforward as in Wu and Tai (2010) and we omit the details.

Augmented Lagrangian Method for Total Generalized Variation Model

Total generalized variation (TGV) is a very successful generalization of total variation, which involves high-order derivatives to reduce staircase effect (Bredies

et al. 2010). In this section, we consider the following discrete second-order total generalized variation (Bredies et al. 2010)-based image restoration model

$$\min_{x \in \mathcal{X}, w \in \mathcal{Y}} \left\{ \frac{1}{2} \|Kx - d\|^2 + \alpha_1 R(\nabla x - w) + \alpha_0 R_{\text{HO}}(\mathcal{E}w) \right\}, \tag{60}$$

where $R(\nabla x - w)$ is defined by replacing ∇x by $\nabla x - w$ in (3), \mathcal{E} denotes a distributional symmetrized gradient operator

$$\begin{aligned} \mathcal{E}: \mathcal{Y} &\rightarrow \widehat{\mathcal{Y}} \\ w = (w^1, w^2) &\rightarrow \mathcal{E}w = \frac{1}{2}(\nabla w + \nabla w^T), \end{aligned}$$

with

$$\begin{aligned} (\mathcal{E}w)_{ij} &= \frac{1}{2}(\nabla w + \nabla w^T)_{ij} \\ &= \begin{pmatrix} (\mathring{D}_1^+ w^1)_{ij} & \frac{1}{2}((\mathring{D}_2^+ w^1)_{ij} + (\mathring{D}_1^+ w^2)_{ij}) \\ \frac{1}{2}((\mathring{D}_2^+ w^1)_{ij} + (\mathring{D}_1^+ w^2)_{ij}) & (\mathring{D}_2^+ w^2)_{ij} \end{pmatrix}, \end{aligned}$$

and $R_{\text{HO}}(\cdot)$ is defined in (51). Similarly, by using the inner products of $\widehat{\mathcal{Y}}$ and \mathcal{Y} and the definitions of the finite difference operators the adjoint operator of $-\mathcal{E}$ is as follows

$$\begin{aligned} \text{div}_2: \widehat{\mathcal{Y}} &\rightarrow \mathcal{Y} \\ z = \begin{pmatrix} z^1 & z^3 \\ z^3 & z^2 \end{pmatrix} &\rightarrow \text{div}_2 z, \end{aligned}$$

where

$$\text{div}_2 z = \begin{pmatrix} \mathring{D}_1^- z^1 + \mathring{D}_2^- z^3 \\ \mathring{D}_1^- z^3 + \mathring{D}_2^- z^2 \end{pmatrix}$$

with

$$(\text{div}_2 z)_{ij} = \begin{pmatrix} (\mathring{D}_1^- z^1)_{ij} + (\mathring{D}_2^- z^3)_{ij} \\ (\mathring{D}_1^- z^3)_{ij} + (\mathring{D}_2^- z^2)_{ij} \end{pmatrix}.$$

Augmented Lagrangian-based methods for total generalized variation-related models can be found in Gao et al. (2018). Here, we propose the augmented Lagrangian method to solve (60). We first introduce two auxiliary variable

$y = (y^1, y^2) \in \mathcal{Y}$ and $z = \begin{pmatrix} z^1 & z^3 \\ z^3 & z^2 \end{pmatrix} \in \widehat{\mathcal{Z}}$ and transform it into an equivalent constrained optimization problem

$$\begin{aligned} \min_{x \in \mathcal{X}, w \in \mathcal{W}, y \in \mathcal{Y}, z \in \widehat{\mathcal{Z}}} & \left\{ G_{\text{TGV}}(x, y, z) = \frac{1}{2} \|Kx - d\|^2 + \alpha_1 R(y) + \alpha_0 R_{\text{HO}}(z) \right\} \\ \text{s.t.} & \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix}. \end{aligned} \tag{61}$$

We then define the augmented Lagrangian function as follows

$$\begin{aligned} \mathcal{L}_{\text{TGV}}(x, w, y, z; \lambda_y, \lambda_z) &= \frac{1}{2} \|Kx - d\|^2 + \alpha_1 R(y) + \alpha_0 R_{\text{HO}}(z) \\ &+ \left\langle \begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \right\rangle \\ &+ \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \right\|_{\mathcal{S}}^2, \end{aligned} \tag{62}$$

where $\begin{pmatrix} \lambda_y \\ \lambda_z \end{pmatrix}$ is the Lagrange multiplier and $\mathcal{S} = \begin{pmatrix} \beta_y \mathcal{I}_2 & \\ & \beta_z \widehat{\mathcal{I}}_2 \end{pmatrix}$ with the identity operator $\widehat{\mathcal{I}}_2 : \widehat{\mathcal{Z}} \rightarrow \widehat{\mathcal{Z}}$, and consider the saddle-point problem

$$\begin{aligned} \text{Find } & (x^*, w^*, y^*, z^*, \lambda_y^*, \lambda_z^*) \in \mathcal{X} \times \mathcal{W} \times \mathcal{Y} \times \widehat{\mathcal{Z}} \times \mathcal{Y} \times \widehat{\mathcal{Z}} \\ \text{s.t. } & \mathcal{L}_{\text{TGV}}(x^*, w^*, y^*, z^*; \lambda_y, \lambda_z) \\ & \leq \mathcal{L}_{\text{TGV}}(x^*, w^*, y^*, z^*; \lambda_y^*, \lambda_z^*) \\ & \leq \mathcal{L}_{\text{TGV}}(x, w, y, z; \lambda_y^*, \lambda_z^*), \\ & \forall (x, w, y, z, \lambda_y, \lambda_z) \in \mathcal{X} \times \mathcal{W} \times \mathcal{Y} \times \widehat{\mathcal{Z}} \times \mathcal{Y} \times \widehat{\mathcal{Z}}. \end{aligned} \tag{63}$$

Finally, the iterative algorithm for seeking a saddle point is given by Algorithm 12.

The Solution to Sub-problem w.r.t. (x, w)

Given $\begin{pmatrix} y \\ z \end{pmatrix}$, we concern with the following minimization problem

Algorithm 12 Augmented Lagrangian method for TGV model

Initialization: $\begin{pmatrix} x^{-1} \\ w^{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} y^{-1} \\ z^{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} \lambda_y^0 \\ \lambda_z^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Iteration: For $k = 0, 1, \dots$:

1. Compute (x^k, w^k, y^k, z^k) from $\begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}$, i.e.,

$$(x^k, w^k, y^k, z^k) \approx \arg \min_{(x, w, y, z) \in \mathcal{X} \times \mathcal{W} \times \mathcal{Y} \times \mathcal{Z}} \mathcal{L}_{\text{TGV}}(x, w, y, z; \lambda_y^k, \lambda_z^k). \quad (64)$$

2. Update

$$\begin{pmatrix} \lambda_y^{k+1} \\ \lambda_z^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix} + \begin{pmatrix} \beta_y(y^k - \nabla x^k + w^k) \\ \beta_z(z^k - \mathcal{E}w^k) \end{pmatrix}.$$

$$\min_{(x, w) \in \mathcal{X} \times \mathcal{W}} \left\{ \frac{1}{2} \|Kx - d\|^2 - \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \right\rangle + \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \right\|_{\mathcal{S}}^2 \right\}. \quad (65)$$

This problem is a quadratic optimization problem, whose optimality condition gives a linear system equations

$$\begin{pmatrix} K^*K - \beta_y \Delta & \beta_y \operatorname{div} \\ -\beta_y \nabla & \beta_y - \beta_z \operatorname{div}_2 \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} K^*d - \operatorname{div}(\lambda_y^k + \beta_y y) \\ -\lambda_y^k - \beta_y y - \operatorname{div}_2(\lambda_z^k + \beta_z z) \end{pmatrix},$$

i.e.

$$\begin{cases} (K^*K - \beta_y \mathring{D}_1^- \mathring{D}_1^+ - \beta_y \mathring{D}_2^- \mathring{D}_2^+)x + \beta_y \mathring{D}_1^- w^1 + \beta_y \mathring{D}_2^- w^2 = g^1, \\ -\beta_y \mathring{D}_1^+ x + (\beta_y \mathcal{I} - \beta_z \mathring{D}_1^- \mathring{D}_1^+ - \frac{\beta_z}{2} \mathring{D}_2^- \mathring{D}_2^+)w^1 - \frac{\beta_z}{2} \mathring{D}_2^- \mathring{D}_1^+ w^2 = g^2, \\ -\beta_y \mathring{D}_2^+ x - \frac{\beta_z}{2} \mathring{D}_1^- \mathring{D}_2^+ w^1 + (\beta_y \mathcal{I} - \frac{\beta_z}{2} \mathring{D}_1^- \mathring{D}_1^+ - \beta_z \mathring{D}_2^- \mathring{D}_2^+)w^2 = g^3, \end{cases} \quad (66)$$

where

$$\begin{aligned} g^1 &= K^*d - \mathring{D}_1^-((\lambda_y^k)^1 + \beta_y y^1) - \mathring{D}_2^-((\lambda_z^k)^2 + \beta_y y^2), \\ g^2 &= -(\lambda_y^k)^1 - \beta_y y^1 - \mathring{D}_1^-((\lambda_z^k)^1 + \beta_z z^1) - \mathring{D}_2^-((\lambda_z^k)^3 + \beta_z z^3), \\ g^3 &= -(\lambda_y^k)^2 - \beta_y y^2 - \mathring{D}_1^-((\lambda_z^k)^3 + \beta_z z^3) - \mathring{D}_2^-((\lambda_z^k)^2 + \beta_z z^2). \end{aligned}$$

This linear system with periodic boundary condition can be efficiently solved by Fourier transform via FFT implementation (Yang et al. 2009). Firstly, we apply FFTs to both sides of (66) to get

$$\begin{pmatrix} a^{11} & a^{12} & a^{13} \\ a^{21} & a^{22} & a^{23} \\ a^{31} & a^{32} & a^{33} \end{pmatrix} \begin{pmatrix} \mathcal{F}(x) \\ \mathcal{F}(w^1) \\ \mathcal{F}(w^2) \end{pmatrix} = \begin{pmatrix} \mathcal{F}(g^1) \\ \mathcal{F}(g^2) \\ \mathcal{F}(g^3) \end{pmatrix}. \tag{67}$$

where a^{ij} , ($i, j = 1, \dots, 3$) are Fourier coefficients of the operators in the left side of (66). Secondly, we solve the resulting systems by block Gaussian elimination method for $\mathcal{F}(x)$, $\mathcal{F}(w^1)$ and $\mathcal{F}(w^2)$. Finally, we apply inverse FFTs to obtain x and $w = (w^1, w^2)$.

The Solution to Sub-problem w.r.t. (y, z)

Given $\begin{pmatrix} x \\ w \end{pmatrix}$, we concern with the following minimization problem

$$\min_{(y,z) \in \mathcal{Y} \times \mathcal{Z}} \left\{ \alpha_1 R(y) + \alpha_0 R_{HO}(z) + \left\langle \begin{pmatrix} \lambda_y^k \\ \lambda_z^k \end{pmatrix}, \begin{pmatrix} y \\ z \end{pmatrix} \right\rangle + \frac{1}{2} \left\| \begin{pmatrix} y \\ z \end{pmatrix} - \begin{pmatrix} \nabla - \mathcal{I}_2 \\ \mathcal{E} \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \right\|_{\mathcal{S}}^2 \right\}. \tag{68}$$

It can be separated into two independent minimization problems:

- y -subproblem:

$$\min_{y \in \mathcal{Y}} \left\{ \alpha_1 R(y) + \langle \lambda_y^k, y \rangle + \frac{\beta_y}{2} \|y - \nabla x + w\|^2 \right\}; \tag{69}$$

- z -subproblem:

$$\min_{z \in \mathcal{Z}} \left\{ \alpha_0 R_{HO}(z) + \langle \lambda_z^k, z \rangle + \frac{\beta_z}{2} \|z - \mathcal{E}w\|^2 \right\}. \tag{70}$$

The problem (69) and (70) have the closed form solutions

$$y_{i,j} = \max \left(0, 1 - \frac{\alpha_1}{\beta_y |\eta_{i,j}|} \right) \eta_{i,j}, \text{ and } z_{i,j} = \max \left(0, 1 - \frac{\alpha_0}{\beta_z |\xi_{i,j}|} \right) \xi_{i,j}, \tag{71}$$

where

$$\eta = \nabla x - w - \frac{\lambda_y^k}{\beta_y} \in \mathcal{Y}, \text{ and } \xi = \mathcal{E}w - \frac{\lambda_z^k}{\beta_z} \in \widehat{\mathcal{Y}}.$$

After knowing the solutions of the subproblems (65) and (68), we use the following alternative minimization procedure to solve (64); see Algorithm 13.

Algorithm 13 Augmented Lagrangian method for TGV model–solve the minimization problem (64)

Initialization: $\begin{pmatrix} x^{k,0} \\ w^{k,0} \end{pmatrix} = \begin{pmatrix} x^{k-1} \\ w^{k-1} \end{pmatrix}, \begin{pmatrix} y^{k,0} \\ z^{k,0} \end{pmatrix} = \begin{pmatrix} y^{k-1} \\ z^{k-1} \end{pmatrix}.$

Iteration: For $l = 0, 1, \dots, L - 1$:

- Compute $\begin{pmatrix} x^{k,l+1} \\ w^{k,l+1} \end{pmatrix}$ from (67) for $\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} y^{k,l} \\ z^{k,l} \end{pmatrix}$;
- Compute $\begin{pmatrix} y^{k,l+1} \\ z^{k,l+1} \end{pmatrix}$ from (71) for $\begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} x^{k,l+1} \\ w^{k,l+1} \end{pmatrix}.$

Output: $\begin{pmatrix} x^k \\ w^k \end{pmatrix} = \begin{pmatrix} x^{k,L} \\ w^{k,L} \end{pmatrix}, \begin{pmatrix} y^k \\ z^k \end{pmatrix} = \begin{pmatrix} y^{k,L} \\ z^{k,L} \end{pmatrix}.$

Augmented Lagrangian Method for Euler Elastic-Based Model

As basic geometric measurements of curves, both length and curvatures are natural regularities that are widely used in various image processing problems. Euler's elastica is defined as the line energy for a smooth planar curves γ

$$E(\gamma) = \int_{\gamma} (a + b\kappa^2) ds, \quad (72)$$

where κ is the curvature of the curve, s is arc length, and a, b are positive constants. By summing up the Euler's elastica energies of all the level sets for an image x , it gives the following energy for image denoising task

$$\min_x R_{EE}(\kappa(x), \nabla x) + \frac{1}{2} \|Kx - d\|^2, \quad (73)$$

where $\kappa(x) = \operatorname{div}\left(\frac{\nabla x}{|\nabla x|}\right)$ and $R_{EE}(\kappa(x), \nabla x)$ is defined by

$$R_{EE}(\kappa(x), \nabla x) = \sum_{1 \leq i, j \leq N} \left(a + b\kappa^2(x_{i,j}) \right) |(\nabla x)_{i,j}|.$$

Euler's elastica regularization has lots of applications in shape and image processing. However, the non-convexity, the non-smoothness, and the nonlinearity of the Euler's elastica energy make its minimization a challenging task. Chan et al. (2002) developed a computational scheme based on numerical PDEs for inpainting problem. Bae et al. (2010) presented an efficient minimization algorithm based on graph cuts for minimizing the Euler's elastica energy. Tai et al. (2011) proposed an augmented Lagrangian method based on the operator-splitting and relaxation techniques, which greatly improved the efficiency of the Euler's elastica model. Since then, operator-splitting and augmented Lagrangian method have been extensively studied for Euler's elastica (Duan et al. 2013; Yashtini and Kang 2016). Recent advances include functional lifting to get a convex, lower semi-continuous, coercive approximation of the Euler's elastica energy (Bredies et al. 2015), and a lie operator-splitting-based time discretization scheme (Deng et al. 2019). In Tai et al. (2011), Euler's elastica regularized model (73) is reformulated as the following constrained minimization problem

$$\begin{aligned} \min_{x,y,n,m} R_{EE}(\operatorname{div} n, y) + \frac{1}{2} \|Kx - d\|^2 + I_{\mathcal{M}}(m) \\ \text{s.t. } y = \nabla x, n = m, |y| = m \cdot y, \end{aligned} \quad (74)$$

where $I_{\mathcal{M}}(\cdot)$ is an indicator function of the set

$$\mathcal{M} = \{m_{ij} : |m_{i,j}| \leq 1, \forall 1 \leq i, j \leq N\}.$$

Note that the variable m was introduced to relax the constraint on variable n . By requiring m to be lain in the set \mathcal{M} , the term $|y| - y \cdot m$ is guaranteed non-negative, which make the sub-minimization problem w.r.t. m easy to handle with. We can further define the augmented Lagrangian functional as follows

$$\begin{aligned} \mathcal{L}_{EE}(x, y, n, m; \lambda_y, \lambda_n, \lambda_m) = R_{EE}(\operatorname{div} n, y) + \frac{1}{2} \|Kx - d\|^2 + I_{\mathcal{M}}(m) \\ + \langle \lambda_y, y - \nabla x \rangle + \frac{\beta_y}{2} \|y - \nabla x\|^2 + \langle \lambda_n, n - m \rangle + \frac{\beta_n}{2} \|n - m\|^2 \\ + \langle \lambda_m, |y| - m \cdot y \rangle + \langle |y| - m \cdot y, \beta_m \rangle, \end{aligned} \quad (75)$$

where $\lambda_y, \lambda_n, \lambda_m$ are the Lagrange multipliers and $\beta_y, \beta_n, \beta_m$ are positive parameters. The iterative algorithm is used to find a point satisfying the first-order condition; see Algorithm 14.

Before we discuss the solution to the minimization problem (76), we define a staggered grid system in Fig. 2; see more details of the implementation in Tai et al. (2011). We separate the minimization problem (76) into subproblems to pursue the solutions in an alternative mechanism.

Algorithm 14 Augmented Lagrangian method for Euler elastic model

Initialization: $x^{-1} = 0, y^{-1} = 0, n^{-1} = 0, m^{-1} = 0, \lambda_y^0 = 0, \lambda_n^0 = 0, \lambda_m^0 = 0$.

Iteration: For $k = 0, 1, \dots$:

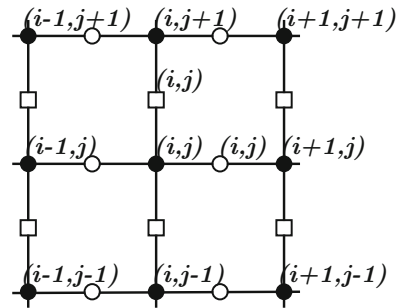
1. Compute (x^k, y^k, n^k, m^k) from

$$(x^k, y^k, n^k, m^k) \approx \arg \min_{(x,y,n,m)} \mathcal{L}_{EE}(x, y, n, m; \lambda_y^k, \lambda_n^k, \lambda_m^k), \tag{76}$$

2. Update

$$\begin{pmatrix} \lambda_y^{k+1} \\ \lambda_n^{k+1} \\ \lambda_m^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_y^k \\ \lambda_n^k \\ \lambda_m^k \end{pmatrix} + \begin{pmatrix} \beta_y(y^k - \nabla x^k) \\ \beta_n(n^k - m^k) \\ \beta_m(|y^k| - m^k \cdot y^k) \end{pmatrix}$$

Fig. 2 The rule of indexing variables in the augmented Lagrangian functional (75): $x, z, \lambda_z, \lambda_m$ are defined on \bullet -nodes. The first and second component of $y, n, m, \lambda_y, \lambda_n$ are defined on \circ -nodes and \square -node, respectively



The Solution to Sub-problem w.r.t. x

Given y , we solve the following minimization problem

$$\min_x \frac{1}{2} \|Kx - d\|^2 + \frac{\beta_y}{2} \|y - \nabla x\|^2 - \langle \lambda_y^k, \nabla x \rangle, \tag{77}$$

the first-order optimal condition of which gives us

$$(K^*K - \beta_y \Delta)x = K^*d - \beta_y \operatorname{div} y - \operatorname{div} \lambda_y^k.$$

Fast numerical methods can be used to solve the above equation such as fast Fourier transform (FFT) and iterative schemes.

The Solution to Sub-problem w.r.t. y

Given x, n , and m , we have the subproblem of y as follows

$$\min_y \langle a + b(\operatorname{div} n)^2, |y| \rangle + \langle \lambda_y^k, y \rangle + \langle \lambda_m^k + \beta_m, |y| - m \cdot y \rangle + \frac{\beta_y}{2} \|y - \nabla x\|^2, \tag{78}$$

which can be simplified as

$$\min_y \frac{\beta_y}{2} \left\| y - \left(\nabla x + \left(\frac{\lambda_m^k + \beta_m}{\beta_y} \right) m - \frac{\lambda_y^k}{\beta_y} \right) \right\|^2 + \langle |y|, a + b(\operatorname{div} n)^2 + \lambda_m^k + \beta_m \rangle.$$

Such the L^1 regularized minimization problem can be efficiently solved by the closed form solution.

The Solution to Sub-problem w.r.t. m

Given n and y , the sub-minimization problem of variable m becomes

$$\min_m I_{\mathcal{M}}(m) - \langle \lambda_m^k, m \rangle + \frac{\beta_n}{2} \|n - m\|^2 - \langle (\lambda_m^k + \beta_m)y, m \rangle. \quad (79)$$

We can reformulate the above minimization into a quadratic problem as follows

$$\min_m I_{\mathcal{M}}(m) + \frac{\beta_n}{2} \left\| m - \frac{(\lambda_m^k + \beta_m)y + \lambda_m^k}{\beta_n} - n \right\|^2,$$

the optimal solution of which can be achieved by performing the one-step projection to the solution of the quadratic minimization.

The Solution to Sub-problem w.r.t. n

Given m and y , we are going to solve the following minimization problem of n

$$\min_n \langle b(\operatorname{div} n)^2, |y| \rangle + \langle \lambda_n^k, n \rangle + \frac{\beta_n}{2} \|n - m\|^2, \quad (80)$$

the Euler-Lagrange equation of which is

$$-2\nabla(b|y| \operatorname{div} n) + \beta_n(n - m) + \lambda_n^k = 0,$$

and can be solved by a frozen coefficient method for easier implementation (Tai et al. 2011; Yashtini and Kang 2016).

Augmented Lagrangian Method for Mean Curvature-Based Model

Mean curvature-based model (Zhu and Chan 2012) considers an image restoration problem as a surface smoothing task. A basic model is as follows

$$\min_x \int_{\Omega} \left| \operatorname{div} \left(\frac{\nabla x}{\sqrt{1 + |\nabla x|^2}} \right) \right| dx + \frac{\alpha}{2} \int_{\Omega} (Kx - d)^2 dx. \quad (81)$$

Originally, the smoothed mean curvature model (81) was numerically solved by the gradient descent method, which involves high-order derivatives and converges slowly in practice. Zhu et al. (2013) developed an augmented Lagrangian method

for a mean curvature-based image denoising model (81), with similar ideas further studied in Myllykoski et al. (2015). Following Zhu et al. (2013), we rewrite the mean curvature-regularized model into the following constrained minimization problem

$$\begin{aligned} \min_{x,y,q,n,m} \quad & R_{MC}(q) + \frac{\alpha}{2} \|Kx - d\|^2 + I_{\mathcal{M}}(m) \\ \text{s.t.} \quad & y = \langle \nabla x, 1 \rangle, \quad q = \operatorname{div} n, \quad n = m, \quad |y| = y \cdot m, \end{aligned} \quad (82)$$

where $R_{MC}(q)$ is defined as

$$R_{MC}(q) = \sum_{1 \leq i, j \leq N} |q_{i,j}|.$$

The corresponding augmented Lagrangian functional for the constrained minimization problem is defined as

$$\begin{aligned} \mathcal{L}_{MC}(x, y, q, m, n; \lambda_y, \lambda_q, \lambda_n, \lambda_m) &= R_{MC}(q) + \frac{\alpha}{2} \|Kx - d\|^2 + I_{\mathcal{M}}(m) \\ &+ \langle \lambda_y, \langle \nabla x, 1 \rangle \rangle + \frac{\beta_y}{2} \|y - \langle \nabla x, 1 \rangle\|^2 + \langle q - \nabla \cdot n \rangle + \frac{\beta_q}{2} \|q - \nabla \cdot n\|^2 \\ &+ \langle \lambda_n, n - m \rangle + \frac{\beta_n}{2} \|n - m\|^2 + \langle \lambda_m, |y| - y \cdot m \rangle + \beta_m (|y| - y \cdot m), \end{aligned} \quad (83)$$

where $\lambda_y, \lambda_q, \lambda_n, \lambda_m$ are Lagrange multipliers and $\beta_y, \beta_q, \beta_n, \beta_m$ are positive parameters. The iterative algorithm is used to find a point satisfying the first-order condition; see Algorithm 15.

Algorithm 15 Augmented Lagrangian method for mean curvature-based model

Initialization: $x^{-1} = 0, y^{-1} = 0, q^{-1} = 0, n^{-1} = 0, m^{-1} = \mathbf{0}, \lambda_y^0 = 0, \lambda_q^0 = 0, \lambda_n^0 = 0, \lambda_m^0 = 0.$

Iteration: For $k = 0, 1, \dots$:

1. Compute $(x^k, y^k, q^k, n^k, m^k)$ from

$$(x^k, y^k, q^k, n^k, m^k) \approx \arg \min_{(x,y,q,n,m)} \mathcal{L}_{MC}(x, y, q, n, m; \lambda_y^k, \lambda_q^k, \lambda_n^k, \lambda_m^k), \quad (84)$$

2. Update

$$\begin{pmatrix} \lambda_y^{k+1} \\ \lambda_q^{k+1} \\ \lambda_n^{k+1} \\ \lambda_m^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda_y^k \\ \lambda_q^k \\ \lambda_n^k \\ \lambda_m^k \end{pmatrix} + \begin{pmatrix} \beta_y (y^k - \langle \nabla x^k, 1 \rangle) \\ \beta_q (q^k - \nabla \cdot n^k) \\ \beta_n (n^k - m^k) \\ \beta_m (|y^k| - y^k \cdot m^k) \end{pmatrix}$$

We can separate the minimization problem (84) into subproblems to obtain the solutions in an alternative way. Similarly as discussed for Euler's elastica model, the minimizers to the variable y , q , and m have closed form solutions, while the minimizers to the variable x and n are obtained by solving the associated Euler-Lagrange equations by either FFT or fast iterative schemes. Therefore, we omit the details here.

Numerical Experiments

In this section, we give some numerical results of augmented Lagrangian methods for solving the total variation-related image restoration models. For each model, we test only one image by considering the limit space. For more examples, please refer to literatures (Tai and Wu 2009; Wu and Tai 2010; Wu et al. 2011; Chan et al. 2013; Tai et al. 2011; Zhu et al. 2013). We perform the numerical experiments in MATLAB R2018A (Version 9.4) on a MacBook Pro with 2.3 GHz dual-core Intel Core i5 processor and 8GB memory. For each experiment, we stop the iteration until the following criterion

$$\frac{\|x^{k+1} - x^k\|}{\|x^k\|} < 1e - 3 \text{ (for multichannel case } \frac{\|x^{k+1} - x^k\|}{\|x^k\|} < 1e - 3)$$

satisfies. We measure the quality of the restored images by the improvement of signal to noise ratio (ISNR)

$$\text{ISNR}(x^*) = 10 \log_{10} \frac{\|\underline{x} - x^*\|}{\|\underline{x} - d\|},$$

where \underline{x} is the ground truth image, d is the observed image, and x^* is the recovered image. For multichannel case, we have the similar definition of ISNR. For each model, the parameter α is tuned to obtain the highest ISNR. The performances of augmented Lagrangian methods are demonstrated in Figs. 3, 4, 5, 6, 7, 8, 9, 10, and 11.

Figure 3 shows the results of augmented Lagrangian method for solving TV- L^2 model. In this experiment, we corrupt the clean image (size 512×512) with Gaussian blur and Gaussian noise. We set the parameters by following the recommendations in Wu and Tai (2010) and let $\beta = 10$. We report the recovered image and its ISNR in Fig. 3c. We also record the used CPU time t when the algorithm terminates. We can see that augmented Lagrangian method can solve TV- L^2 model efficiently and obtain high-quality recovered image.

Figure 4 shows the results of augmented Lagrangian method for solving TV- L^2 model with box constraint and the comparisons with TV- L^2 model. In this experiment, the degraded image (size 217×181) is corrupted with Gaussian blur and Gaussian noise. We set the parameters $\beta = \beta_y = 10$, and $\beta_z = 400$. We

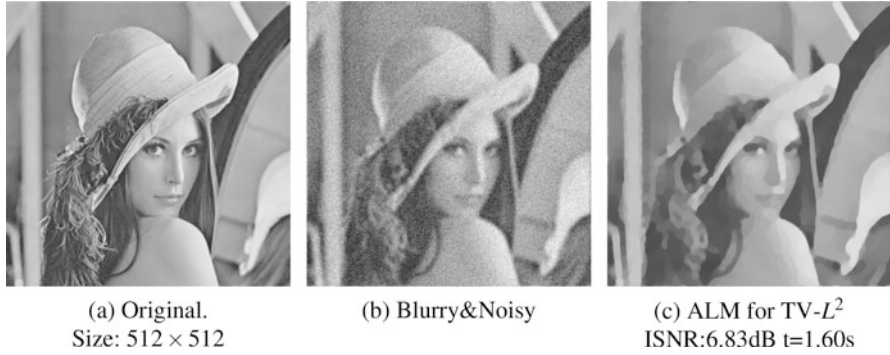


Fig. 3 Augmented Lagrangian method (ALM) for solving $TV-L^2$ model. (b) is a corruption of (a) with Gaussian blur $f_{\text{special}}('gaussian', 11, 3)$ and Gaussian noise with variation $1e-2$; (c) is the recovered result

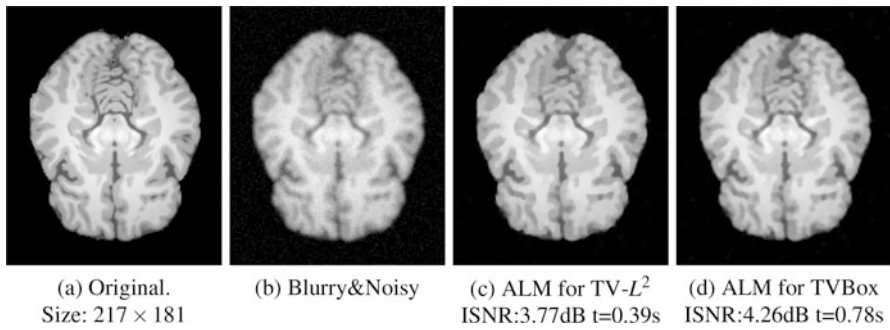


Fig. 4 Augmented Lagrangian method for solving $TV-L^2$ model with box constraint (TVBox). (b) is a corruption of (a) with Gaussian blur $f_{\text{special}}('gaussian', 5, 1.5)$ and Gaussian noise with variation $1e-3$; (c) and (d) are the recovered results

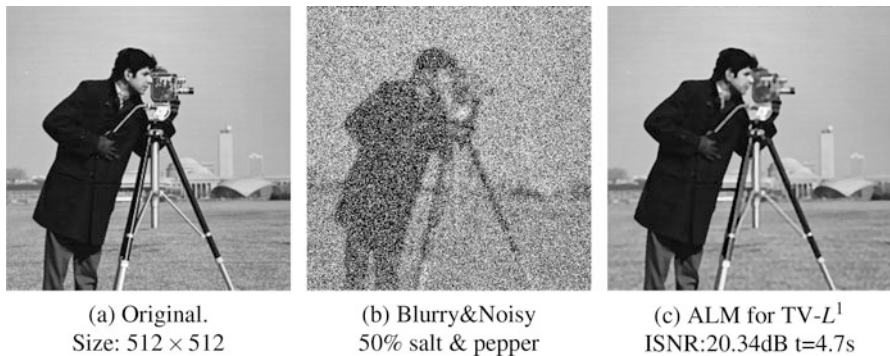


Fig. 5 Augmented Lagrangian method for solving $TV-L^1$ model. (b) is a corruption of (a) with Gaussian blur $f_{\text{special}}('gaussian', 11, 3)$ and 50% salt and pepper noise; (c) is the recovered result

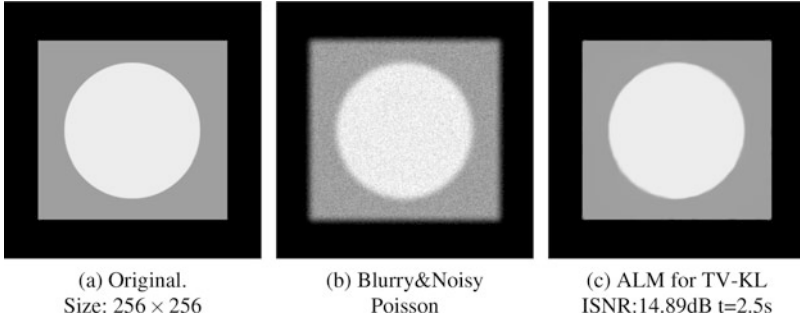


Fig. 6 Augmented Lagrangian method for solving TV-KL model. (b) is a corruption of (a) with Gaussian blur $f_{\text{special}}('gaussian', 11, 3)$ and Poisson noise; (c) is the recovered result



Fig. 7 Augmented Lagrangian method for multichannel TV (MTV) restoration (b) is a corruption of (a) with within-channel Gaussian blur $f_{\text{special}}('gaussian', 21, 5)$, and Gaussian noise with variation $1e - 3$; (c) is the recovered result

report the recovered images and their ISNRs in Fig. 4c, d. We also record the used CPU times t when the algorithms terminate. We can see that augmented Lagrangian method can solve TV- L^2 model with box constraint efficiently and obtain high-quality recovered image. The TV- L^2 model with box constraint gains higher ISNR than the TV- L^2 model.

Figures 5 and 6 show the results of augmented Lagrangian methods for TV- L^1 model and TV-KL model. In the experiment for TV- L^1 model, the observed image (size 512×512) is degraded with Gaussian blur and 50% salt and pepper noise. We set $\beta_y = 20$ and $\beta_z = 100$. In the experiment for TV-KL model, the observed image (size 256×256) is corrupted with Gaussian kernel and Poisson noise. We let $\beta_y = 20$ and $\beta_z = 20$. We can see that augmented Lagrangian methods can recover high-quality images in these two experiments and the CPU costs are low.

Figure 7 shows the results of augmented Lagrangian method for multichannel TV restoration. In this experiment, the degraded image is generated by first blurring the ground truth image (size $512 \times 512 \times 3$) with within-channel Gaussian blur and then adding Gaussian noise to the blurred image. We set $\beta = 100$. We also can see that

Fig. 8 Augmented Lagrangian method for solving TV^2 model. (b) is a corruption of (a) with Gaussian blur `fspecial('gaussian', 11, 3)` and Gaussian noise with variation $1e - 2$; (c) and (d) are the recovered results

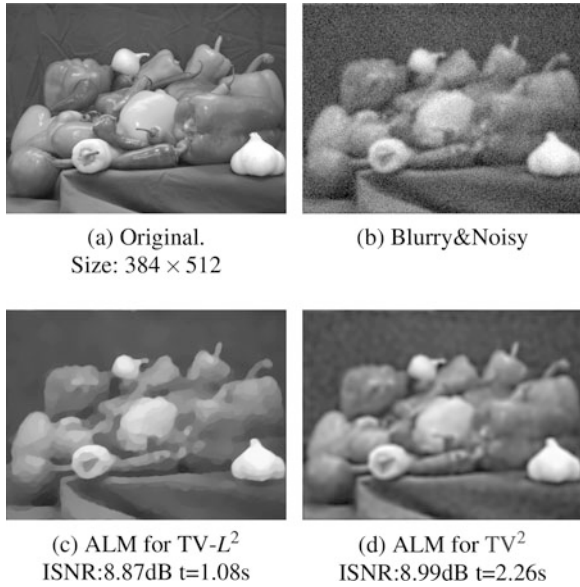
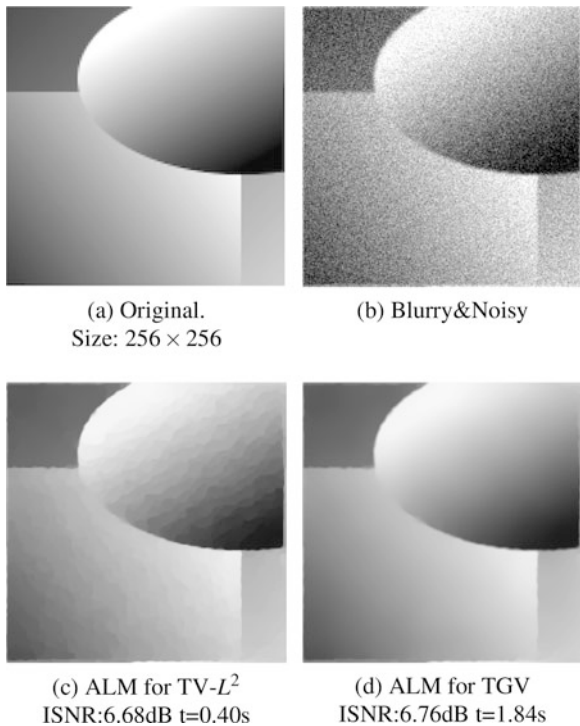


Fig. 9 Augmented Lagrangian method for solving TGV model. (b) is a corruption of (a) with Gaussian blur `fspecial('gaussian', 5, 1.5)` and Gaussian noise with variation $1e - 2$; (c) and (d) are the recovered results



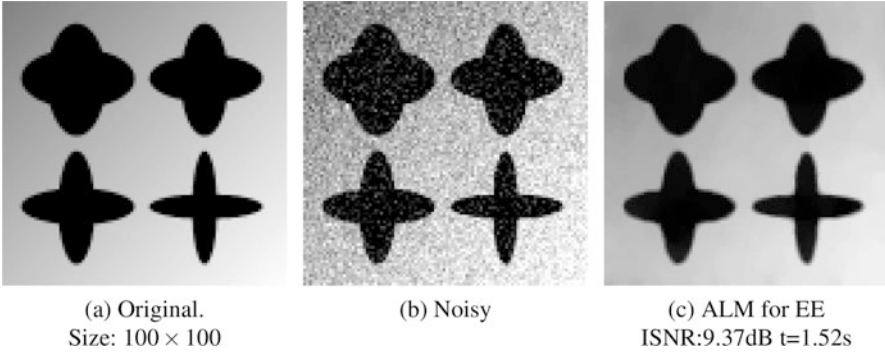


Fig. 10 Augmented Lagrangian method for solving Euler's elastica (EE) based image denoising model. (b) is a corruption of (a) with Gaussian noise with variation $1e - 2$; (c) is the recovered result

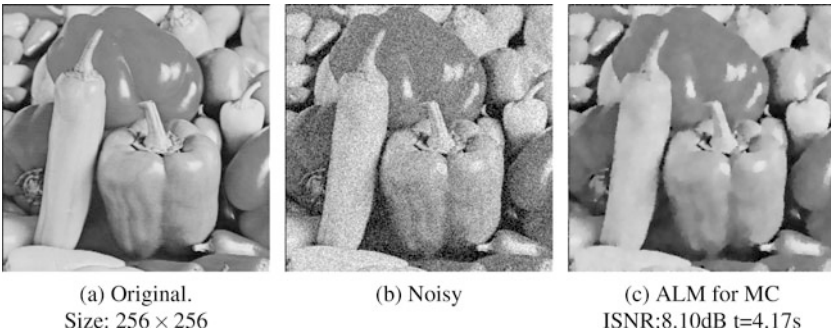


Fig. 11 Augmented Lagrangian method for solving mean curvature (MC)-based image denoising model. (b) is a corruption of (a) with Gaussian noise with variation $1e - 2$; (c) is the recovered result

augmented Lagrangian method can restore high-quality multichannel image with a low CPU cost.

Figures 8 and 9 show the results of augmented Lagrangian methods for solving TV^2 model and TGV model and the comparisons with $TV-L^2$ model. In the experiment for TV^2 model, the degraded image (size 384×512) is generated with Gaussian blur and Gaussian noise. We set $\beta = 10$. In the experiment for TGV model, the degraded image (size 256×256) is also generated with Gaussian blur and Gaussian noise. We let $(\alpha_0, \alpha_1) = (1.0, 0.1)$, $\beta_y = 10$ and $\beta_z = 20$. We report the recovered images and their ISNRs in Figs. 8c, d and 9c, d. We also record the used CPU times t when the algorithms terminate. We can see that augmented Lagrangian method can solve TV^2 model and TGV model efficiently and obtain high-quality recovered images. The TV^2 model and TGV model, which use high-order regularization, can suppress the staircase effect well.

Figures 10 and 11 show the results of augmented Lagrangian methods for solving Euler's elastica-based image denoising model and mean curvature-based image denoising model. Both these two models include curvature term in the regularization and are non-convex and highly nonlinear. We generate the degraded images Figs. 10b, 11b by adding Gaussian noise to the clean images Figs. 10a and 11a, respectively. In the experiment for Euler's elastica based model, we use $\beta_y = 200$, $\beta_n = 500$ and $\beta_m = 1$. In the experiment for mean curvature-based model, we use $\beta_y = 40$, $\beta_q = 1e5$, $\beta_n = 1e5$ and $\beta_m = 40$. We report the recovered images and their ISNRs and show the CPU costs in Figs. 10c and 11c. We can see that augmented Lagrangian methods can solve non-convex curvature-based models efficiently and obtain high-quality recovered images.

Conclusions

In this survey, we have reviewed variable splitting and augmented Lagrangian methods for total variation-related image restoration models. Due to the closed form solutions of subproblems and fast linear solvers like the FFT implementations, these methods are efficient for both total variation-related convex models and non-convex Euler's elastica and mean curvature-based models.

Acknowledgments Tai is supported by NSFC/RGC Joint Research Scheme (N_HKBU214/19), Initiation Grant for Faculty Niche Research Areas(RC-FNRA-IG/19-20/SCI/01) and CRF (C1013-21GF).

References

- Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Prob.* **10**(6), 1217–1229 (1994)
- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, 2nd edn. Springer, New York (2010)
- Bae, E., Shi, J., Tai, X.C.: Graph cuts for curvature based image denoising. *IEEE Trans. Image Process.* **20**(5), 1199–1210 (2010)
- Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **18**(11), 2419–2434 (2009)
- Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.* **12**(8), 882–889 (2003)
- Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Optimization and Neural Computation Series, Athena Scientific, Belmont, Mass (1996 (firstly published in 1982))
- Blomgren, P., Chan, T.F.: Color TV: Total variation methods for restoration of vector-valued images. *IEEE Trans. Image Process.* **7**(3), 304–309 (1998)
- Boyd, S.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2010)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
- Bredies, K., Pock, T., Wirth, B.: A convex, lower semicontinuous approximation of Euler's elastica energy. *SIAM J. Math. Anal.* **47**(1), 566–613 (2015)

- Brune, C., Sawatzky, A., Burger, M.: Bregman-em-tv methods with application to optical nanoscopy. In: Tai, X.C., Mørken, K., Lysaker, M., Lie, K.A. (eds.) *Scale Space and Variational Methods in Computer Vision*. Springer, Berlin/Heidelberg, pp. 235–246 (2009)
- Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* **20**(1/2), 89–97 (2004)
- Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. *Numer. Math.* **76**(2), 167–188 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Chan, R.H., Tao, M., Yuan, X.: Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers. *SIAM J. Imaging Sci.* **6**(1), 680–697 (2013)
- Chan, T., Wong, C.K.: Total variation blind deconvolution. *IEEE Trans. Image Process.* **7**(3), 370–375 (1998)
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chan, T.F., Kang, S.H., Shen, J.: Euler’s elastica and curvature-based inpainting. *SIAM J. Appl. Math.* **63**(2), 564–592 (2002)
- Chang, H., Lou, Y., Ng, M., Zeng, T.: Phase retrieval from incomplete magnitude information via total variation regularization. *SIAM J. Sci. Comput.* **38**(6), A3672–A3695 (2016)
- Chen, C., Chen, Y., Ouyang, Y., Pasiliao, E.: Stochastic accelerated alternating direction method of multipliers with importance sampling. *J. Optim. Theory Appl.* **179**(2), 676–695 (2018)
- Chen, X., Ng, M.K., Zhang, C.: Non-Lipschitz ℓ_p -regularization and box constrained model for image restoration. *IEEE Trans. Image Process.* **21**(12), 4709–4721 (2012)
- Chen, Y., Levine, S., Rao, M.: Variable exponent, linear growth functionals in image restoration. *SIAM J. Appl. Math.* **66**(4), 1383–1406 (2006)
- Deng, L.J., Glowinski, R., Tai, X.C.: A new operator splitting method for the Euler elastica model for image smoothing. *SIAM J. Imaging Sci.* **12**(2):1190–1230 (2019)
- Deng, W., Yin, W.: On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.* **66**(3), 889–916 (2016)
- Duan, Y., Wang, Y., Hahn, J.: A fast augmented Lagrangian method for Euler’s elastica models. *Numer. Math. Theory Methods Appl.* **006**(001), 47–71 (2013)
- Fazel, M., Pong, T.K., Sun, D., Tseng, P.: Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. Appl.* **34**(3), 946–977 (2013)
- Feng, X., Wu, C., Zeng, C.: On the local and global minimizers of ℓ_0 gradient regularized model with box constraints for image restoration. *Inverse Prob.* **34**(9), 095,007 (2018)
- Gao, Y., Liu, F., Yang, X.: Total generalized variation restoration with non-quadratic fidelity. *Multidim. Syst. Sign. Process.* **29**(4), 1459–1484 (2018)
- Glowinski, R., Tallec, P.L.: *Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia (1989)
- Glowinski, R., Osher, S.J., Yin, W. (eds.): (2016) *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, Cham
- Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
- Güven, H.E., Güngör, A., Çetin, M.: An augmented Lagrangian method for complex-valued compressed SAR imaging. *IEEE Trans. Comput. Imag.* **2**(3), 235–250 (2016)
- Hahn, J., Wu, C., Tai, X.C.: Augmented Lagrangian method for generalized TV-Stokes model. *J. Sci. Comput.* **50**(2), 235–264 (2012)
- He, B., Yuan, X.: On the $o(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.* **50**(2), 700–709 (2012)
- Hinterberger, W., Scherzer, O.: Variational methods on the space of functions of bounded Hessian for convexification and denoising. *Computing* **76**(1–2), 109–133 (2006)
- Hintermüller, M., Wu, T.: Nonconvex TV^q -models in image restoration: Analysis and a trust-region regularization-based superlinearly convergent solver. *SIAM J. Imaging Sci.* **6**(3), 1385–1415 (2013)

- Kang, S.H., Zhu, W., Jianhong, J.: Illusory shapes via corner fusion. *SIAM J. Imaging Sci.* **7**(4), 1907–1936 (2014)
- Lai, R., Chan, T.F.: A framework for intrinsic image processing on surfaces. *Comput. Vis. Image Und* **115**(12), 1647–1661 (2011)
- Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. *J. Math. Imaging Vis.* **27**, 257–263 (2007)
- Li, C., Yin, W., Jiang, H., Zhang, Y.: An efficient augmented Lagrangian method with applications to total variation minimization. *Comput. Optim. Appl.* **56**(3), 507–530 (2013)
- Li, G., Pong, T.K.: Global convergence of splitting methods for nonconvex composite optimization. *SIAM J. Optim.* **25**(4), 2434–2460 (2015)
- Liu, Z., Wali, S., Duan, Y., Chang, H., Wu, C., Tai, X.C.: Proximal ADMM for Euler’s elastica based image decomposition model. *Numer. Math. Theory Methods Appl.* **12**(2), 370–402 (2018)
- Lou, Y., Zhang, X., Osher, S., Bertozzi, A.L.: Image recovery via nonlocal operators. *J. Sci. Comput.* **42**(2), 185–197 (2010)
- Lysaker, M., Lundervold, A., Tai, X.: Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Process.* **12**(12), 1579–1590 (2003)
- Micchelli, C.A., Shen, L., Xu, Y.: Proximity algorithms for image models: denoising. *Inverse Prob.* **27**(4), 045,009 (2011)
- Mylykoski, M., Glowinski, R., Karkkainen, T., Rossi, T.: A new augmented Lagrangian approach for L^1 -mean curvature image denoising. *SIAM J. Imaging Sci.* **8**(1), 95–125 (2015)
- Nikolova, M.: A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**(1–2), 99–120 (2004)
- Nikolova, M.: Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. *Multiscale Model. Simul.* **4**(3), 960–991 (2005)
- Ouyang, Y., Chen, Y., Lan, G., Pasiliao, E.: An accelerated linearized alternating direction method of multipliers. *SIAM J. Imaging Sci.* **8**(1), 644–681 (2015)
- Persson, M., Bone, D., Elmqvist, H.: Total variation norm for three-dimensional iterative reconstruction in limited view angle tomography. *Phys. Med. Biol.* **46**(3), 853–866 (2001)
- Ramani, S., Fessler, J.A.: Parallel MR image reconstruction using augmented Lagrangian methods. *IEEE Trans. Med. Imaging* **30**(3), 694–706 (2011)
- Rockafellar, R.T.: Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM J. Control* **12**(2), 268–285 (1974)
- Rockafellar, R.T., Wets, R.J.B.: *Variational Analysis*. Springer, Berlin/Heidelberg (1998)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
- Sapiro, G., Ringach, D.: Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Image Process.* **5**, 1582–1586 (1996)
- Selesnick, I., Lanza, A., Morigi, S., Sgallari, F.: Non-convex total variation regularization for convex denoising of signals. *J. Math. Imaging Vis.* **62**(6), 825–841 (2020)
- Tai, X.C., Wu, C.: Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In: *Scale Space and Variational Methods in Computer Vision, Second International Conference, SSVM 2009, Voss, 1–5 June 2009*. Proceedings, pp 502–513 (2009)
- Tai, X.C., Hahn, J., Chung, G.J.: A fast algorithm for Euler’s elastica model using augmented Lagrangian method. *SIAM J. Imaging Sci.* **4**(1), 313–344 (2011)
- Vese, L.A., Osher, S.J.: Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.* **19**(1/3), 553–572 (2003)
- Wang, X., Yuan, X.: The linearized alternating direction method of multipliers for dantzig selector. *SIAM J. Sci. Comput.* **34**(5), A2792–A2811 (2012)
- Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci.* **1**(3), 248–272 (2008)
- Wang, Y., Yin, W., Zeng, J.: Global convergence of ADMM in nonconvex nonsmooth optimization. *J. Sci. Comput.* **78**(1), 29–63 (2019)

- Wu, C., Tai, X.C.: Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sci.* **3**(3), 300–339 (2010)
- Wu, C., Zhang, J., Tai, X.C.: Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Probl. Imaging* **5**(1), 237–261 (2011)
- Wu, C., Zhang, J., Duan, Y., Tai, X.C.: Augmented lagrangian method for total variation based image restoration and segmentation over triangulated surfaces. *J. Sci. Comput.* **50**(1), 145–166 (2012)
- Wu, C., Liu, Z., Wen, S.: A general truncated regularization framework for contrast-preserving variational signal and image restoration: Motivation and implementation. *Sci. China Math.* **61**(9), 1711–1732 (2018)
- Yan, M., Duan, Y.: Nonlocal elastica model for sparse reconstruction. *J. Math. Imaging Vis.* **62**, 532–548 (2020)
- Yang, J., Yin, W., Zhang, Y., Wang, Y.: A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM J. Imaging Sci.* **2**(2), 569–592 (2009)
- Yashtini, M., Kang, S.H.: A fast relaxed normal two split method and an effective weighted TV approach for Euler’s elastica image inpainting. *SIAM J. Imaging Sci.* **9**(4), 1552–1581 (2016)
- Zeng, C., Wu, C.: On the edge recovery property of nonconvex nonsmooth regularization in image restoration. *SIAM J. Numer. Anal.* **56**(2), 1168–1182 (2018)
- Zeng, C., Wu, C.: On the discontinuity of images recovered by nonconvex nonsmooth regularized isotropic models with box constraints. *Adv. Comput. Math.* **45**(2), 589–610 (2019)
- Zhang, H., Wu, C., Zhang, J., Deng, J.: Variational mesh denoising using total variation and piecewise constant function space. *IEEE Trans. Vis. Comput. Graphics* **21**(7), 873–886 (2015)
- Zhang, J., Chen, K.: A total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution. *SIAM J. Imaging Sci.* **8**(4), 2487–2518 (2015)
- Zhu, W., Chan, T.: Image denoising using mean curvature of image surface. *SIAM J. Imaging Sci.* **5**(1), 1–32 (2012)
- Zhu, W., Tai, X.C., Chan, T.: Augmented Lagrangian method for a mean curvature based image denoising model. *Inverse Prob. Imaging* **7**(4), 1409–1432 (2013)



Sparse Regularized CT Reconstruction: An Optimization Perspective

14

Elena Morotti and Elena Loli Piccolomini

Contents

Introduction	552
Tomographic Imaging	554
Mathematics of Sparse Tomography	557
Lambert Beer's Law	557
The Radon Transform and Its Discretization	559
The Filtered Back Projection Algorithm	559
Model-Based Approaches for Sparse-View CT	561
From Lambert-Beer's Law to a Linear System	561
Implementation of the Forward Operator M	562
The Optimization Framework	564
Iterative Algorithms for Optimization	566
Regularization: Little or Too Much?	566
Toward the Convergence of the Iterative Method	568
New Frontiers of CT Reconstruction with Deep Learning	569
Case Study: Reconstruction of Digital Breast Tomosynthesis Images	570
DBT 3D Imaging	571
Model and Analysis	572
Reconstructions of the Accreditation Phantom	574
Reconstructions of a Human Dataset	576
Distance-Driven Approach for 3D CT Imaging	578
Code Parallelization	579
Conclusion	581
References	581

E. Morotti

Department of Political and Social Sciences, University of Bologna, Bologna, Italy

e-mail: elena.morotti4@unibo.it

E. L. Piccolomini (✉)

Department of Computer Science and Engineering, University of Bologna, Bologna, Italy

e-mail: elena.loli@unibo.it

Abstract

In Computed Tomography (CT), decreasing the X-rays dose is essential to reduce the negative effects of radiation exposure on the human health. One possible way to accomplish it is to reduce the number of projections acquired, hence the name of *sparse* CT. Traditional methods for image reconstruction cannot recover reliable images in this case: the lack of information due to the missed projections produces strong artifacts. Alternatively, optimization frameworks are flexible models where incorporated regularization functions impose regularity constraints on the solution, thus avoiding unwanted artifacts and contrasting noise propagation. Since the iterative methods solving the optimization problem calculate more accurate solutions as iterations (and computational time) increase, it is possible to choose a better reconstructed image at the expense of execution time, or viceversa. Parallel implementations of the iterative solvers significantly reduce the computational time, allowing for a large number of iterations in a prefixed short time.

Here, the effectiveness of the optimization approach is shown on the case study of 3D reconstruction of breast images from tomosynthesis with tests on real projection data.

Keywords

Sparse-view CT · Tomographic image reconstruction · Model-based iterative methods · Breast tomosynthesis

Introduction

X-ray computed tomography (CT) is an imaging technique which has first been experimented in the medical area, as the evolution of the projection radiography. In particular, medical imaging was born not long after Wilhelm Röntgen discovered X-rays in 1895, as soon as scientists realized X-ray capability of crossing objects: for decades 2D planar images (projection radiographies) have been used to investigate the inner parts of human bodies. However, these images represent a mean of the information of the 3D scanned object which is squeezed on a 2D plane. In the 1930s, a new mathematical theory by Johann Radon published in 1917, the studies by the physician Grossmann, together with the desire to overcome the averaging process of the conventional X-ray radiography, led to the definition of tomography as a new tool for object inspection. Since the advent of computers in the 1970s, CT has raised and revolutionized the non-intrusive diagnostic imaging by allowing the three-dimensional orientation of anatomy to be reconstructed in transverse (cross-sectional) sections.

To achieve it, the CT imaging device acquires several projections of the same slice of the object under exam, from angled views in a round trajectory. Then, a software reconstructs the digital image from the acquired projection data. Hence, tomographic image reconstruction mathematically represents an *inverse problem*.

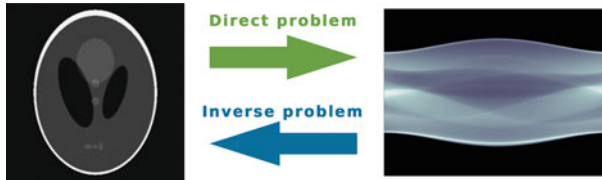


Fig. 1 In CT imaging, the direct problem (from the object to the data) is represented by the acquisition of the sinogram, whereas the inverse problem (from the data to the object) is the reconstruction of the image

In an inverse problem, in fact, only the measurements of an effect are known, as given data, whereas the cause represents the unknown that must be retrieved as solution of the problem. The cause-effect pair in CT is represented in Fig. 1, which shows the well-known Shepp-Logan brain phantom as scanned object and its projection dataset, acquired by the CT system in an entire 2D scan and organized as a *sinogram* image (as introduced in the following). The inverse problem aims at recovering the phantom image as accurately as possible from the sinogram.

Mathematically, inverse problems are generally *ill-posed* in the sense of Hadamard (1902), i.e., one of the following conditions is not satisfied:

1. At least one solution of the problem exists.
2. The solution of the problem is unique.
3. The solution continuously depends on the data.

Traditional methods for CT cannot face the ill-posedness and compute images with unwanted artifacts and noise. To face this, a more recent approach models the CT imaging process as an optimization problem where the inverse problem is solved by inverting the discrete model, represented by a linear system, constrained by means of regularization functions. Imposing regularization allows to choose a good solution among the infinite possible ones.

In particular, the optimization problem is solved by iterative algorithms (called *model-based iterative algorithms*). They converge to the problem solution in many iterations, but they should possibly compute a good solutions far before convergence. In fact, a slow convergence would make model-based iterative algorithms not usable on real systems, where very fast executions are required for clinical needs. However, acceleration techniques make iterative algorithms produce good solutions in few iterations, and efficient parallel executions on low-cost GPU boards greatly reduce the execution time; hence the optimization approach is effective in real applications.

The aim of this chapter is:

- to derive the optimization framework from the mathematical model of CT;
- to highlight the flexibility of the optimization framework, where different regularization terms can be easily incorporated and different iterative algorithms can be used for solving the minimization problem;

- to show that the solution of the optimization problem computed by an iterative method converges toward an accurate reconstruction;
- to present a 3D real case of limited angle tomography with an example of parallel execution on GPU boards, demonstrating that it is possible to achieve short execution times compatible to the speed and quality standards in clinical settings.

The chapter is organized as follows. The next section contains a brief survey both on the CT scan geometries (with particular attention to few-view protocols) and on the mathematics of CT imaging. Then, the regularized optimization framework is presented for the CT image reconstruction; examples of iterative reconstructions as solution of the optimization problem from a 2D phantom prove the effectiveness of the approach. Finally, a case study on 3D breast tomosynthesis is analyzed with results from a parallel implementation on GPUs.

Tomographic Imaging

From the primordial systems to the most modern gantries currently used in medicine and industrial applications, many studies have been led by different research groups, collecting engineers, physicists, mathematicians, and computer scientists, with the aim of improving both the technologies and the reconstruction software. For each prefixed angled position of the X-ray source, first-generation CT devices performed long-lasting projections where parallel rays allowed simple reconstruction algorithms (top-left image in Fig. 2). Among the numerous developments, the shift from parallel- to fan-beam X-ray projections has been the most significant. Fan-beam geometries are preferred today, since they enable to acquire all the single-view measurements in one fan simultaneously (top-right image in Fig. 2). However, computation speedups are required when recovering objects from fan-beam projections in real scenarios (Averbuch et al. 2011).

Historically, a further step forward has been the blooming of 3D CT imaging systems. The first developments led to helical CT, where the X-ray source walked on a narrow helical trajectory scanning a volume with fan beams, slice by slice. As depicted in Fig. 2, another approach exploits cone-beam projections to run over a volume in just one scan. In this case, the X-ray source rotates on a circular planar trajectory.

In the last years, many tomographic devices have been designed to fit different medical needs, and, on the other hand, interesting technical, anthropomorphic, forensic, and archeological as well as paleontological applications of CT have been developed too (Hughes 2011; De Chiffre et al. 2014). As a consequence, the CT technique is evolving into new inquiring forms. In particular, motivated by an increasing focus on the potentially harmful effects of X-ray ionizing radiation, a recent trend in CT research is to develop safer protocols to reduce the radiation dose per patient. This allows to apply CT techniques to a wider class of medical examinations, including vascular, dental, orthopedic, musculoskeletal, chest, and mammographic imaging. Safer protocols are of interest not only for medicine but

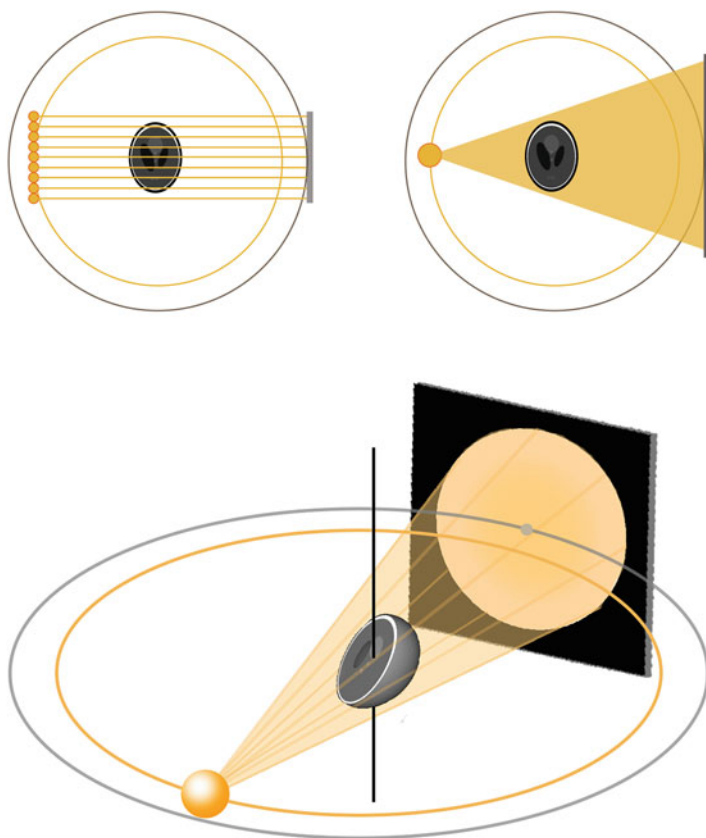


Fig. 2 Sketches of tomographic devices, from the primordial technology with parallel X-ray scans (top left) to the most modern solution exploiting fan beams for 2D (top right) and cone beams (bottom) for 3D CT

also for material science and cultural heritage, to prevent damage to the subject under study, due to excessive radiations.

Specifically, there are two main techniques allowing for a significant reduction of the total radiation exposure per patient. The first one, usually named *low-dose CT*, consists in reducing the X-ray tube current at each scan. In this case, the geometry traditionally used in CT, where up to one thousand projections are taken along the circular trajectory, does not change, but the measured data presents higher quantum noise. The second practical way to lower the radiation consists in reducing the number of X-ray projections. The resulting protocols are labeled as *sparse tomography* (or sparse-view, few-view tomography), and it leads to incomplete tomographic data, but very fast examinations (Kubo et al. 2008; Yu et al. 2009). Figure 3 shows a graphical draft of the reconstruction process. In the first row, the classical full-dose CT case is represented; in the second row, a *sparse-view*

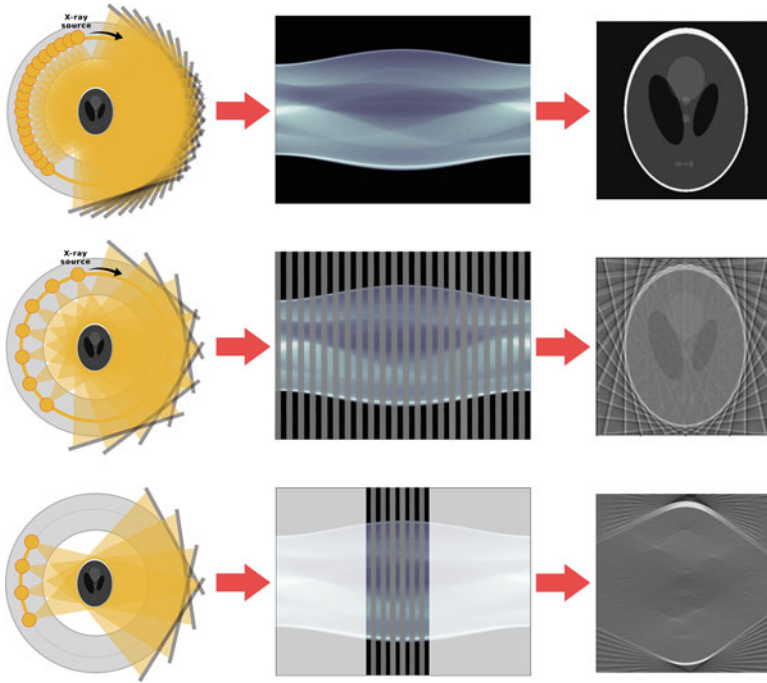


Fig. 3 Sketches of the tomographic image reconstruction workflow, for full-view, sparse-view full-angle, and limited-angle protocols (from top to bottom, respectively). From the different geometries on the left, the acquired projections and the reconstructed image of the Shepp-Logan phantom. The missing portions of sinogram in the sparse-view and limited-angle protocols are depicted in light gray

full-angle tomography is considered where a reduced number of views are taken in the whole circular orbit. A different sparse-view geometry using few projections is called *limited-angle tomography* (see the third row of Fig. 3). Here, a further reduction of X-ray scans is made by limiting the source trajectory to a C-shaped path, i.e., by restricting the 360-degree angular scanning interval to a range smaller than 180 degrees. In some tomographic applications, the human anatomy does not allow a complete circular motion to the X-ray source; thus, the use of a reduced range is mandatory and the resulting technique is called *tomosynthesis*. An example is breast imaging, where the breast is in a stationary position between the detector surface and the compression plate (Wu et al. 2004; Zhang et al. 2006; Reiser et al. 2009; Barca et al. 2021). The source moves through a quite limited arc (at most 80 degrees) over the breast. Another possible reason for using limited angles is the impossibility of probing through a ball in the center of the target, such as in nondestructive testing (Quinto 1993).

A low radiation dose and high in-plane resolution make sparse tomography an attractive alternative to full-view computed tomography. However, the incompleteness of the projection data results in image artifacts that may disable diagnostic

interpretation. As depicted in Fig. 3, the sets of projection data are severely sub-sampled in case of sparse-view and limited-angle acquisitions with respect to the full-dose case. The resulting lack of information causes well-studied artifacts on the images reconstructed with the algorithms traditionally used for full-view protocols. However, thanks to the efficiency of new reconstruction approaches, some low-dose and sparse-view protocols have already been approved for screening tests: safer tomographic exams can indeed be led without compromising the reliability of their diagnosis (Mueller and Siltanen 2012; He et al. 2018).

Mathematics of Sparse Tomography

What is there behind the X-ray imaging techniques? From a physical point of view, the projection data reflects the absorption of the photons constituting the X-rays, and the image of the scanned object is a picture of the attenuation coefficient map in pseudo-colors. The physical model describing photons absorption in terms of attenuation coefficients is described in the Lambert Beer's law.

Lambert Beer's Law

All the physical mechanisms leading to the attenuation of radiation intensity (i.e., reduction of photons) measured by a detector behind a homogeneous object are usually described by a single attenuation coefficient $\mu = \mu(w) \geq 0$ depending on the crossed point w . The total attenuation of a monochromatic X-ray beam passing through a dense object of thickness Δw can be calculated in the following way (Buzug 2011):

$$m(w + \Delta w) = m(w) - \mu(w)m(w)\Delta w \quad (1)$$

where $m(w)$ is the intensity of the incoming beam. Reordering (1) and computing the limit, it holds:

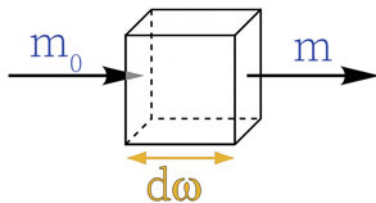
$$\lim_{\Delta w \rightarrow 0} \frac{m(w + \Delta w) - m(w)}{\Delta w} = \frac{dm}{dw} = -\mu(w)m(w). \quad (2)$$

By assuming the object to be homogeneous (i.e., $\mu(w) = \mu$ along the entire path Δw), the solution of the differential equation (2) computed by separation of variables and integration is:

$$\ln |m(w)| = -\mu w + C. \quad (3)$$

Imposing the initial condition $m(0) = m_0$ (where m_0 is the known emitted photon count, as in Fig. 4) and considering that all the measured intensities are positive quantities, the previous equation can be written as:

Fig. 4 Scheme of the X-ray absorption by an infinitesimal object. The number m_0 of input photons is reduced to $m < m_0$ at output after crossing a thickness $d\omega$



$$m(w) = m_0 e^{-\mu w}. \tag{4}$$

Equation (4) is known as the *Lambert Beer's law* of attenuation.

In practice, the attenuation coefficient $\mu(w)$ is not constant along the ray path. In this case the solution for the intensity measured after a running length W is given by:

$$m = m_0 e^{-\int_0^W \mu(w) dw} \tag{5}$$

and the *projection integral* of μ along a segment of length W is computed as:

$$\mathcal{P}_W \mu = -\ln \left(\frac{m}{m_0} \right) = \int_0^W \mu(w) dw. \tag{6}$$

By setting the plane coordinates as $w = (x, y)$ (the attenuation coefficient is a continuous function $\mu(w) = \mu(x, y)$ over the spatial domain of the slice) and naming L the integration path, the following relation holds:

$$-\ln \left(\frac{m}{m_0} \right) = + \int_L \mu(x, y) dw \tag{7}$$

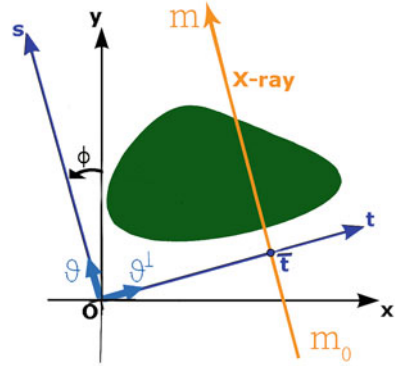
by assuming the air coefficient outside the object $\mu(x, y) = 0$.

Suppose to rotate the xy -plane of an angle Φ and to set a change of variables from (x, y) to (t, s) as in Fig. 5. By considering the X-ray parallel to the direction of the vector θ with $t = \bar{t}$, the projection of the attenuation coefficient μ along L becomes:

$$\int_L \mu(x, y) dw = \int_{-\infty}^{+\infty} \mu(\bar{t}\theta^\perp + s\theta) ds. \tag{8}$$

Since the direction of the vector θ is uniquely determined by the rotation angle Φ , it is convenient to denote with θ also the rotation angle. Now, by considering the X-ray parallel beam emitted from the θ -angled position, the projection of the whole object described by μ is the map $\mathcal{P}_\theta \mu : \mathbb{R} \rightarrow \mathbb{R}$ such that:

Fig. 5 Scheme of the X-ray absorption by an object, to illustrate the rotation of the coordinate system used in (8)



$$\mathcal{P}_\theta \mu(t) = \int_{-\infty}^{+\infty} \mu(t\theta^\perp + s\theta) ds, \quad \forall t \in \mathbb{R}. \tag{9}$$

The Radon Transform and Its Discretization

In 1917, a well-known paper by the Austrian mathematician Johann Radon provided the mathematical foundation for tomographic imaging reconstruction. The *Radon transform* of μ is defined as the map $\mathcal{R} : [0, 2\pi] \times \mathbb{R} \rightarrow \mathbb{R}$ such that:

$$(\mathcal{R}\mu)(\theta, t) = \mathcal{P}_\theta \mu(t), \quad \forall \theta \in [0, 2\pi], \forall t \in \mathbb{R} \tag{10}$$

In other words, the Radon transform \mathcal{R} of an object slice described by μ is the set of projections acquired along the full-angle circular trajectory, in a continuous model.

The process defining the full-dose tomography represents a discrete realization of the (continuous) Radon transform. The graphical representation of all the measured data, in the bidimensional case, is called *sinogram*, and it is represented in Fig. 3 for full-view, sparse-view, and limited-angle geometries. As it is clearly visible, in case of sparse-view and limited-angle protocols, the incomplete projections provide only a portion of the entire sinogram, making the corresponding inverse problems trickier and the reconstruction process more complicated than in the full-view case.

The Filtered Back Projection Algorithm

Historically, the first technique implemented to reconstruct CT images from projections is the *filtered back projection* (FBP) (Feldkamp et al. 1984). To recover the attenuation coefficient function, the basic idea of FBP is to project backward every

data onto the original ray path causing such absorption (see Kak and Slaney (2001) for more details).

The FBP algorithm is still implemented in many commercial systems, since it computes the output image in a very short time, which is a fundamental request in medical setting. However, it is well known that in the case of few views the FBP algorithm produces images corrupted by artifacts and noise (Natterer 2001).

Figure 6 shows some FBP reconstructions of the well-known Shepp-Logan digital phantom obtained at different sparse geometries. The sparsity is boosted by decreasing the angular range (from top to bottom) and the number of views (from left to right). The FBP image quality deteriorates: the large angular step characterizing sparse-view projections leads to streaking artifacts on the image, whereas a limited-angle acquisition produces a swiped band corresponding to the lost projecting directions. In the last row, where the scan is limited to a 60-degree arc, the object inside the brain is deformed and not distinguishable, regardless of the number of projection numbers.

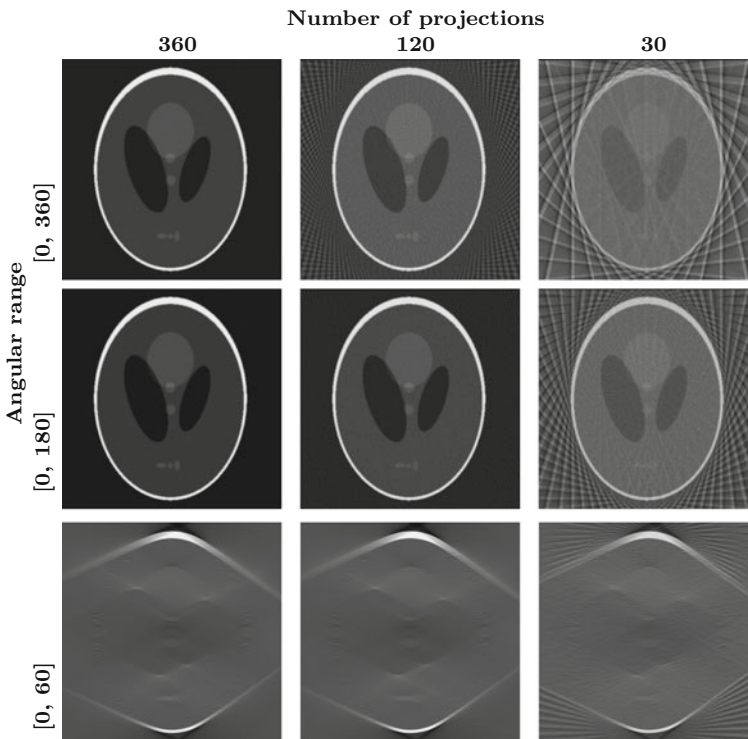


Fig. 6 Shepp Logan reconstructions by the popular FBP algorithm, at different geometric settings

Model-Based Approaches for Sparse-View CT

A valid alternative to FBP for sparse-view CT image reconstruction is represented by model-based iterative methods, which derive from the discretization of Lambert-Beer’s law (4).

From Lambert-Beer’s Law to a Linear System

In the real discrete setting, both the scanned object and the system detector are discrete. The attenuation coefficient function $\mu(x, y)$ is discretized into an image of $N = N_x \times N_y$ picture elements (pixels), with values $f_{i,j}, \forall i \in 1, \dots, N_x, j \in 1, \dots, N_y$, which can be re-ordered in a vector f .

The detector is made of n_p recording units of length $\delta_x \mu m$; hence, at each X-ray shot, n_p is the number of measured data. Figure 7 depicts a graphical example of the discrete CT configuration where $N_x = N_y = 4$ and $n_p = 7$. The whole scan is constituted by N_θ projections acquired at equally spaced θ_k angles, $\forall k = 1, \dots, N_\theta$, and performed in the angular range $[-\Theta, +\Theta]$. Let $N_d = N_\theta \cdot n_p$ be the total number of data: in classical CT $N_d \gg N$, while $N_d < N$ in case of sparse tomography.

Fixing the k -th projection (acquired from the θ_k -th angled position) and calling m_i the photon counting measured at the i -th recording unit (with $i \in 1, \dots, n_p$), from equation (7), it is possible to define:

$$g_i = -\ln\left(\frac{m_i}{m_0}\right) \quad \forall i \in 1, \dots, n_p. \tag{11}$$

The line integral of equation (7) can be discretized into a sum over all the pixels; hence:

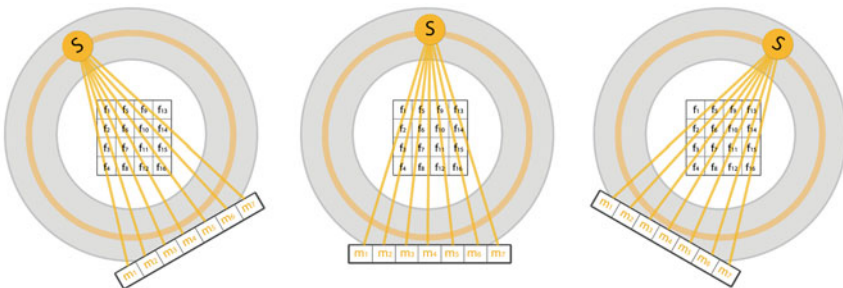


Fig. 7 Scheme of the scanning process for three different angled projections. The sources rotate around the 2D object along a circular trajectory. The slice of interest is discretized into $N = 16$ pixels and the detector has $n_p = 7$ recording units

$$g_i = \sum_{j=0}^N M_{i,j}^{\theta_k} f_j \quad \forall i \in 1, \dots, n_p. \tag{12}$$

In matrix-vector notation, equation (12) becomes:

$$g^{\theta_k} = M^{\theta_k} f \tag{13}$$

where M^{θ_k} is a matrix of size $n_p \times N$ and $g^{\theta_k} = \{g_i\}_{i=1, \dots, n_p}$ is a vector collecting the projections obtained from the angle θ_k (hence g^{θ_k} is the discretization of (9)).

Collecting together all the equations (13) for $k = 1, \dots, N_\theta$, the following large size linear system is obtained:

$$\begin{bmatrix} \text{-----} M^{\theta_1} \text{-----} \\ \text{-----} M^{\theta_2} \text{-----} \\ \text{-----} M^{\theta_3} \text{-----} \\ \text{-----} M^{\theta_4} \text{-----} \\ \dots \\ \dots \\ \dots \\ \dots \\ \text{-----} M^{\theta_{N_\theta}} \text{-----} \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ f_N \end{bmatrix} = \begin{bmatrix} g^{\theta_1} \\ g^{\theta_2} \\ g^{\theta_3} \\ g^{\theta_4} \\ \dots \\ \dots \\ \dots \\ \dots \\ g^{\theta_{N_\theta}} \end{bmatrix} \tag{14}$$

Equation (14) represents the discretization of the Radon transform (10). Using a more compact notation, the CT process is described by the linear system:

$$Mf = g \tag{15}$$

where $M \in \mathbb{R}^{N_d} \times \mathbb{R}^N$, $f \in \mathbb{R}^N$, and $g \in \mathbb{R}^{N_d}$.

Implementation of the Forward Operator M

The most crucial issue in the discrete formulation concerns the computation of the matrix coefficients $M_{i,j}$: although very simple in principle, elaborate computer algorithms and a significant amount of computer time are required to determine its entries. Really, the matrix M is the mathematical description of the physical process of CT data acquisition; hence, it must mirror the forward projection of a slice onto the detector units, for all the scanning views. We recall that M is obtained by collecting the matrices M^{θ_k} corresponding to the single projections at angle θ_k , $k = 1, \dots, N_\theta$ as in (14). Different algorithms have been proposed in literature

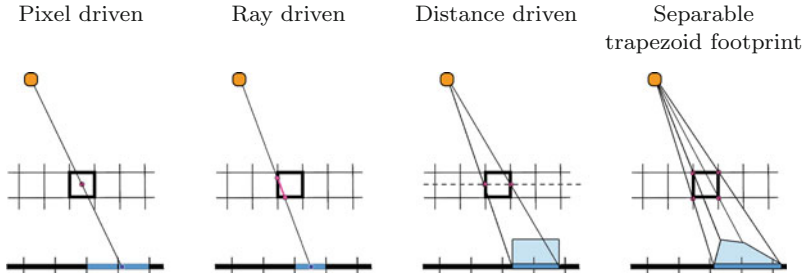


Fig. 8 2D schemes of different approaches to compute the projection matrix M

to efficiently compute the value $M_{i,j}^{\theta_k}$ as the contribution of the object element f_j onto the detector unit g_i . The most common are *pixel-driven*, *ray-driven*, *distance-driven*, and *separable trapezoid footprints*. Figure 8 schematically draws the idea behind each approach.

Historically, the first proposed approach has been the *pixel-driven* (Peters 1981) one: according to the geometry of the device, the f_j pixel is projected from its center onto the element g_i of the detector; its contribution is split among the adjacent measuring units with a linear (or more complex) interpolation routine (Harauz and Ottensmeyer 1983; Fessler 1997). When the spatial resolution of the reconstruction is much bigger than the detector cell size, too few rays are taken into account, and it may happen that some detector cells do not receive any values at all (which is, of course, unrealistic).

In the *ray-driven* (or ray-casting) approach (Lacroute and Levoy 1994; Matej et al. 2004; O'Connor and Fessler 2006), only a straight line is considered reaching the center of each detector unit g_i from the source, and then for each element f_j crossed by the line, $M_{i,j}^{\theta_k}$ is proportional to the length of segment intersecting f_j .

In the *distance-driven* approach, proposed by De Mann in 2002, the idea is to project onto the detector, for each element f_j , not only a point but the element in-plane expansion. This provides a linear shadow, enlarged for the height of f_j , creating a rectangular footprint over one or more detector elements. For each element g_i , the value of $M_{i,j}^{\theta_k}$ is proportional to the area of the portion of rectangle built on it. An extension of the distance-driven algorithm to the 3D case is presented for the case study on limited-angle tomography in a following section.

The *separable trapezoid footprint* algorithm was introduced in 2010 by Long and Fessler. In this method, all the vertices of the element f_j are projected onto the detector, and the element footprint is approximated by a trapeze, to shape a more accurate footprint than in the distance driven case.

The last two methods better model the physical nature of X-ray beams; hence, they compute more accurate projection matrices at the expense of a higher computational cost. All these approaches are conceptually straightforward to be generalized to the 3D case.

Some final considerations about the matrix M implementing the forward operator:

- M is a very sparse matrix, because few pixels are effective for a single value of a projection; hence, each row has mostly zero elements;
- M is under-determined in case of few views; hence, no unique solution exists for the linear system (15);
- M cannot be stored because of its huge dimensions, neither in sparse form, for most of the real CT imaging: whenever we need a matrix product, M must be recalculated element by element and this represents a noticeable computational effort.

The Optimization Framework

In case of sparse-view CT, the linear system (15) is under-determined ($N \gg N_d$); hence, it has infinite possible solutions. Moreover, due to the ill-posedness of the inverse problem and to the lack of data, unwanted artifacts corrupt the solutions.

The model-based approach is introduced to overcome these numerical controversies, by adding some a priori information. The resulting formulation can be stated as a minimization problem involving a data-fitting function \mathcal{F} and a prior operator \mathcal{R} (acting here as a regularizer). The optimization framework is flexible and can be stated as an unconstrained minimization on the objective function \mathcal{J} as:

$$\arg \min_x \mathcal{J}(f) = \mathcal{F}(f) + \lambda \mathcal{R}(f) \quad (16)$$

where $\lambda \geq 0$ is a regularization parameter, or as a constrained minimization:

$$\arg \min_f \mathcal{R}(f) \quad s.t. \quad \mathcal{F}(f) \leq \epsilon^2. \quad (17)$$

or

$$\arg \min_f \mathcal{F}(f) \quad s.t. \quad \mathcal{R}(f) \leq \sigma^2, \quad (18)$$

where $\epsilon \geq 0$ and $\sigma \geq 0$ are estimates of the noise and of the value of $\mathcal{R}(f)$ in the object, respectively.

A meaningful physical constraint to impose is the non-negativity of the solution which reflects the non-negativity property of the linear attenuation coefficient μ ; hence, model (16) could be reinforced as:

$$\arg \min_{f \geq 0} \mathcal{J}(f) = \mathcal{F}(f) + \lambda \mathcal{R}(f). \quad (19)$$

A detailed overview of model-based methods can be found in Graff and Sidky (2015).

Common choices for $\mathcal{F}(f)$ are the least squares (LS) function:

$$LS(f) = \|Mf - g\|_2^2 \quad (20)$$

or the weighted least squares (WLS) function (Thibault et al. 2007):

$$WLS(f) = \left\| \sum_{i=1}^{N_d} W_i (Mf - g)_i \right\|_2^2 \quad (21)$$

where W_i are positive weights.

Focusing on the regularization $\mathcal{R}(f)$, different functions have been proposed in literature. The most widely used convex regularizer in sparse-view CT is the total variation (TV) defined as (Vogel 2002):

$$TV(f) = \sum_{j=1}^N \|\nabla f_j\|_2. \quad (22)$$

or in its smoothed differentiable form:

$$TV_\beta(f) = \sum_{j=1}^N \sqrt{\|\nabla f_j\|_2^2 + \beta^2} \quad (23)$$

where β is a small positive parameter (Vogel 2002). The TV function is chosen by many authors because of its excellent shape recovering and denoising properties, even if it is known that it can produce staircasing effects when the regularization parameter is too high (Sidky et al. 2009; Choi et al. 2010; Ritschl et al. 2011; Hashemi et al. 2013; Graff and Sidky 2015; Luo et al. 2017). Alternative choices preserving convexity are the total generalized variation (TGV) (Niu et al. 2014), the weighted TV (Yu and Zeng 2014), the normal-dose induced non-local means filter (Huang et al. 2013), and the tight frame (Jia et al. 2011) regularizers. Recently, an $l1/l2$ regularizer has been proposed in Wang et al. (2021).

To reduce the TV oversmoothing, also the non-convex and non-differentiable TpV regularization function:

$$TpV(f) = \|\nabla f\|_p^p, \quad 0 \leq p \leq 1 \quad (24)$$

has been proposed (Sidky et al. 2014).

Iterative Algorithms for Optimization

For the solution of the minimization problem expressed in one of the formulations (16), (17), (18), and (19), a suitable optimization algorithm is used. For clinical applications, not only an accurate reconstruction but also a low computational time is required. Hence, the optimization algorithm should meet the following demands:

- have a fast error decreasing in the initial iterations;
- have a low computational cost per iteration, to efficiently run the solver in a short time;
- have a limited request of memory, to solve real-size problems on commercially affordable hardware. For this reason first-order descent methods are generally preferred to methods exploiting second-order information, which require further storage space.

Various iterative methods have been considered and efficiently used in CT reconstruction, such as the scaled gradient projection (Loli Piccolomini and Morotti 2016; Loli Piccolomini et al. 2018) and alternate directions of multipliers method (ADMM) (Wang et al. 2021) for the solution of a convex problem or the proximal dual hybrid gradient (PDHG, also known as Chambolle-Pock) in the non-convex case (Sidky et al. 2014). A new method accelerating both ADMM and PDHG has been recently proposed in Liu et al. (2021).

Regularization: Little or Too Much?

Both the unconstrained (16) and constrained (17), (18), and (19) minimization formulations depend on a parameter: λ , ϵ , or σ . The amount of regularization on the solution depends on the choice of this parameter.

To investigate the effects of regularization on the reconstructed image, a dataset freely downloadable from the web page of the Finnish Inverse Problems Society www.fips.fi/dataset.php is considered (the relative documentation can be found in Bubba et al. 2016). The object in exam is a lotus root (see Fig. 9) which has been filled with several objects of different shapes, sizes, and attenuation coefficients.

The scanning process consists in 120 fan-beam projections, performed from a circular trajectory with angular step size $\Delta_\theta = 3$ degrees; each real projection array has been downsampled into 429 recorded values; hence, the sinogram is a data matrix of size 429×120 and it is shown in Fig. 9. The dataset also provides the forward projector, as a sparse matrix of size $51,480 \times 65,536$; hence, the reconstruction will be an image of 256×256 pixels.

The reconstructions in Fig. 10 are obtained with the minimization model (19) setting $\mathcal{F}(f)$ as the LS function (20) and $\mathcal{R}(f)$ as the TV_β function defined in (23) (with $\beta_{TV} = 10^{-3}$). The images are computed with different values of the

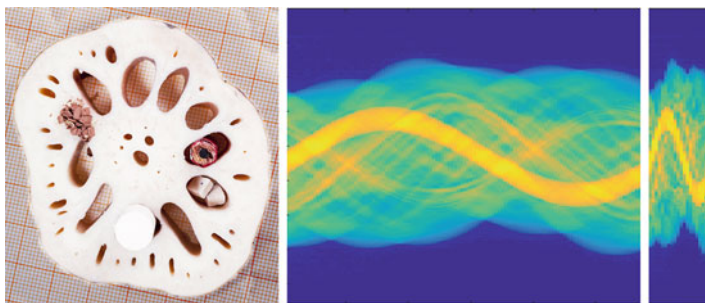


Fig. 9 On the left: a picture of the lotus root, filled with different materials. At the center: the sinogram of the lotus dataset with 120 sparse projections. On the right: the sinogram with 20 highly sparse views

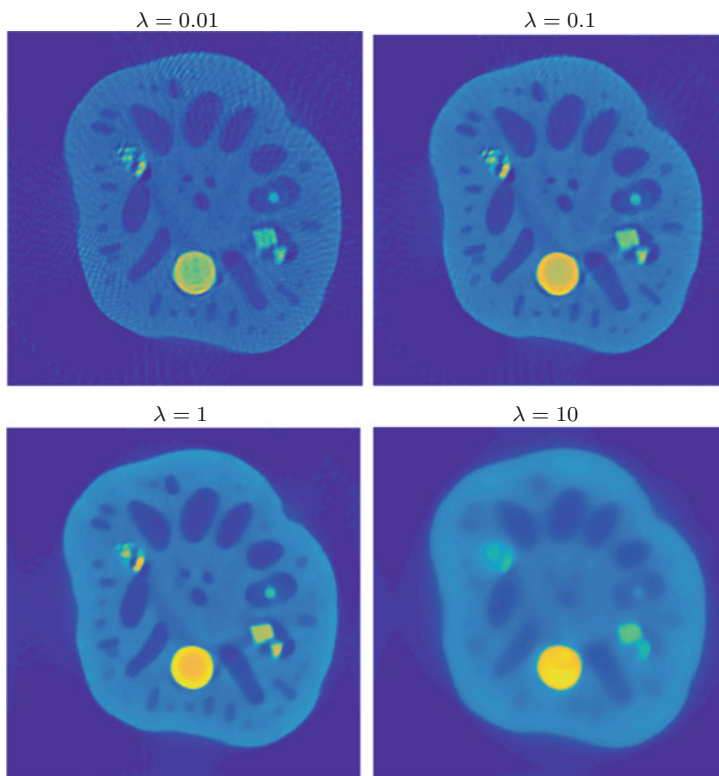


Fig. 10 Results achieved at convergence, for increasing values of the regularization parameter $\lambda = 0.01, 0.1, 1, 10$

regularization parameter λ , getting the increasing values $\lambda = 0.01, 0.1, 1, 10$. The artifacts visible in the reconstruction with lowest value of λ disappear when the regularization parameter increases. However, a too large value of λ blurs the image as shown in the bottom row of Fig. 10.

Toward the Convergence of the Iterative Method

Figure 11 reports the lotus images reconstructed at 10, 50, and 100 iterations and at convergence (about 1000 iterations) using the sinogram with 120 projections over 360 degrees. The regularization parameter λ is set to 1 in all the tests. From the zoomed crops aside each reconstructed image, it is visible how the objects of interest are better enhanced and detected with increasing iterations. It is also evident that after very few iterations, the contours of the objects are defined, whereas more iterations are necessary to obtain a good contrast. Moreover, Fig. 11 confirms that the chosen model well approximates the desired image and that the iterative method is converging toward the problem solution. In practice, the more iterations are executed, the better will be the reconstructed image.

Finally, some considerations about the model when applied to a sparser geometry can be deduced from Fig. 12, where the images are reconstructed from only 20

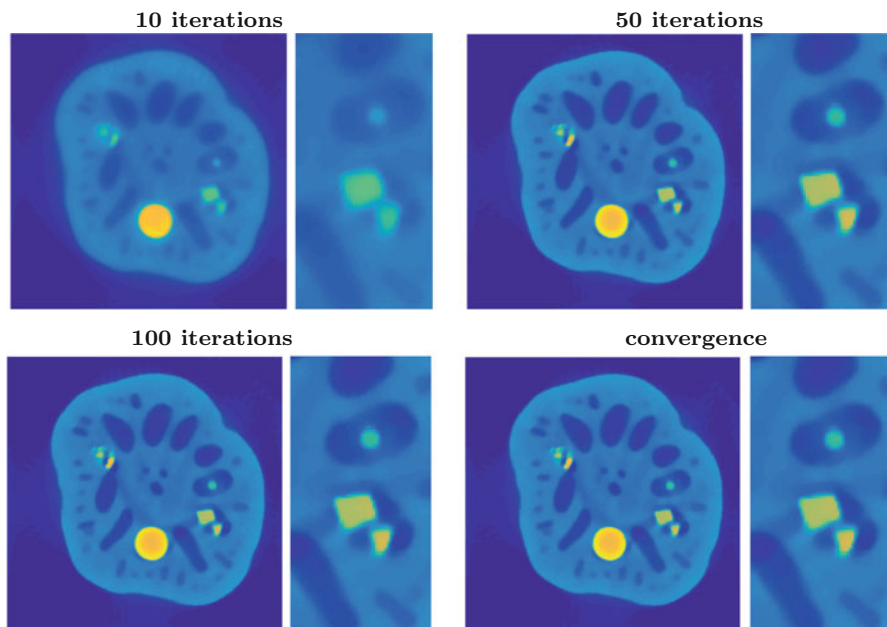


Fig. 11 Results obtained with $\lambda = 1$ at 10, 50, and 100 iterations and at convergence, from 120 projections

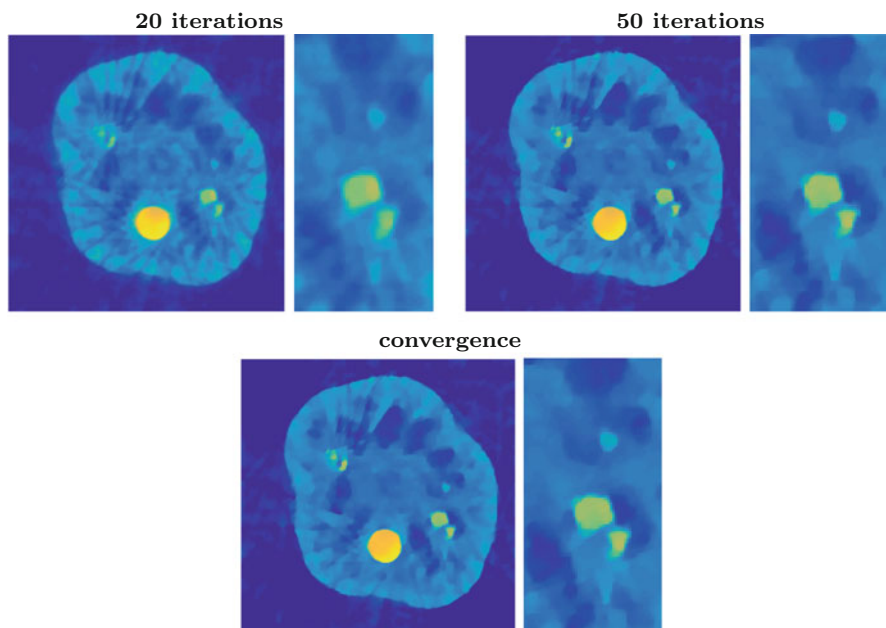


Fig. 12 Results obtained with $\lambda = 1$ at 20 and 50 iterations and at convergence, from 20 projections

projections over 360 degrees (with angular step of 18 degrees) in 20 and 50 iterations and at convergence (145 iterations).

In this case of very sparse-view full-angle CT, some artifacts are present in all the reconstructions, and more iterations must be performed to achieve reasonable results, compared to the previous geometry with many more projections. In 20 iterations, not all the objects are detectable and they have low contrast with the background. However, increasing the iterations enhances the images better, and the results obtained at the algorithm convergence are very promising.

By the way, these tests show the importance of running the reconstructing solvers for a longer time, when the CT problem is characterized by a severe subsampling, and it mirrors the difficulty to back-project the dataset and fit it, in case of few tomographic projections.

New Frontiers of CT Reconstruction with Deep Learning

Since few years ago, deep learning (DL)-based methods have emerged over fully conventional or variational approaches for sparse-view tomographic reconstruction (Wang et al. 2018). In the first experiments, neural networks have been mainly used as a postprocessing tool to remove artifacts and noise from fast reconstructions

(typically obtained with analytical solver, as FBP). Such approach is usually called learnt postprocessing (LPP). Here the network learns from a set of *ground truth* images reconstructed from full-dose acquisitions (see, e.g., Han and Ye (2018), Pelt et al. (2018), Zhang et al. (2019), Schnurr et al. (2019), Urase et al. (2020), Morotti et al. (2021) and the references therein). However, in their inspiring work (Sidky et al. 2020), Sidky et al. have claimed that the popular LPP schemes lack of mathematical characterization and a new framework has been recently proposed in Evangelista et al. (2022) to face this drawback.

Neural networks have been also introduced into model-based schemes to improve their efficiency. In the so-called *unrolling* (or unfolding) strategies, each iteration is executed by a layer of the neural network which learns, in the training phase, some parameters of the optimization algorithm (Monga et al. 2021). The proposals differ for the considered iterative scheme and for the block-per-iteration learned by the neural network. For instance, in 2017, Adler and Öktem have developed a partially learned gradient descent algorithm, whereas they have worked on the Chambolle-Pock scheme in Adler and Öktem (2018). In Gupta et al. (2018) a convolutional neural network is trained to act like a projector in a gradient descent algorithm, whereas in Xiang et al. (2021) both the proximal operator and gradient operator of an unrolled FISTA scheme are learned. In Zhang et al. (2020) the neural network learns the initial iterate of the inner conjugate gradient solver in a splitting scheme for optimization. A different approach is constituted by the plug-and-play scheme. In this case, the minimization problem is solved by a splitting optimization method, such as ADMM, and the neural network is plugged in the denoising substep of the method at each iteration (Venkatakrishnan et al. 2013; He et al. 2018).

Case Study: Reconstruction of Digital Breast Tomosynthesis Images

Digital breast tomosynthesis (DBT) is a quite recent development of the mammographic imaging system for breast tumor detection. DBT, in fact, provides a volumetric breast reconstruction as a stack of 2D images, each representing a cross-sectional slice of the breast itself (Cavicchioli et al. 2020). The detection of breast cancer by mammography suffers from the obscuring effect of overlapping breast tissue, due to the projection onto a flat image of all the breast volume: the cancer can be masked by surrounding overlapping structures, especially in woman with radiographically dense breasts. On the contrary, DBT has the advantage of separating the anatomical tissues, and this generally reduces false-negative diagnosis (Fig. 13). At the same time, DBT provides a low radiation dose (comparable to the radiation dose used in one standard mammography), since the X-ray source emits only few projecting cone beams from few angled points along a narrow C-shaped trajectory. In 2011, the Food and Drug Administration (the federal agency of the US Department of Health and Human Services) recommended the DBT technique over mammography as breast cancer screening in the USA, due to its established

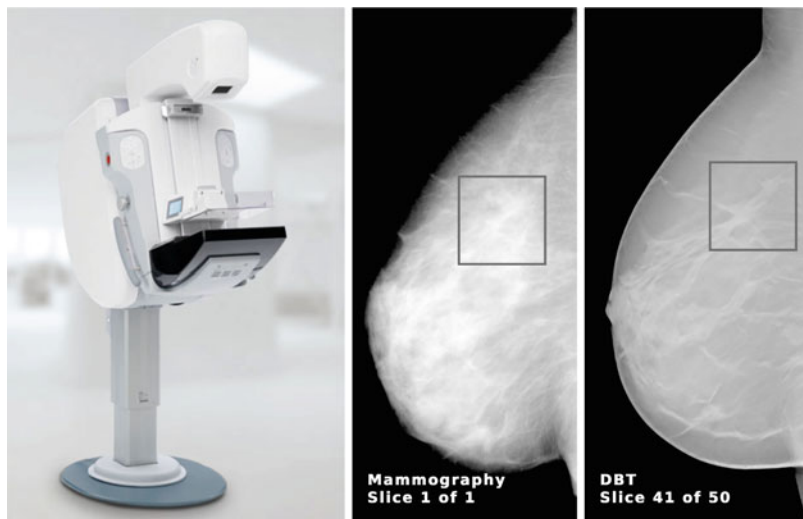


Fig. 13 A modern DBT device on the left and a comparison between a 2D mammographic image and a DBT image slice, showing the same spiculated mass

higher accuracy in the most important breast diagnostic imaging tasks, i.e., finding microcalcifications and suspected masses (Andersson et al. 2008; Das et al. 2010).

DBT 3D Imaging

DBT puts into practice a limited-angle sparse tomographic protocol for three-dimensional imaging; hence, its image reconstruction is not trivial technically. As schematically reported in Fig. 14 where a Cartesian axis system is introduced for clarity, in a modern DBT machinery, the X-ray source moves on the YZ -plane, drawing an arc which spans 11 to 60 degrees typically (hence $\Theta \approx 5$ to 30 degrees, according to the notation previously introduced). From equally spaced angled points on such trajectory, $N_\theta = 9 - 25$ projection images are acquired by the detector. The detector is flat, built as a $n_x \times n_y$ grid of recording units with a uniform sensitive area of $\delta_x \times \delta_y \mu\text{m}^2$. Typically, δ_x and δ_y are 85–160 μm . Moreover, the detector is fixed on a XY -plane and stationary during the whole scanning process.

The breast volume is numerically discretized into $N_v = N_x \times N_y \times N_z$ volumetric elements (called *voxels*) of size $\Delta_x \times \Delta_y \times \Delta_z \mu\text{m}^3$. Due to the high resolution of the projection images, DBT allows for very high in-plane resolution (i.e., the resolution on the reconstructed slices which are parallel to the detector plane): Δ_x and Δ_y are smaller than 0.1 mm. On the contrary, because of the severe narrowness of the scanning range $[-\Theta, +\Theta]$, DBT is unfeasible to reconstruct thin slices as classical CT and its Z -axis resolution Δ_z is 1 to 1.5 mm typically.

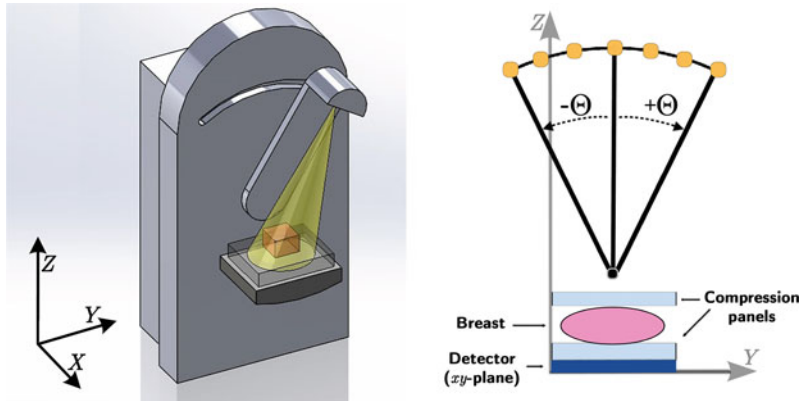


Fig. 14 On the left, a sketch of a modern DBT device where the Cartesian axis system is added for clarity. On the right, a view of the DBT geometry, projected onto the YZ -plane

In contrast to classical medical CT, DBT also makes use of *soft X-rays* with few tens of electron volts: this choice helps to reduce the provided radiations and it is further motivated by the anatomical structure of the breast. In breast imaging, there are no bones nor metallic objects, but adipose and fibro-glandular tissues that have very low attenuating properties: breast materials would not capture many photons from high-radiation X-rays. Since much more photon scattering occurs, this choice provides noisier data; nevertheless, it also allows to detect the breast objects in a more distinguishable way.

A further relevant feature of DBT imaging is due to its actual use in hospitals and clinics, where the high frequency of DBT screening tests makes long executions too expensive for a variety of reasons. As a consequence, an iterative solver can perform few iterations and it is stopped far before its convergence, typically. Such disadvantage is partially alleviated by parallel implementations (Jia et al. 2010; Matenine et al. 2015; Cavicchioli et al. 2020), but as the allowed computational time is shorter than 1 min, the huge amount of data and the complexity of the matrix computation make only four or five iterations feasible.

Model and Analysis

TV-Based Framework

All the following reconstructions are computed as solutions of the non-negative constrained and differentiable optimization problem:

$$\arg \min_{f \geq 0} \mathcal{J}(f) = LS(f) + \lambda TV_{\beta}(f). \quad (25)$$

As solver, the scaled gradient projection (SGP) method, which is a gradient descent-like algorithm, is used (Loli Piccolomini et al. 2018). It is a first-order accelerated method, already proposed in Loli Piccolomini and Morotti (2021) for real 3D subsampled tomography. Essentially, the method follows a gradient projection approach accelerated by choosing the step lengths with Barzilai-Borwein techniques and by introducing a suitable scaling matrix improving the matrix conditioning. Its convergence to the unique minimum of (25) is proved in Bonettini and Prato (2015), under feasible assumptions. Numerically, the SGP solver runs until the following stopping condition on the objective function \mathcal{J} is satisfied by an iterate $f^{(k)}$:

$$\left| \frac{f(x^{(k)}) - f(x^{(k-1)})}{f(x^{(k)})} \right| < 10^{-6}. \quad (26)$$

A comparison among different solvers is out of the scope of this paper. However, results on the same data with different iterative methods can be found in Loli Piccolomini and Morotti (2021).

Measure and Graphics of Merits for 3D Tomography

To quantitatively evaluate the digitally reconstructed objects of interest, two widely used measure of merits are used in literature: the contrast-to-noise ratio (CNR) and the full width at half maximum (FWHM).

The CNR measure on a mass is calculated as:

$$CNR_{MS} = \frac{\mu_{MS} - \mu_{BG}}{\sigma_{MS} - \sigma_{BG}} \quad (27)$$

where μ and σ are the mean and standard deviation computed on the reconstructed volume, in small regions located inside the mass (MS) or in the background (BG). Similarly, the CNR measure on a microcalcification is defined as:

$$CNR_{MC} = \frac{m_{MC} - \mu_{BG}}{\sigma_{BG}} \quad (28)$$

where m_{MC} is the maximum intensity inside the considered microcalcification (MC). Higher values of the CNR indices reflect a better detection of an object from the background.

To compute the FWHM parameter, the transverse slice (parallel to the XY -plane) where the microcalcification lies must be considered, and then it is required to extract the plane profile (PP) of the MC, along the Y direction. The FWHM index is thus computed as:

$$FWHM = 2\sqrt{2 \ln(2)}d \quad (29)$$

where d is the standard deviation of the Gaussian curve fitting the PP. In particular,

$$w = FWHM \cdot \Delta_y \quad (30)$$

approximates the width of the examined microcalcification. The plane profiles are also useful tools to evaluate the reconstruction accuracy on the transverse plane.

To estimate the solver effectiveness along the Z direction, which is the most challenging purpose in DBT imaging, it is convenient to extract the artifact spread function (ASF) vector from the digital reconstruction. The ASF components are computed on a microcalcification as:

$$ASF(z) = \frac{|\mu_{MC}(z) - \mu_{BG}(z)|}{|\mu_{MC}(\bar{z}) - \mu_{BG}(\bar{z})|}, \quad \forall z = 1, \dots, N_z \quad (31)$$

where $\mu(z)$ is the mean of the reconstructed values inside a circular region of three pixels diameter inside the considered MC and in the background, \bar{z} corresponds to the slice where the object is on focus, and N_z is the total number of discrete slices. Similarly, we compute the ASF for the masses.

Reconstructions of the Accreditation Phantom

The tests here reported are performed on the *Giotto Class* digital system by the Italian I.M.S. Giotto Spa company in Bologna ([IMS Giotto Class](#)). To get the considered data, the source executes $N_\theta = 11$ scans from equally spaced angles in an approximately 30-degree range. The detector has squared pixel pitch of $85 \mu\text{m}$, whereas the reconstructed voxel dimensions along the three Cartesian axes are $\Delta_x = \Delta_y = 90 \mu\text{m}$ and $\Delta_z = 1 \text{ mm}$, respectively.

The scanned object is a breast imaging phantom, the model 020 of BR3D, produced by CIRS Tissue Simulation and Phantom company ([Computerized Imaging Reference Systems](#)). It is characterized by a heterogeneous background, where adipose-like and gland-like tissues are mixed in about 50:50 ratio. Inside, objects of interest for breast cancer detection are inserted at the same depth: they are acrylic spheres simulating breast masses (MSs), acrylic short fibers, and clusters of calcium carbonate specks simulating microcalcifications (MCs), of different dimensions and thickness.

Running the gradient descent solver, the convergence criterion (26) stops the execution after 44 iterations. The fast decreasing behavior of the \mathcal{J} function along the iterative reconstruction process is remarkable, as visible in Fig. 15. The objective function exhibits a very fast reduction in the first five iterations, whereas it has a very flat trend from ten iterations on, as confirmed by the red-labeled values. Indeed, the reconstructed images are visually almost indistinguishable after 30 iterations.

Figure 16 presents the reconstructions of a 4.7 mm mass and of a cluster with the 165- μm -thick MCs, obtained in 5, 15, and 30 iterations. In Fig. 17 the corresponding PP and ASF plots are reported.

Simulated and anatomical masses are larger than microcalcifications, but their lower photon absorption capability makes their detection difficult. In fact, even if

Fig. 15 Objective function values vs. iteration number for the iterative reconstruction of the phantom test. The red labels outline the function values at 5, 15, and 30 iterations

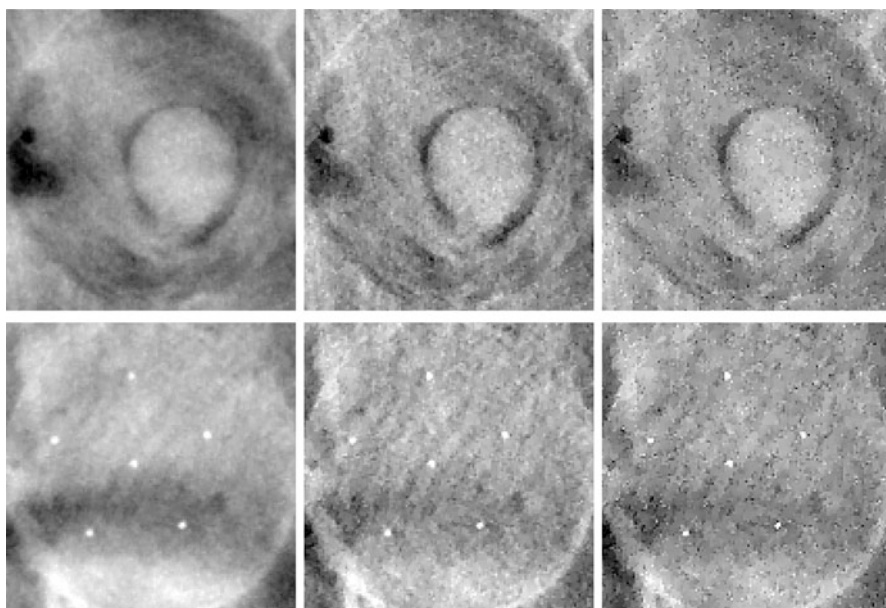
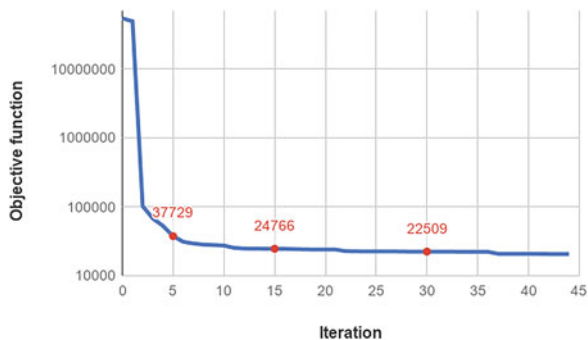


Fig. 16 Crops of a reconstructed slice on BR3D phantom, obtained in 5, 15, and 30 iterations (from left to right). First row: zooms in of a mass. Last row: zooms in of a MC cluster

visible in only five iterations, the mass tends to present smooth edges, and more iterations are required to enhance the mass contrast to the background (see the first row of Fig. 16 and the corresponding plane profile in Fig. 17). The perfect location of the mass at its correct depth still remains critical, since it tends to be out of focus and blurred along the Z direction.

In spite of their smallness, microcalcifications are immediately visible on the earliest model-based reconstructions, as high absorbing structures of a breast. In fact, all the six MCs of the reported cluster are clearly detected in only five iterations, but again the effect of the TV regularization needs longer executions to make them less blurry and more contrasted from the background. It is remarked by the PP

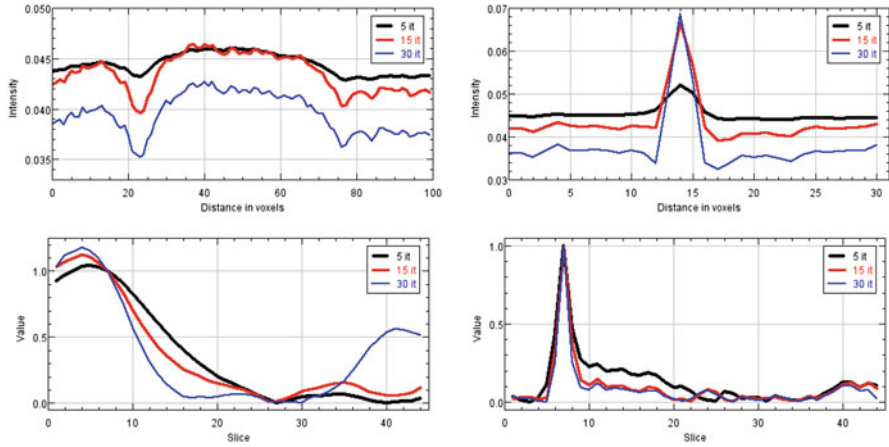


Fig. 17 Plane profiles (first row) and ASF plots (second row) of the mass and one microcalcification from the BR3D phantom reconstructions shown in Fig. 16. In all the plots: black line corresponds to five iterations, red line to 15 iterations, and blue line to 30 iterations

Table 1 FWHM index (29) and w measures (30) computed on images reconstructed in 5, 15, and 30 SGP iterations. In the first column are the actual diameters of the microcalcification spheres of the BR3D phantom

Diameters (μm) of the MC	FWHM			w (μm)		
	5 it.	15 it.	30 it.	5 it.	15 it.	30 it.
230	4.77	3.32	2.70	430	299	243
165	3.52	2.65	2.32	317	238	209
130	–	2.05	1.52	–	185	137

plots of Fig. 17. Even the object detection along the Z axis improves with ongoing iterations, as deductible from the depth-oriented inspection by the ASF plot. The FWHM values and the corresponding MCs width w (reported in Table 1) denote that the regularized iterative approach is indeed effective in recovering very small microcalcifications: MCs of $130 \mu\text{m}$ width, which should approximately fill inside only two voxels, are not discernible from the background in only five iterations (the FWHM is not measurable here), but they can be well recovered after more iterations with a good approximation of their real size.

At last, the increasing values of CNR in Table 2 denote the strong effect of the regularized model in denoising the objects of interest.

Reconstructions of a Human Dataset

The performances of an iterative model-based reconstruction are further confirmed when it is used on real screening DBT datasets. For example, the considered breast contains here a microcalcification and a mass, on the same reconstructed slice, and

Table 2 CNR measure for microcalcifications as in (28) and for masses as in (27) computed on images reconstructed in 5, 15, and 30 SGP iterations. In the first column are the actual diameters of the considered objects of the BR3D phantom

	Diameters (μm)	5 it.	15 it.	30 it.
MS	4700	0.82	1.07	1.66
MS	3100	0.87	1.00	1.33
MC	230	24.21	33.34	38.00
MC	165	10.03	19.00	28.00
MC	130	7.27	11.02	17.00

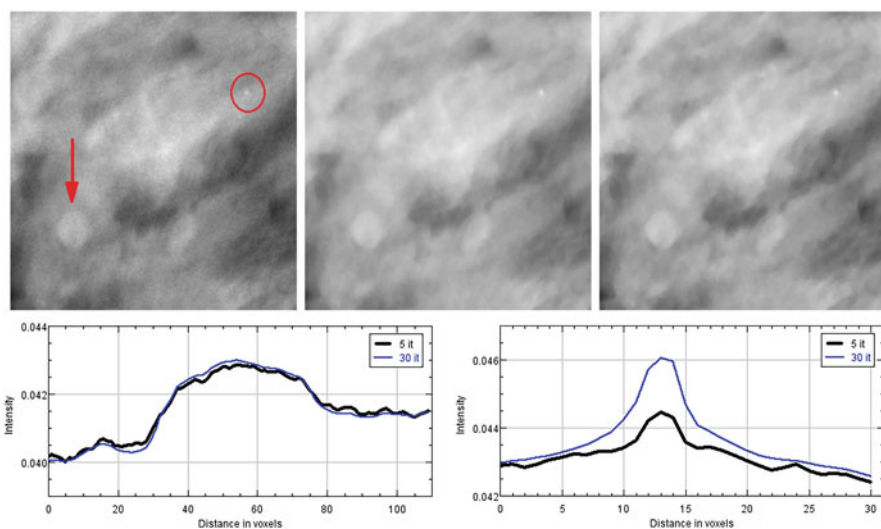


Fig. 18 Results obtained after 5, 15, and 30 SGP iterations on a human breast dataset. First row: zooms in of a 440×400 pixels region presenting both a spherical mass (pointed by the arrow) and a microcalcification (identified by the circle). Last row: plane profiles on the mass and on the microcalcification. In the plots: black line corresponds to 5 iterations and blue line to 30 iterations

the images in Fig. 18 zoom over such objects of interest on the reconstructions computed in 5, 15, and 30 iterations. Figure 18 also shows the plot profiles of the mass and the microcalcification. In this case, the mass detection is already effective in the earliest reconstruction and its gray level intensity does not change remarkably, but the denoising effect of the TV prior in the last iterations is evident on the PP. Also the microcalcification is detected in few iterations, even if a more time-consuming SGP execution enhances the contrast of the object with respect to the background and the corresponding FWHM values (reported in Table 3) confirm its getting more and more defined, from 5 to 30 iterations.

Table 3 FWHM measures on the microcalcification visible in Fig. 18

	FWHM		
	5 it.	15 it.	30 it.
MC	8.57	7.81	7.29

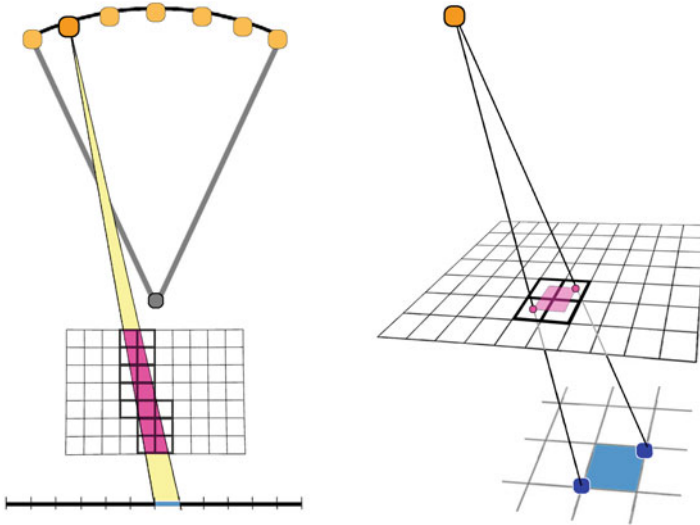


Fig. 19 The distance-driven approach for the forward projection on a DBT-like device. On the left, the process is seen on the YZ -plane (hence the detector is reduced to a 1D array and the volume to a grid of voxels); on the right, one volume slice is considered over the detector. In all the images, one detector unit is considered and remarked in blue, whereas its backward projection cone is highlighted in pink, defining the voxels that are indeed involved in the forward projections

Distance-Driven Approach for 3D CT Imaging

In DBT, the projection matrix M can be efficiently computed with a distance-driven (DD) approach. The standard DD extension to 3D imaging is presented in De Man and Basu (2004) for a general CT process, but due to the presence of a flat and stationary detector, it is necessary to specifically tune the algorithm for DBT devices.

Recalling the notation used in this chapter, $\Delta_x, \Delta_y,$ and Δ_z are the spatial resolution of the discretization into voxels of the volume, respectively, along the Cartesian axis, whereas δ_x, δ_y are the dimensions of each detector unit mounted on the DBT machinery. For prefixed scanning angle θ_k and element g_i of the k -th projection (where $k \in \{1, \dots, N_\theta\}, i \in \{1, \dots, n_p\}$), the i -th row of the forward projection operator M^{θ_k} models the X-ray cone having as a basis the i -th detector pixel itself and vertex on the X-ray source (see Fig. 19 as reference); then the DD algorithm determines the backward footprints of the detector unit onto each object slice, at its middle height $\frac{\Delta_z}{2}$ (as indicated on the left image in Fig. 20). Fixing one object slice and the j -th voxel on it (as the green one in Fig. 20), let A_i be the area of

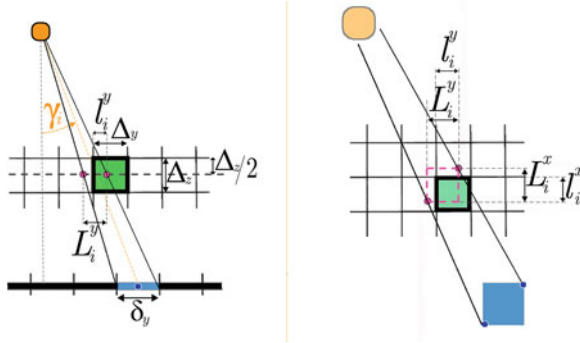


Fig. 20 The distance-driven approach for the forward projection on a DBT-like device. On the left, the process is seen on the YZ -plane for one slice; on the right, it is projected onto the XY -plane. In all the images, one detector unit is considered and remarked in blue, whereas its backward projection area is highlighted in pink, and the considered voxel is green

the backward footprints onto the slice (dashed pink in Fig. 20) of the i -th pixel and $a_{i,j}$ be the area of the intersection between the pink and the gray squares, as denoted in Fig. 20. The matrix element is computed as:

$$M_{i,j}^{\theta_k} = \frac{\Delta_z}{\alpha_i \gamma_i} \frac{a_{i,j}}{A_i} \tag{32}$$

for all the voxels on that slice.

In this equation, α_i and γ_i are the in- and out-of-plane angles, respectively, i.e., the two angles describing the X-ray linking the source to the center of the i -th detector pixel. Drawing the perpendicular from the X-ray linking the source to the center of the i -th pixel, γ_i is the angle between these two elements on the YZ -plane, as shown in Fig. 20, while α_i is that angle on the XY -plane. Moreover, the factor $\frac{\Delta_z}{\alpha_i \gamma_i}$ can be interpreted as a normalization by the popular $\frac{1}{r^2}$ term, which is known as the *inverse-square* physical law, stating that a specific physical quantity (like the photon intensity in our case) is inversely proportional to the square of the distance from the source of that physical quantity.

Code Parallelization

The required accuracy on the breast digital volume and the resolution of the detector make the DBT problem of very high dimensions. The magnitude of the involved numerical objects prevents the storage of the system matrix M on the hardware; hence, its entries must be computed at each invocation of the matrix itself. This causes an extremely long execution of the optimization solver (which also impacts on the number of iterations allowed in a real clinical setting). In fact, by profiling a serial execution of an iterative solver, two main kernels can be identified as heavy

computational tasks in each iteration, and they are the forward and the backward applications of the matrix operator, i.e., the steps with the matrix-vector products involving M and M^T , respectively.

To set a realistic example, consider a volume with $N = 1.5 \cdot 10^8$ voxels to be recovered from $N_\theta = 11$ views of 3000×1500 pixel projection images (resulting in $N_d \approx 5 \cdot 10^7$ data). Table 4 reports the output of the profiling analysis of the scaled gradient projection algorithm, compiled on an i7 high-end computer with 32 GB of RAM and 1 TB of solid state disk (Cavicchioli et al. 2020). In such a configuration, almost 90% of the computational time is spent for forward and backward projections in a gradient descent solver, where both the kernels occur only once per iteration. A third task addressing all the computations for the TV function covers 5% of the execution time per iteration, whereas only the 8% is spent for all the remaining SGP steps.

By parallelizing the C code on NVIDIA GPU by means of the CUDA SDK, the execution times drastically go down: GPU implementation exploits the massive parallel architecture of graphical boards and distributes work to hundreds of small cores. However, if the algorithm cannot store all the necessary variables in the GPU memory entirely, many data transfers between the CPU and the GPU are required during each iteration of the solver (see Fig. 21): as visible from the second row of

Table 4 Results of the profiling of the iterative solver, according to its different implementations on a CPU (Intel i7 7700K CPU at 4.3 GHz, 32 GB of RAM, and 1 TB of solid state disk) and on the Titan V board by NVIDIA (12 GB of RAM and 5120 CUDA cores). In each row: the computing time of the four considered kernels, the whole iteration time, the number of feasible iterations in 50 s, and the resulting speedup (with respect to the serial implementation). All the times are relative to a single iteration of a gradient descent-like solver and are expressed in milliseconds

	Forward (ms)	Backward (ms)	TV (ms)	Other (ms)	1 iter. (ms)	Iters. in 50 s	Speedup
Serial	235,368	237,556	23,841	39,735	536,500	–	–
Parallel on CPU	270	263	1229	7613	9375	5	57 ×
Parallel on GPU	116	110	372	548	1146	50	468 ×

Fig. 21 Logical view of a system composed by host and accelerator. The data stored in the host memory (DRAM) must be transferred to the graphics card memory (global memory) to execute the parallel computation, and then the results must be transferred back to be saved in DRAM

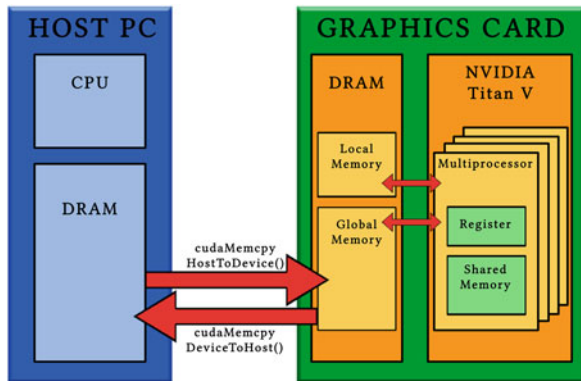


Table 4, the resulting parallel execution achieves a $57\times$ speedup with respect to the serial one, allowing for only five iterations in less than 1 min. On the contrary, if the GPU has a larger global memory, a higher level of parallelism can be exploited to completely run the SGP solver on the GPU so that one iteration requires about only 1 s (reflecting an impressive speedup of almost 470). This means to achieve a close-to-convergence reconstruction in less than 1 min.

Conclusion

Nowadays the medical world aims at enlarging the class of CT exams with new, safe, and fast X-ray protocols, which can be defined by reducing the number of projection views. Model-based iterative methods are efficient methods for sparse-view CT image reconstruction, since they solve an optimization problem where a priori information are embedded by means of a regularization function. When approaching convergence, iterative solvers achieve very accurate images where low-contrast objects and very small structures are well detected and shaped. On a case study on real projections of 3D breast tomosynthesis, model-based approaches reconstruct in very few iterations images where the objects of interest, such as masses and microcalcifications, are clearly distinguishable. Moreover, a parallel reconstruction of breast imaging on a GPU board can be obtained from real data in less than 1 min, a time compatible with clinical requests.

Indeed, if the main drawback of iterative solver lays in their high computational costs and slow executions, the ongoing development of GPU boards (which are more and more powerful and affordable) paves the way to almost real-time reconstructions, making this approach feasible for real-life applications.

Finally, the flexibility of the optimization framework also allows to incorporate external information by means of neural networks to improve the quality of the reconstructed image.

References

- Computerized Imaging Reference Systems: <https://www.cirsinc.com/products/a11/51/br3d-breast-imaging-phantom/>. BR3D Breast Imaging Phantom, Model 020
- IMS Giotto Class: <http://www.imsgiotto.com/>
- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**(12), 124007 (2017)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Andersson, I., Ikeda, D.M., Zackrisson, S., Ruschin, M., Svahn, T., Timberg, P., Tingberg, A.: Breast tomosynthesis and digital mammography: a comparison of breast cancer visibility and birads classification in a population of cancers with subtle mammographic findings. *Eur. Radiol.* **18**(12), 2817–2825 (2008)
- Averbuch, A., Sedelnikov, I., Shkolnisky, Y.: CT reconstruction from parallel and fan-beam projections by a 2-d discrete radon transform. *IEEE Trans. Image Process.* **21**(2), 733–741 (2011)

- Barca, P., Lamastra, R., Tucciariello, R., Traino, A., Marini, C., Aringhieri, G., Caramella, D., Fantacci, M.: Technical evaluation of image quality in synthetic mammograms obtained from 15° and 40° digital breast tomosynthesis in a commercial system: a quantitative comparison. *Phys. Eng. Sci. Med.* **44**(1), 23–35 (2021). cited By 0
- Bonettini, S., Prato, M.: New convergence results for the scaled gradient projection method. *Inv. Probl.* **31**(9), 1196–1211 (2015)
- Bubba, T.A., Hauptmann, A., Huotari, S., Rimpeläinen, J., Siltanen, S.: Tomographic x-ray data of a lotus root filled with attenuating objects. *arXiv preprint arXiv:1609.07299* (2016)
- Buzug, T.M.: Computed tomography. In: *Springer Handbook of Medical Technology*, pp. 311–342. Springer, Muller and Siltanen, Philadelphia(2011)
- Cavicchioli, R., Hu, J., Loli Piccolomini, E., Morotti, E., Zanni, L.: A first-order primal-dual algorithm for convex problems with applications to imaging. GPU acceleration of a model-based iterative method for digital breast tomosynthesis. *Sci. Rep.* **10**(1), 120–145 (2020)
- Choi, K., Wang, J., Zhu, L., Suh, T.-S., Boyd, S.P., Xing, L.: Compressed sensing based cone-beam computed tomography reconstruction with a first-order method. *Med. Phys.* **37**(9), 5113–5125 (2010)
- Das, M., Gifford, H.C., O'Connor, J.M., Glick, S.J. Penalized maximum likelihood reconstruction for improved microcalcification detection in breast tomosynthesis. *IEEE Trans. Med. Imaging* **30**(4), 904–914 (2010)
- De Chiffre, L., Carmignato, S., Kruth, J.-P., Schmitt, R., Weckenmann, A.: Industrial applications of computed tomography. *CIRP Ann.* **63**(2), 655–677 (2014)
- De Man, B., Basu, S.: Distance-driven projection and backprojection. In: *2002 IEEE Nuclear Science Symposium Conference Record*, vol. 3, pp. 1477–1480. IEEE (2002)
- De Man, B., Basu, S.: Distance-driven projection and backprojection in three dimensions. *Phys. Med. Biol.* **49**(11), 2463 (2004)
- Evangelista, D., Morotti, E., Piccolomini, E.L.: Rising a new framework for few-view tomographic image reconstruction with deep learning. *arXiv preprint arXiv:2201.09777* (2022)
- Feldkamp, L.A., Davis, L.C., Kress, J.W.: Practical cone-beam algorithm. *J. Opt. Soc. Am. A* **1**(6), 612–619 (1984)
- Fessler, J.A.: Equivalence of pixel-driven and rotation-based backprojectors for tomographic image reconstruction (1997)
- Graff, C., Sidky, E.: Compressive sensing in medical imaging. *Appl. Opt.* **54**(8), C23–C44 (2015)
- Gupta, H., Jin, K.H., Nguyen, H.Q., McCann, M.T., Unser, M.: CNN-based projected gradient descent for consistent CT image reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1440–1453 (2018)
- Hadamard, J.: Sur les problèmes aux dérivées partielles et leur signification physique, pp. 49–52. Princeton University Bulletin, Natterer, Stuttgart (1902)
- Han, Y., Ye, J.C.: Framing U-NET via deep convolutional framelets: application to sparse-view CT. *IEEE Trans. Med. Imaging* **37**(6), 1418–1429 (2018)
- Harauz, G., Ottensmeyer, F.: Interpolation in computing forward projections in direct three-dimensional reconstruction. *Phys. Med. Biol.* **28**(12), 1419 (1983)
- Hashemi, S., Beheshti, S., Gill, P.R., Paul, N.S., Cobbold, R.S.: Fast fan/parallel beam CS-based low-dose CT reconstruction. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1099–1103. IEEE (2013)
- He, J., Yang, Y., Wang, Y., Zeng, D., Bian, Z., Zhang, H., Sun, J., Xu, Z., Ma, J.: Optimizing a parameterized plug-and-play ADMM for iterative low-dose CT reconstruction. *IEEE Trans. Med. Imaging* **38**(2), 371–382 (2018)
- He, Y., Luo, S., Wu, X., Yang, H., Zhang, B.B., Bleyer, M., Chen, G.: Computed tomography angiography with 3d reconstruction in diagnosis of hydronephrosis cause by aberrant renal vessel: a case report and mini review. *J. X-Ray Sci. Technol.* **26**(1), 125–131 (2018)
- Huang, J., Zhang, Y., Ma, J., Zeng, D., Bian, Z., Niu, S., Feng, Q., Liang, Z., Chen, W.: Iterative image reconstruction for sparse-view CT using normal-dose image induced total variation prior. *PLoS One* **8**(11), e79709 (2013)

- Hughes, S.: CT scanning in archaeology. In: Saba, L. (ed.) *Computed Tomography-Special Applications*, pp. 57–70. InTech Europe, Buzug, Berlin (2011)
- Jia, X., Dong, B., Lou, Y., Jiang, S.B.: GPU-based iterative cone-beam CT reconstruction using tight frame regularization. *Phys. Med. Biol.* **56**(13), 3787 (2011)
- Jia, X., Lou, Y., Li, R., Song, W.Y., Jiang, S.B.: GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation. *Med. Phys.* **37**(4), 1757–1760 (2010)
- Kak, A.C., Slaney, M.: *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia (2001)
- Kubo, T., Lin, P.-J.P., Stiller, W., Takahashi, M., Kauczor, H.-U., Ohno, Y., Hatabu, H.: Radiation dose reduction in chest CT: a review. *Am. J. Roentgenol.* **190**(2), 335–343 (2008)
- Lacroute, P., Levoy, M.: Fast volume rendering using a shear-warp factorization of the viewing transformation. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pp. 451–458 (1994)
- Liu, Y., Xu, Y., Yin, W.: Acceleration of primal–dual methods by preconditioning and simple subproblem procedures. *J. Sci. Comput.* **86**(2), 1–34 (2021)
- Loli Piccolomini, E., Coli, V., Morotti, E., Zanni, L.: Reconstruction of 3D X-ray CT images from reduced sampling by a scaled gradient projection algorithm. *Comput. Optim. Appl.* **71**, 171–191 (2018)
- Loli Piccolomini, E., Morotti, E.: A fast TV-based iterative algorithm for digital breast tomosynthesis image reconstruction. *J. Algorithms Comput. Technol.* **10**(4), 277–289 (2016)
- Loli Piccolomini, E., Morotti, E.: A model-based optimization framework for iterative digital breast tomosynthesis image reconstruction. *J. Imaging* **7**(2), 36 (2021)
- Long, Y., Fessler, J.A., Balter, J.: 3d forward and back-projection for x-ray CT using separable footprints with trapezoid functions. In: *Proceedings of First International Conference on Image Formation in X-Ray Computed Tomography*, pp. 216–219 (2010)
- Luo, X., Yu, W., Wang, C.: An image reconstruction method based on total variation and wavelet tight frame for limited-angle CT. *IEEE Access* **6**, 1–1 (2017)
- Matej, S., Fessler, J.A., Kazantsev, I.G.: Iterative tomographic image reconstruction using fourier-based forward and back-projectors. *IEEE Trans. Med. Imaging* **23**(4), 401–412 (2004)
- Matenine, D., Goussard, Y., Després, P.: GPU-accelerated regularized iterative reconstruction for few-view cone beam CT. *Med. Phys.* **42**(4), 1505–1517 (2015)
- Monga, V., Li, Y., Eldar, Y.C.: Algorithm unrolling: interpretable, efficient deep learning for signal and image processing. *IEEE Sig. Process. Mag.* **38**(2), 18–44 (2021)
- Morotti, E., Evangelista, D., Loli Piccolomini, E.: A green perspective for learned post-processing in sparse-view tomographic reconstruction. *J. Imaging* **7**(8), 139 (2021)
- Mueller, J.L., Siltanen, S.: *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, Huges, Croatia (2012)
- Natterer, F.: *The Mathematics of Computerized Tomography*. SIAM, Hadamard, Princeton (2001)
- Niu, S., Gao, Y., Bian, Z., Huang, J., Chen, W., Yu, G., Liang, Z., Ma, J.: Sparse-view x-ray CT reconstruction via total generalized variation regularization. *Phys. Med. Biol.* **59**(12), 2997 (2014)
- O’Connor, Y., Fessler, J.A.: Fourier-based forward and back-projectors in iterative fan-beam tomographic image reconstruction. *IEEE Trans. Med. Imaging* **25**(5), 582–589 (2006)
- Pelt, D.M., Batenburg, K.J., Sethian, J.A.: Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks. *J. Imaging* **4**(11), 128 (2018)
- Peters, T.: Algorithms for fast back-and-re-projection in computed tomography. *IEEE Trans. Nucl. Sci.* **28**(4), 3641–3647 (1981)
- Quinto, E.T.: Singularities of the x-ray transform and limited data tomography in R^2 and R^3 . *SIAM J. Math. Anal.* **24**(5), 1215–1225 (1993)
- Reiser, I., Bian, J., Nishikawa, R.M., Sidky, E.Y., Pan, X.: Comparison of reconstruction algorithms for digital breast tomosynthesis. arXiv:0908.2610 (2009)
- Ritschl, L., Bergner, F., Fleischmann, C., Kachelrieß, M.: Improved total variation-based CT image reconstruction applied to clinical data. *Phys. Med. Biol.* **56**(6), 1545–1561 (2011)

- Schnurr, A.-K., Chung, K., Russ, T., Schad, L.R., Zöllner, F.G.: Simulation-based deep artifact correction with convolutional neural networks for limited angle artifacts. *Zeitschrift für Medizinische Physik* **29**(2), 150–161 (2019)
- Sidky, E., Chartrand, R., Boone, J., Pan, X.: Constrained TpV-minimization for enhanced exploitation of gradient sparsity: application to CT image reconstruction. *IEEE J. Transl. Eng. Health Med.* **2**, 1800418 (2014)
- Sidky, E.Y., Kao, C.M., Pan, X.: Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. *J. Xray Sci. Technol.* **14**(2), 119–139 (2009)
- Sidky, E.Y., Lorente, I., Brankov, J.G., Pan, X.: Do cnns solve the CT inverse problem? *IEEE Trans. Biomed. Eng.* **68**(6), 1799–1810 (2020)
- Thibault, J.-B., Sauer, K.D., Bouman, C.A., Hsieh, J.: A three-dimensional statistical approach to improved image quality for multislice helical CT. *Med. Phys.* **34**(11), 4526–4544 (2007)
- Urase, Y., Nishio, M., Ueno, Y., Kono, A.K., Sofue, K., Kanda, T., Maeda, T., Nogami, M., Hori, M., Murakami, T.: Simulation study of low-dose sparse-sampling CT with deep learning-based reconstruction: usefulness for evaluation of ovarian cancer metastasis. *Appl. Sci.* **10**(13), 4446 (2020)
- Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948. IEEE (2013)
- Vogel, C.R.: *Computational Methods for Inverse Problems*. SIAM, Philadelphia (2002)
- Wang, C., Tao, M., Nagy, J.G., Lou, Y.: Limited-angle CT reconstruction via the l_1/l_2 minimization. *SIAM J. Imaging Sci.* **14**(2), 749–777 (2021)
- Wang, G., Ye, J.C., Mueller, K., Fessler, J.A.: Image reconstruction is a new frontier of machine learning. *IEEE Trans. Med. Imaging* **37**(6), 1289–1296 (2018)
- Wu, T., Moore, R.H., Rafferty, E.A., Kopans, D.B.: A comparison of reconstruction algorithms for breast tomosynthesis. *Med. Phys.* **31**(9), 2636 (2004)
- Xiang, J., Dong, Y., Yang, Y.: Fista-net: learning a fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Trans. Med. Imaging* **40**(5), 1329–1339 (2021)
- Yu, L., Liu, X., Leng, S., Kofler, J.M., Ramirez-Giraldo, J.C., Qu, M., Christner, J., Fletcher, J.G., McCollough, C.H.: Radiation dose reduction in computed tomography: techniques and future perspective. *Imaging Med.* **1**(1), 65 (2009)
- Yu, W., Zeng, L.: A novel weighted total difference based image reconstruction algorithm for few-view computed tomography. *PLoS One* **9**(10), e109345 (2014)
- Zhang, H., Liu, B., Yu, H., Dong, B.: MetaInv-net: meta inversion network for sparse view CT image reconstruction. *IEEE Trans. Med. Imaging* **40**(2), 621–634 (2020)
- Zhang, T., Gao, H., Xing, Y., Chen, Z., Zhang, L.: Dualres-UNET: limited angle artifact reduction for computed tomography. In: 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), pp. 1–3. IEEE (2019)
- Zhang, Y., Chan, H.H.-P., Sahiner, B., Wei, J., Goodsitt, M., Hadjiiski, L.M.L., Ge, J., Zhou, C.: A comparative study of limited-angle cone-beam reconstruction methods for breast tomosynthesis. *Med. Phys.* **33**(10), 3781 (2006)



Recent Approaches for Image Colorization

15

Fabien Pierre and Jean-François Aujol

Contents

Context and Modeling	586
Challenge	586
Mathematical Modeling of Colorization	587
Range of Chrominance	588
Color Diffusion	589
State-of-the-Art of Color Diffusion	590
Coupled Total Variation for Image Colorization	592
Constrained TV-L2 Debiasing Algorithm	594
Exemplar-Based Colorization	599
Morphing-Based Approach	600
Segmentation-Based Techniques	601
Patch-Based Methods	602
A Variational Model for Image Colorization with Channel Coupling	605
Colorization from Dataset	607
Coupled Approaches	608
Coupling Manual Approach with Exemplar-Based Colorization	609
Coupling CNN with a Variational Approach	611
Conclusion and Future Works	618
References	619

F. Pierre (✉)

LORIA, UMR CNRS 7503, Université de Lorraine, INRIA projet Tangram, Nancy, France
e-mail: fabien.pierre@loria.fr

J.-F. Aujol (✉)

Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, Talence, France
e-mail: jean-francois.aujol@math.u-bordeaux.fr

Abstract

In the last years, image and video colorization has been considered from many points of view. The technique consists of the addition of a color component to a grayscale image. This operation needs additional priors which can be given by manual intervention of the user from an example image or be extracted from a large dataset of color images. A very large variety of approaches has been used to solve this problem, like PDE models, non-local methods, variational frameworks, learning approaches, etc. In this chapter, we aim at providing a general overview of state-of-the-art approaches with a focus on few representative methods. Moreover, some recent techniques from the different types of priors (manual, exemplar-based, dataset-based) are explained and compared. The organization of the chapter aims at describing the evolution of the techniques in relation to each other. A focus on some efficient strategies is proposed for each kind of methodology.

Keywords

Image colorization · Variational approaches · Deep learning · Patch-based methods

Context and Modeling

Challenge

Image colorization consists of the transformation of a grayscale image into a color one. The reverse transformation, i.e., turning a color image into a grayscale one, is based on visual assumptions and it is also an active research topic (Kuhn et al. 2008; Cui et al. 2010; Song et al. 2013). Image colorization is useful for the entertainment industry to make old film productions attractive to young people, for instance. In France, in 2014, *Apocalypse*, a historical documentary by I. Clarke and D. Costelle, was made from archives colorized by F. Montpellier of the *ImaginColor* company. The broadcast gathered over 18.5% of viewers over the age bracket 11–14 during the first 2 episodes (Lannaud 2009). The colorization for movies is mostly performed manually, which is a very tedious work. As an example, the colorization of about 4 hours of video sequences for the *Apocalypse* documentary required 47 weeks by F. Montpellier and his team. Image colorization can also be used to help a user to analyze an image, for example, for sensor fusion in Zheng and Essock (2008). For instance, to assist in airport security screening, color is added to the X-ray scanner result based on the density of the objects, so that the operator can know their composition and quickly interpret the result (Abidi et al. 2006). Image colorization can also be used to restore artistic heritage, for example, Fornasier (2006) or Wolfgang Baatz Massimo Fornasier and Schönlieb (2008). This old subject started with the ability of screens to display color. A first approach, very basic, consists of matching each grayscale to a color (Gonzalez and Woods 2008). However, it is impossible to recover every color without additional information

(there are 256 gray levels and about 16 million colors displayable on standard screens). In existing approaches, this information can be added by three ways: the first one directly adds color to the image by the user (see, e.g., the approach of Levin et al. 2004), the second one provides an example image (also called source image, see, e.g., the method of Welsh et al. 2002), and the third one uses a deep learning approach based on a large database (see for instance the method of Zhang et al. 2016).

In this chapter, we propose a general overview of colorization methods which have been described in the literature with a focus on few representative approaches. This review is not based on the application point of view but it has been done from a methodological perspective. The term “automatic” has been widely used, but it means in fact that the algorithms are able to assist the user. For manual methods, the diffusion of the colors put by the user is automatic, and for exemplar-based approaches, the diffusion of colors from a given reference image to the target one is automatic but actually it requires the choice of the source image. For dataset-based colorization, the colorization is automatic after training on a large dataset given by the user. In this chapter, an overview of the three different approaches to colorize images (manual, exemplar-based, and dataset-based) is proposed. In particular, a highlight on a variational model is used as a thread along the chapter because this model enables some coupling of different approaches such as manual with exemplar-based. More generally, we focus on the different strategies available among state-of-the-art methods for each kind of methodology. Moreover, a final section proposes an overview of coupled strategies.

In this chapter, the mathematical modeling of the colorization problem is reviewed in section “[Mathematical Modeling of Colorization](#)”. Next, in section “[Range of Chrominance](#)”, we recall the definition of the range of the solution, and we present an algorithm to compute an orthogonal projection onto this set. The three next sections deal with, respectively, the manual, the exemplar-based, and the dataset-based colorization. Finally, in section “[Coupled Approaches](#)”, we propose an overview about the coupling of some techniques within a variational formulation.

Mathematical Modeling of Colorization

In order to model the colorization problem, let us consider the luminance-chrominance color spaces. The results of this section are based on the papers (Pierre et al. 2015c, 2017b) that can be considered as the state of the art for luminance specification. In all state-of-the-art approaches, the grayscale image is considered as the luminance channel of a color image. The luminance can be defined as a weighted average of the RGB channels:

$$Y = 0.299R + 0.587G + 0.114B. \quad (1)$$

Some other definitions are also sometimes used. For instance, the L channel of the CIE Lab color space can be used. In order to preserve its content, colorization methods must always require that the luminance channel of the image of interest is equal

to the target image. Most methods compute only the two chrominance channels, complementary to the luminance, which is enough to provide a displayable color image.

Some different spaces have been introduced, such as YUV, YCbCr, YIQ, etc. The transformation from RGB to YUV is linear and defined with the following matrices:

$$R, G, B, Y \in [0, 255], U \in [-111.18, 111.18], V \in [-156.825, 156.825]. \quad (2)$$

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0, 299 & 0, 587 & 0, 114 \\ -0, 14713 & -0, 28886 & 0, 436 \\ 0, 615 & -0, 51498 & -0, 10001 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (3)$$

Let us notice that the main problem raised by these color spaces is that all the luminance-chrominance values cannot be converted into a RGB color between 0 and 255. Thus, some additional techniques have to be employed to recover the RGB color image (Pierre et al. 2015c). These techniques are out of the scope of this chapter, but the reader has to keep in mind that they are essential to compute the final result. The next section recalls the basis of gamut problem in the case of the YUV color space.

Range of Chrominance

The natural problem arising when editing a color while keeping its luminance or intensity constant is the preservation of the RGB standard range of the produced image. Most of the methods of the literature work directly in the RGB space (Nikolova and Steidl 2014; Fitschen et al. 2015; Pierre et al. 2015c), since it is easier to maintain the standard range. Nevertheless, working in the RGB space needs to process three channels, while two chrominance channels are enough to edit a color image while keeping the luminance.

Description of the Range

In this section, we recall the geometric description of the set of chrominance values which correspond to a particular luminance level and which are contained in the RGB standard range. Let us denote by $T(y, u, v)$ the invertible linear operator mapping YUV colors onto the RGB ones. This operator corresponds to the inverse of the operation described in Equation (3).

Proposition 1. *Let y be a value of luminance between 0 and 255. The set of chrominance values (u, v) that satisfy $T(y, u, v) \in [0, 255]^3$ is a convex polygon.*

Remark 1. For a given luminance, the chrominance values out of this polygon can be transformed into the RGB space, but they are out of the bounds of the RGB cube. A truncation of the coordinates is usually done, but it generally changes both the luminance and the hue of the result.

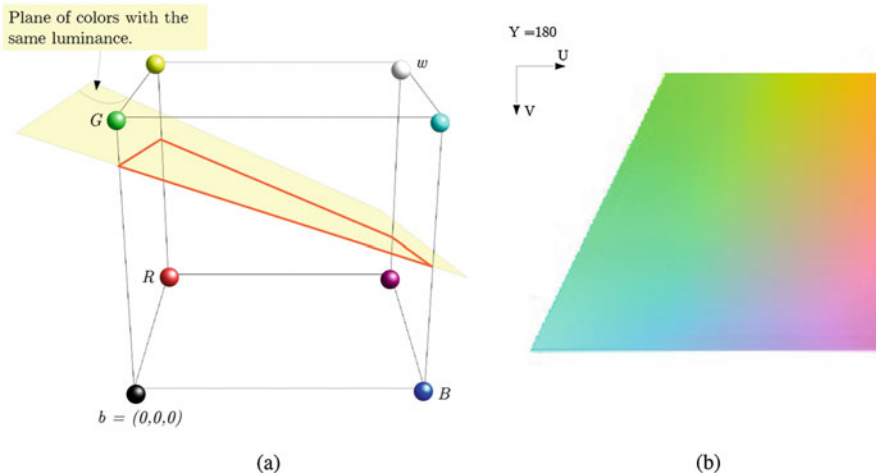


Fig. 1 The set of the RGB colors with a particular luminance is a convex polygon. The map from RGB to YUV being affine, the set of the corresponding chrominances is also a convex polygon. (a) Set of the RGB colors with a fixed luminance. (b) Corresponding colors in the YUV space

Proof. [of Proposition 1] The intuition of the proof is given in Fig. 1. The set of the colors in the RGB cube whose luminance is equal to a particular value y is a convex polygon (see, e.g., Pierre et al. 2015c). Indeed, the set of colors with a particular luminance is an affine plane in \mathbb{R}^3 and the intersection of the RGB cube with it is a polygon. The transformation of the RGB values into the YUV space being affine, the set of corresponding colors is thus also a convex polygon included in the set $Y = y$. □

Orthogonal Projection onto the Convex Range

Pixel-wise, the valid chrominances are contained in a convex polygon that has, at most, six edges. The numerical computation of the vertex coordinates has been detailed in Pierre et al. (2017b). When the vertices are computed, and denoted by P1, P2, etc., the orthogonal projection onto the polygon is computed as follows.

The algorithm first checks if the corresponding RGB value is between 0 and 255. If so, the point is its own orthogonal projection. If not, the orthogonal projection is onto one of the edges and can be computed for each of them. Finally, the closest result is retained as the solution. The algorithm is summarized in Algorithm 1 and illustrated in Fig. 2.

Color Diffusion

In this section, we first summarize the state-of-the-art methods. We then present a strategy for image colorization based on the total variation minimization. This framework uses some recent state-of-the-art approaches in order to diffuse color

Algorithm 1 Algorithm computing projection $P_{\mathcal{R}}$ **Require:** X : chrominance vector; Y luminance value.

```

1: if  $RGB(Y, X) \notin [0, 255]^3$  then
2:   for  $i = 1 : n$  do
3:      $j \leftarrow i + 1 \bmod n$ 
4:      $\alpha \leftarrow \left( \overrightarrow{P_i P_j} | \overrightarrow{P_i X} \right) / \left( \|\overrightarrow{P_i P_j}\|_2 \right)$ 
5:     if  $\alpha > 1$  then
6:        $X_{i,j} \leftarrow P_j$ 
7:     else if  $\alpha < 0$  then
8:        $X_{i,j} \leftarrow P_i$ 
9:     else
10:       $X_{i,j} \leftarrow P_i + \alpha \overrightarrow{P_i P_j}$ 
11:    end if
12:  end for
13:   $X \leftarrow \arg \min_{X_{i,j}} \|X - X_{i,j}\|_2$ 
14: end if

```

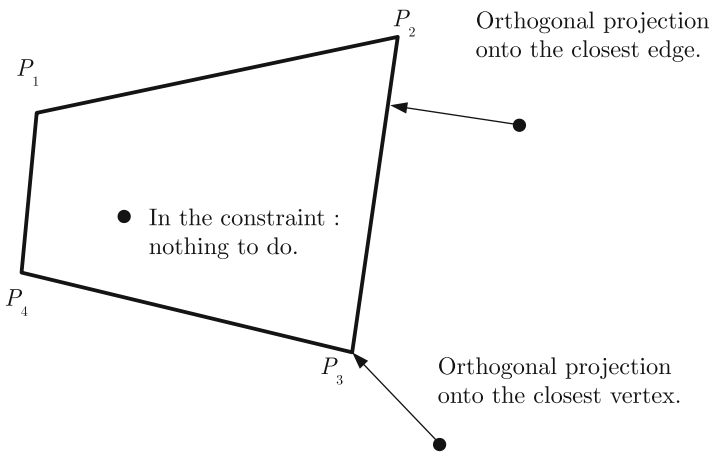


Fig. 2 To compute the orthogonal projection, different cases can appear. If the YUV color respects the constraint, the projection is the identity. Otherwise, the orthogonal projection onto the closest edge or vertex should be done

strokes on grayscale images. We review some work addressing a coupled total variation with a L2 data-fidelity term. Since this estimator is biased, we then review a debiasing strategy that can be applied on this last model.

State-of-the-Art of Color Diffusion

Some papers of the literature aim at helping the user to perform manual colorization. This is done by a diffusion of the colors over the grayscale image by various techniques. The diffusion approaches can also take inspiration from manual colorization

to improve the results of other colorization approaches. In this section, we will describe the diffusion techniques proposed in the literature. This chapter is based on the papers (Pierre et al. 2014b, 2015a, 2017b) that are competitive methods of the literature. Let us remark that there does not exist a perfect diffusion method, all the state-of-the-art approaches having their advantages and drawbacks.

In order to perform manual colorization, a user manually adds color strokes. These are called *scribbles*, and they consist of a set of pixels for which the chrominance channels are defined. Many methods using this process have been proposed. For example, the method of Levin et al. (2004) solves an optimization problem for diffusing scribbles on the target image, assuming that the chrominance must have small variations when the luminance does vary a lot. Specifically, the following functional is minimized:

$$H(U) = \sum_r \left(U(r) - \sum_{r \sim s} w_{rs} U(s) \right)^2, \quad (4)$$

where $r \sim s$ means that pixels r and s are neighbors and U is a chrominance channel (the same functional is minimized for the channel V). w_{rs} denotes the weights which can be either:

$$w_{rs} \propto e^{(Y(r)-Y(s))^2/2\sigma^2},$$

or:

$$w_{rs} \propto 1 + \frac{1}{\sigma_r^2} (Y(r) - \mu_r)(Y(s) - \mu_r),$$

where μ_r and σ_r denote the mean and the variance of the neighborhood of the pixel r . The two types of weights are more or less sensitive to the variation of contrast. The authors of Luan et al. (2007) include texture similarity in the model of Levin et al. (2004) to improve the diffusion process.

The authors of Yatziv and Sapiro (2006) have proposed a simple and fast method using geodesic distance to weight for each pixel the melting of the colors given by the scribbles. For each pixel of the grayscale image, the geodesic distance from the scribble is computed with respect to the gradient of the image. Next, a weighted average of the chrominances given by the scribbles is computed. The weights are computed from a function depending on the geodesic distance. This method enables a diffusion of the chrominance on constant parts of the image with respect to a function having similar properties as the inverse function:

- $\lim_{r \rightarrow 0} w(r) = \infty$;
- $\lim_{r \rightarrow \infty} w(r) = 0$;
- $\lim_{r \rightarrow \infty} w(r + r_0)/w(r) = 1$.

Yatziv et al. have proposed experimental results with the function $\frac{1}{r^b}$ with $1 \leq b \leq 6$.

The authors of Kawulok et al. (2012) have extended this method to textured images by introducing texture descriptors in the diffusion potential.

Some methods are designed as a propagation of the colors from neighbors to neighbors. Some colors are given by strokes drawn by the user. In this way, some of the image pixels are colored. The algorithm then propagates the color to their neighbors with a rule based on the values of the grayscale image. To this aim, the authors of Heu et al. (2009) give an explicit formula for melting the neighbor colors, whereas the ones of Lagodzinski and Smolka (2008) provide a modeling based on probabilistic distance transform, and the authors of Kim et al. (2010) use random walks.

It was also proposed to use diffusion through the regularization of non-local graphs. The method proposed by L  zoray et al. (2008) is based on the regularity of the image. This is modeled as a graph, each pixel being represented by a vertex and each neighborhood relationship by an edge. A local graph is considered, where each edge represents a relationship of eight neighborhoods. The weight of an edge being inversely proportional to the difference between gray levels, the minimization of an energy depending on these weights (see, e.g., L  zoray et al. 2007a) enables to diffuse the chrominances on the constant parts of the image. If a non-local graph is designed with a weight which depends on the distance between patches, a set of pixels is considered constant if the patches are similar. Thus, the color of the scribbles is diffused between pixels close in the graph, therefore belonging to similar textures.

Inspired by the PDE diffusion scheme (Perona and Malik 1990), some chrominance diffusion including a guidance with Di Zenzo tensor structure computed from grayscale image was proposed independently by Peter et al. (2017) and by Drew and Finlayson (2011).

The authors of Quang et al. (2010) have proposed a variational approach in chromaticity-brightness color space (see, e.g., Chan et al. 2001) to interpolate the missing colors. The *reproducing kernel Hilbert spaces* (RKHS) are used to compute a link between the chromaticity and brightness channels. Jin et al. (2016) introduced a variational model with the coupling of contour directions. Based on Mumford-Shah-type functional, the authors of Jung and Kang (2016) introduced a novel variational image colorization model. In the following, we present a recent state-of-the-art method based on total variation minimization. This approach enables to combine various strategies of the literature.

Coupled Total Variation for Image Colorization

In the following we focus on a variational model to denoise the chrominance channels of an image while keeping the luminance unchanged. Similarly to the colorization model of Pierre et al. (2015a), we want to find the minimizer $\hat{u}(c)$ of the denoising functional:

$$\hat{u}(c) = \arg \min_{u=(U,V)} \text{TV}_{\mathfrak{C}}(u) + \lambda \int_{\Omega} \|u(x) - c(x)\|^2 dx + \chi_{\mathcal{R}}(u), \quad (5)$$

with

$$\text{TV}_{\mathfrak{C}}(u) = \int_{\Omega} \sqrt{\gamma \|\nabla Y(x)\|^2 + \|\nabla U(x)\|^2 + \|\nabla V(x)\|^2} dx, \quad (6)$$

where Y , U , and V are the luminance and chrominance channels. This term is a coupled total variation which enforces the chrominance channels to have a contour at the same location as the luminance ones. γ is a parameter which enforces the coupling of the channels. Some other total variation formulations have been proposed to couple the channels; see for instance Kang and March (2007) or Caselles et al. (2009).

The fidelity-data term is a classical L^2 norm between chrominance channels of the unknown u and the data c . For each pixel, the chrominance values live onto the convex polygon denoted by \mathcal{R} and described in section “Range of Chrominance”. This last assumption ensures that the final solution lies onto the RGB cube, avoiding the final truncation that leads to modification of the luminance channel. Model (5) is convex and it can be turned into a saddle-point problem of the form:

$$\min_{u \in \mathbb{R}^2} \max_{z \in \mathbb{R}^6} \frac{\lambda}{2} \|u - c\|^2 + \langle \nabla u | z_{1,\dots,4} \rangle + \langle \gamma \nabla Y | z_{5,\dots,6} \rangle - \chi_{B(0,1)}(z) + \chi_{\mathcal{R}}(u). \quad (7)$$

The primal-dual algorithm (Chambolle and Pock 2011) used to compute such saddle-point problem is recalled in Algorithm 2, where $P_{\mathcal{R}}$ is the orthogonal projection described in Algorithm 1 and $P_{\mathcal{B}}$ is defined as follows for one pixel:

$$P_{\mathcal{B}}(z) = \frac{(z_{1,\dots,4}, z_{5,6} - \sigma \nabla Y)}{\max(1, \|z_{1,\dots,4}, z_{5,6} - \sigma \nabla Y\|_2)}. \quad (8)$$

Algorithm 2 Minimization of (7)

- 1: $u^0 = c$
 - 2: $z^0 \leftarrow \nabla u$
 - 3: **for** $n \geq 0$ **do**
 - 4: $z^{n+1} \leftarrow P_{\mathcal{B}}(z^n + \sigma \nabla \bar{u}^n)$
 - 5: $u^{n+1} \leftarrow P_{\mathcal{R}}\left(\frac{u^n + \tau (\text{div}(z^{n+1}) + \lambda c)}{1 + \tau \lambda}\right)$
 - 6: $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
 - 7: **end for**
 - 8: set $\hat{u}(c) = u^{n+1}$ and $\hat{z} = z^{n+1}$.
-

The results produced by Algorithm 2 are promising, but with a low data parameter λ , they are drab (see, e.g., Pierre et al. 2017b).

Constrained TV-L2 Debiasing Algorithm

In this section we present a debiased algorithm for correcting the loss of colorfulness of the solution given by the optimum of (5).

The CLEAR Method (Deledalle et al. 2017)

The CLEAR method (Deledalle et al. 2017) can be applied for debiasing estimators $\hat{u}(c)$ obtained as:

$$\hat{u}(c) \in \arg \min_{u \in \mathbb{R}^p} F(u, c) + G(u), \quad (9)$$

where F is a convex data-fidelity term with respect to the data c and G is a convex regularizer. For G being the total variation regularization, the estimator $\hat{u}(c)$ is generally computed by an iterative algorithm, and it presents a loss of contrast with respect to the data c . In order to debias this estimator, the CLEAR method refits the data c with respect to some structural information contained in the biased estimator \hat{u} . This information is encoded by the Jacobian of the biased estimator with respect to the data c :

$$J_{\hat{u}}(c)d = \lim_{\varepsilon \rightarrow 0} \frac{\hat{u}(c + \varepsilon d) - \hat{u}(c)}{\varepsilon}. \quad (10)$$

For instance, when G is the anisotropic TV regularization, the Jacobian contains the information concerning the support of the solution \hat{u} , on which a projection of the data can be computed.

In general case, the CLEAR method relies on the *refitting estimator* $\mathcal{R}_{\hat{u}}(c)$ of the data c from the biased estimation $\hat{u}(c)$:

$$\mathcal{R}_{\hat{u}}(c) \in \arg \min_{h \in \mathcal{H}} \|h(c) - c\|_2^2 \quad (11)$$

where \mathcal{H} is defined as the set of maps $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ satisfying $\forall c \in \mathbb{R}^n$:

$$h(c) = \hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)), \text{ with } \rho \in \mathbb{R}. \quad (12)$$

A closed formula for ρ can be given:

$$\rho = \begin{cases} \frac{\langle J_{\hat{u}(c)}(\delta) | \delta \rangle}{\|J_{\hat{u}(c)}(\delta)\|_2^2} & \text{if } J_{\hat{u}(c)}(\delta) \neq 0 \\ 1 & \text{otherwise.} \end{cases}, \quad (13)$$

where $\delta = c - \hat{u}(c)$. In practice, the global value ρ allows to recover most of the bias in the whole image domain.

An algorithm is then proposed in Deledalle et al. (2017) to compute the numerical value of $J_{\hat{u}(c)}(c - \hat{u}(c))$. The process is based on the differentiation of the algorithm providing $\hat{u}(c)$.

It is important to notice that the CLEAR method applies well for estimators obtained from the resolution of unconstrained minimization problems of the form (9). Nevertheless, it is not adapted to the denoising problem (5) that contains an additional constraint $\chi_{\mathcal{R}}(u)$ as CLEAR may violate the constraint.

Direct Extension of CLEAR to Constrained Problems

Extending the CLEAR method to the constrained model (5) requires to take the constraint into account in the axioms of the refitting model (11). The main difference with the original model is the addition of the constraint $\chi_{\mathcal{R}}(u)$. We can first notice that the refitting axioms $h(c) = Ac + b$ for some $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ and $J_h(c) = \rho J_{\hat{u}(c)}$ for some $\rho \in \mathbb{R}$ are in line with the introduction of the constraint. In particular, the definition of the Jacobian $J_{\hat{u}}$ in Equation (10) remains valid with the constraint, since $\hat{u}(c)$ and $\hat{u}(c + \varepsilon d)$ are still in the closed convex \mathcal{R} . The computation of the ρ parameter in Equation (13) may nevertheless produce, from Equation (12), an estimation out of the constraint that has to be post-processed. This points out the main difference between the constrained and the unconstrained debiased estimator.

In Deledalle et al. (2017), the value of ρ is computed from the minimization of a map from \mathbb{R} to \mathbb{R} defined as follows:

$$\rho \mapsto \| (I_d - \rho J_{\hat{u}(c)}) (\hat{u}(c) - c) \|_2^2. \quad (14)$$

In the case of the constrained problem, the function to be minimized is written as:

$$\rho \mapsto \|\hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)) - c\|_2^2 + \chi_{\mathcal{R}}(\hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c))). \quad (15)$$

Let us denote by ρ the value defined in Equation (13). In the case when the constraint is fulfilled, i.e., when $\hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)) \in \mathcal{R}$, then the minimum of (15) is reached with ρ .

If not, since function (15) is convex, it is possible to compute explicitly the minimizer. The value $\rho = 0$ is in the domain of the functional because $\hat{u}(c) \in \mathcal{R}$. The idea is to find the maximum value of ρ such that $\hat{u}(c) + \rho J_{\hat{u}(c)}\delta \in \mathcal{R}$. In this case, since \mathcal{R} is a convex polygon, this computation can be done with a ray-tracing algorithm (Williams et al. 2005). To this aim, we can parametrize the segment $[\hat{u}(c), \hat{u}(c) + \rho J_{\hat{u}(c)}\delta]$:

$$\tilde{\rho} = \max_{t \in [0, 1]} t\rho \text{ such that } \hat{u}(c) + t\rho J_{\hat{u}(c)}(c - \hat{u}(c)) \in \mathcal{R}. \quad (16)$$

Equation (16) can thus be directly solved by the maximum value t such that $\hat{u}(c) + t\rho J_{\hat{u}(c)}(c - \hat{u}(c))$ intersects the border of \mathcal{R} .

Direct Debiasing Process

Let us summarize the refitting algorithm designed for model (5). The first step consists of computing a solution of (5) with Algorithm 2. This iterative algorithm provides at convergence a first biased solution $\hat{u}(c)$ and its dual variable \hat{z} . Once this solution has been computed, the differentiated algorithm, presented in Algorithm 3, is applied in the direction $\delta = c - \hat{u}(c)$. This algorithm requires the definition of the operator $\Pi_{\hat{z}}(\tilde{z})$ which is the linearized version of the projection $P_{\mathcal{B}}$ around \hat{z} and which reads (Deledalle et al. 2017):

$$\Pi_{\hat{z}}(\tilde{z}) = \begin{cases} \tilde{z} & \text{if } \|\hat{z}\| < 1 \\ \frac{1}{\|\hat{z}\|} \left(\tilde{z} - \frac{\langle \hat{z}, \tilde{z} \rangle}{\|\hat{z}\|^2} \hat{z} \right) & \text{otherwise.} \end{cases} \quad (17)$$

Finally, the ray-tracing is applied to obtain $\tilde{\rho}$ and get the debiased solution as $\hat{u}(c) + \tilde{\rho} J_{\hat{u}(c)}(c - \hat{u}(c))$.

Algorithm 3 Differentiation of Algorithm 2 for computing $J_{\hat{u}(c)}\delta$ from $(\hat{u}(c), \hat{z})$

- 1: $\tilde{u}^0 = \delta, \tilde{u}^0 = \delta$
 - 2: $\tilde{z}^0 \leftarrow \nabla \tilde{u}$
 - 3: **for** $n \geq 0$ **do**
 - 4: $\tilde{z}^{n+1} \leftarrow \Pi_{\hat{z}}(\tilde{z}^n + \sigma \nabla \tilde{u}^n)$
 - 5: $\tilde{u}^{n+1} \leftarrow \frac{\tilde{u}^n + \tau (\operatorname{div}(\tilde{z}^{n+1}) + \lambda \delta)}{1 + \tau \lambda}$
 - 6: $\tilde{u}^{n+1} \leftarrow 2\tilde{u}^{n+1} - \tilde{u}^n$
 - 7: **end for**
 - 8: $J_{\hat{u}(c)}\delta = \tilde{u}^{n+1}$.
-

Unfortunately, this direct approach does not lead to valuable results on general cases. Indeed, if for one particular pixel the solution $\hat{u}(c)$ is saturated, and if the debiased solution is out of \mathcal{R} , then $\tilde{\rho} = 0$ is the unique global ρ satisfying $\hat{u}(c) + \rho J_{\hat{u}(c)}(c - \hat{u}(c)) \in \mathcal{R}$. Thus, the debiased solution is equal to the biased one, and the debiasing algorithm has no action.

In the next section, we propose a model with an adaptive ρ parameter, depending on the pixel, to tackle this saturated value issue.

Adaptive Debiasing Model for Constrained Problems

For a pixel ω such that $\hat{u}(c)_\omega + \rho J_{\hat{u}(c),\omega}(c_\omega - \hat{u}(c)_\omega)$ fulfills the constraint, ρ is the best value to refit the model according to the hypothesis of model (11). Here, $J_{\hat{u}(c),\omega}$ denotes the value of $J_{\hat{u}(c)}$ in pixel ω .

On the other hand, if for a pixel ω , the values of $\hat{u}(c)_\omega$ and $J_{\hat{u}(c),\omega}(c_\omega - \hat{u}(c)_\omega)$ are such that $\hat{u}(c)_\omega + \rho J_{\hat{u}(c),\omega}(c_\omega - \hat{u}(c)_\omega) \notin \mathcal{R}$, the ρ value has to be adapted. Thus, let us define for a pixel ω the adapted $\tilde{\rho}_\omega$ as follows:

$$\tilde{\rho}_\omega = \max_{t_\omega \in [0,1]} t_\omega \rho \text{ such that } \hat{u}(c)_\omega + t_\omega \rho J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega) \in \mathcal{R}. \quad (18)$$

The constrained refitting model is then defined pixel-wise as:

$$R_{\hat{u}}^{\mathcal{R}}(c) = \hat{u}(c)_\omega + \tilde{\rho}_\omega J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega) \quad (19)$$

This definition ensures that the debiased estimation fulfills the constraint. Moreover, if the debiasing method of Deledalle et al. (2017) produces an estimation that fulfills the constraint, this solution is retained. Notice however that the CLEAR hypothesis $J_h(c) = \rho J_{\hat{u}}(c)$ for some $\rho \in \mathbb{R}$ in model (11) is not fulfilled anymore. In numerical experiments, for most pixels, the values of $\tilde{\rho}_\omega$ computed with this method are the same as with Model (11).

As illustrated by Fig. 3, such a local debiasing strategy realizes an oblique projection onto \mathcal{R} (Figs. 4 and 5).

Computation of the Oblique Projection

In Pierre et al. (2017b), an algorithm used to compute the oblique projection when the constraint is the chrominance set for a particular value of luminance (see, e.g., section “Range of Chrominance”) is proposed. To simplify the notation, the problem is considered for a single pixel ω and one set $u := \hat{u}(c)_\omega$, $c := J_{\hat{u}(c),\omega} (c_\omega - \hat{u}(c)_\omega)$ and $\rho \in \mathbb{R}$ computed by the algorithm of Deledalle et al. (2017).

For $u + \rho c \notin \mathcal{R}$, the maximum value of $t \in [0, 1]$ such that $u + t\rho c \in \mathcal{R}$ is computed. Since $u \in \mathcal{R}$, thus if $u + \rho c \notin \mathcal{R}$, the segment $[u, u + \rho c]$ crosses one edge of the polygon.

One considers this problem by testing it into the RGB space. Indeed, the edges in the chrominance space correspond to edges in the RGB one, and the intersections between them correspond to intersections in the RGB space. In RGB, the problem of finding the intersection between an edge and the polygon is reduced to computing the intersection between the edge and the cube faces because the edges of the polygon are included in the cube by construction (see, e.g., Fig. 1a).

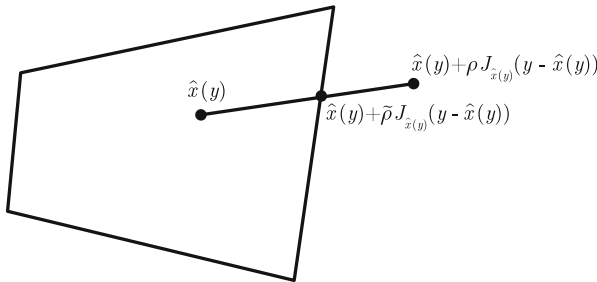


Fig. 3 The refitting of the method of Deledalle et al. (2017) may be out of the constraint. An oblique projection onto this constraint is able to respect most of hypotheses of Model (11) while fulfilling the constraint

The transformation of the chrominance values $u = (U, V)$ to the RGB space with the luminance value Y is denoted by $T_Y(u)$. From the expression of the standard transformation from RGB to YUV, we have $T_Y(u) = Y(1, 1, 1)^t + L(U, V)$ with L a linear function. The following equalities come:

$$\begin{aligned} T_Y(u + \rho c) &= Y(1, 1, 1)^t + L(u + \rho c) \\ &= Y(1, 1, 1)^t + L(u) + \rho L(c) \\ &= T_Y(u) + \rho T_Y(c) - \rho Y(1, 1, 1)^t. \end{aligned} \tag{20}$$

It is required to compute $\tilde{\rho}$ such that $T_Y(u + \tilde{\rho}c)$ is at the boundary of the RGB cube. To this aim, the 6 different values $\tilde{\rho}_c^v$ with $c \in \{R, G, B\}$ and $v \in \{0, 255\}$ corresponding to the cases where the 3 coordinates of $T_Y(u + \tilde{\rho}c)$ are equal to 0 or 255 are computed. For instance, if the first coordinate R of $T_Y(u + \tilde{\rho}c)$ is equal to 255, we have:

$$T_Y(u + \tilde{\rho}_R^{255}c)_R = 255 \tag{21}$$

$$T_Y(u)_1 + \tilde{\rho}_R^{255}T_Y(c)_R - \tilde{\rho}_R^{255}Y = 255. \tag{22}$$

so that

$$\tilde{\rho}_R^{255} = \frac{255 - T_Y(u)_R}{T_Y(c)_R - Y}. \tag{23}$$

For each of the six values $\tilde{\rho}_c^v$ computed as in Equation (23), one can compute $t_c^v = \frac{\tilde{\rho}_c^v}{\rho}$. The values t_c^v that are between 0 and 1 correspond to an intersection of the segment $[u, u + \rho c]$ with the boundaries of \mathcal{R} . One finally takes $t^* = \min_{t_c^v \in [0;1]} t_c^v$ and the result of Equation (18) is given by $t^*\rho$.

Figure 4 and 5 show some numerical results to compare Models (5) and (19). One can remark that Model (5) fits well the contours of images in comparison to the standard TVL2 model on chrominance channels. Moreover, the debiasing approach improves the colorfulness of the results in comparison with Model (5) and it has the advantage of well fitted contours.

To summarize, to design a suitable variational model for image colorization, the three main ingredients are the coupled total variation, the orthogonal projection onto the range of the problem, and the debiasing algorithm. This variational model is a basis for image colorization in many paradigms. In the next sections, some concrete cases of application of this model are presented in the case of exemplar-based approaches or coupled with manual techniques or CNN-based framework.

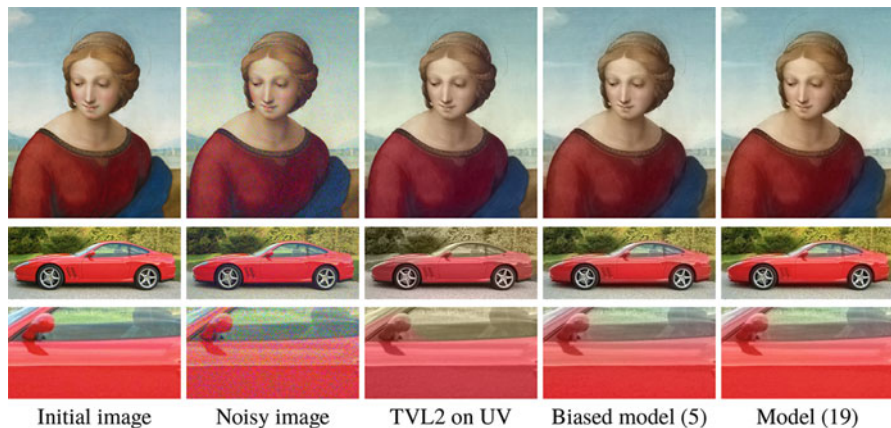


Fig. 4 Results of chrominance channels with a TV-L2 model on chrominance, with the biased method, and with the unbiased method. The debiasing algorithm produces more colorful results

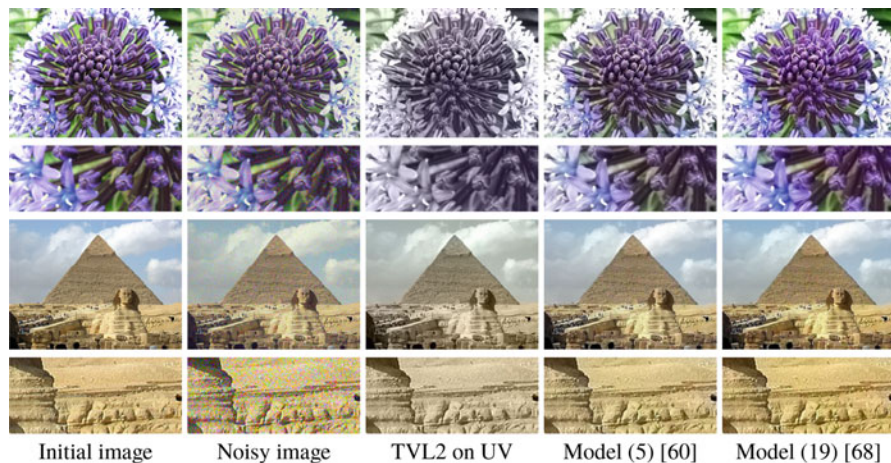


Fig. 5 The advantage of the coupled total variation (5) on the TV-L2 model has been shown in Pierre et al. (2015a). In Pierre et al. (2017b), it is refined in a better colorfulness-preserving model

Exemplar-Based Colorization

The manual methods enable the user to choose the color in each pixel of the image. Nevertheless, their main drawback is the tedious work needed for complex scenes, for instance with textures. In exemplar-based image colorization methods, the color information is provided by a color image called *source image*. The grayscale image to colorize is called *target image*. This color image can be chosen by the user or automatically provided from a database with an indexation algorithm.

The results available in this chapter are based on Pierre et al. (2014b,c, 2015a) which are among the most recent methods in patch-based colorization and on Persch et al. (2017) which is the current most competitive method for exemplar-based colorization of face images.

In order to transfer the colors from the source image to the target one, three concepts have been proposed in the literature. One of them is based on geometry, the two others are based on texture similarities. The first one is specifically well adapted to face colorization. In the first part of this section, we will review the work of Persch et al. (2017) which is the current most competitive method for exemplar-based colorization of face images. Next, we will present an overview of segmentation-based approaches which use the texture similarities on the segmented parts of the images to transfer colors. Finally, we present patch-based technique which avoids the requirement of an efficient segmentation method and which can be coupled with a variational model.

Morphing-Based Approach

In this section, we describe the model of Persch et al. (2017). The authors compute the morphing map between the two grayscale images I_{temp} and I_{tar} with a model inspired by Berkels et al. (2015). This results in the deformation sequence φ which produces the resulting map Φ from the template image to the target one. Due to the discretization of the images, the map Φ is defined, for images of size $n \times m$, on the discrete grid $\mathcal{G} := \{1 \dots n\} \times \{1 \dots m\}$:

$$\Phi : \mathcal{G} \rightarrow [1, n] \times [1, m], \quad x \mapsto \Phi(x), \quad (24)$$

where $\Phi(x)$ is the position in the source image which corresponds to the pixel $x \in \mathcal{G}$ in the target image. Now we colorize the target image by computing its chrominance channels, denoted by $(U_{\text{tar}}(x), V_{\text{tar}}(x))$ at position x as

$$(U_{\text{tar}}(x), V_{\text{tar}}(x)) := (U(\Phi(x)), V(\Phi(x))). \quad (25)$$

The chrominance channels of the target image are defined on the image grid \mathcal{G} , but usually $\Phi(x) \notin \mathcal{G}$. Therefore, the values of the chrominance channels at $\Phi(x)$ have to be computed by interpolation. In the algorithm, bilinear interpolation is simply used, which is defined for $\Phi(x) = (p, q)$ with $(p, q) \in [i, i + 1] \times [j, j + 1]$, $(i, j) \in \{1, \dots, m - 1\} \times \{1, \dots, n - 1\}$ by

$$\begin{aligned} U(\Phi(x)) &= U(p, q) \\ &:= (i + 1 - p, p - i) \begin{pmatrix} U(i, j) & U(i, j + 1) \\ U(i + 1, j) & U(i + 1, j + 1) \end{pmatrix} \begin{pmatrix} j + 1 - q \\ q - j \end{pmatrix}. \end{aligned} \quad (26)$$

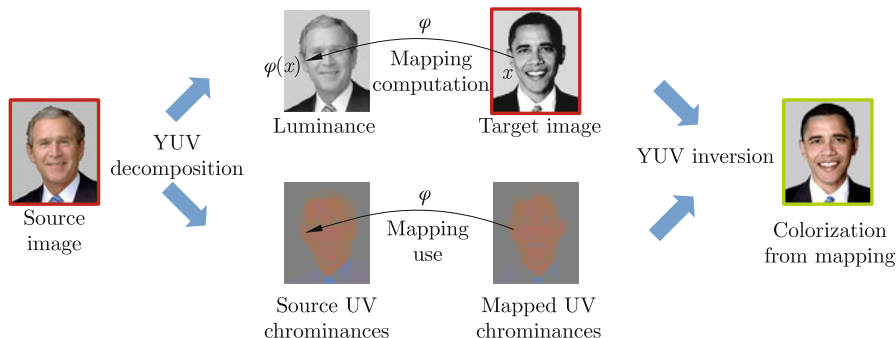


Fig. 6 Overview of the color transfer. The mapping φ is computed from a model inspired by Berkels et al. (2015) between the luminance channel of the source image and the target one. From this map, the chrominances of the source image are mapped. Finally, from these chrominances and the target image the colorization result is computed

Finally, a colorized RGB image is computed from its luminance $I_{\text{tar}} = Y_{\text{tar}}$ and the chrominance channels.

Figure 6 summarizes the color transfer method.

The technique proposed in Persch et al. (2017) is adapted to faces. To address the problem of colorization of textured images, geometric similarities are not reliable. Texture similarities have to be obviously compared. Such approaches are reviewed in the next sections.

Segmentation-Based Techniques

In order to transfer the colors from the source image to the target one, a lot of approaches are based on an image segmentation technique in order to compare the statistical attributes of the textures. For instance, the authors of Irony et al. (2005) proposed to compute the best correspondence between the target image and some segmented parts of the source image. From these correspondences, some *micro-scribbles* are drawn of the target image from the source image and the color strokes are then propagated by the diffusion technique in Levin et al. (2004). In Sýkora et al. (2004), the author used a segmentation approach to colorize images of old cartoons. The method of Gupta et al. (2012) extracts various descriptors from superpixel segmentation (see, e.g., Ren and Malik 2003; Achanta et al. 2012) from target image and matches them with the ones of the target image with these various descriptors (SURF, mean, standard deviation, Gabor filters, etc.). The method hence draws one scribble for each superpixel from this matching. The final color is computed from the optimization of a criterion which favors a spatial consistency of the colors as done in Levin et al. (2004). A similar approach has been proposed in Kuzovkin et al. (2015).

The efficiency of these methods depends on the preliminary segmentation of the images. In the next section, we propose an overview of patch-based techniques which avoid this preliminary step.

Patch-Based Methods

The first patch-based method for image colorization is the one proposed by Welsh et al. (2002), which is widely inspired by the texture synthesis algorithm introduced by the authors of Efros and Leung (1999). It is based on the patch similarities in the colorization process.

First, a luminance remapping (see, e.g., Hertzmann et al. 2001) is done as a first step: in order to make the luminance values more comparable between the source image and the target one, an affine mapping is used on the luminance of the source image in order to better match the histogram of the luminance channel. Indeed, the range of the luminance channels could be different and the comparison of these channels could be senseless.

Next, for each pixel of the target image, the algorithm compare the patch centered in this pixel with a set of patches extracted from the luminance channel of the source image. Once the closest patch is computed, the chrominance values of the pixel at the center of the patch of the source image are extracted and provided to the considered pixel in the target image (see, e.g., Fig. 7). In combination with the luminance of the target image and the chrominance values extracted from the source image, a RGB color is given.

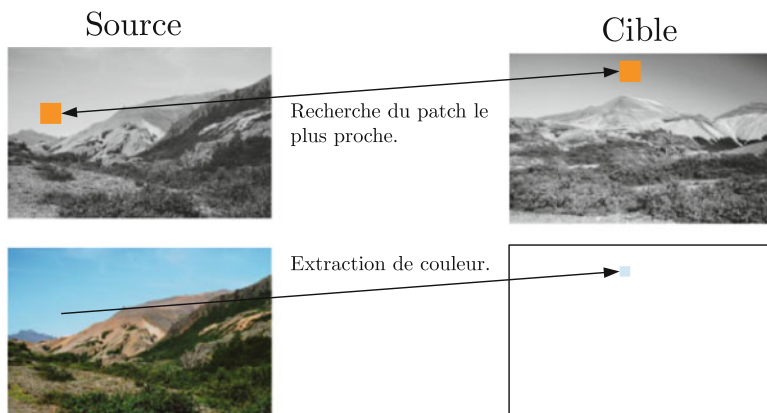
The set of reference patches extracted from the source image is a subset of patches randomly chosen in this way: the image is divided within a regular grid and one pixel is chosen randomly on each part of this grid (see, e.g., Fig. 7b).

The authors of Di Blasi and Reforgiato (2003) proposed an improvement which speeds up the patch research with a tree-clustering algorithm inspired from Wei and Levoy (2000). Next, the authors of Chen and Ye (2011) proposed an improvement based on a Bayesian approach.

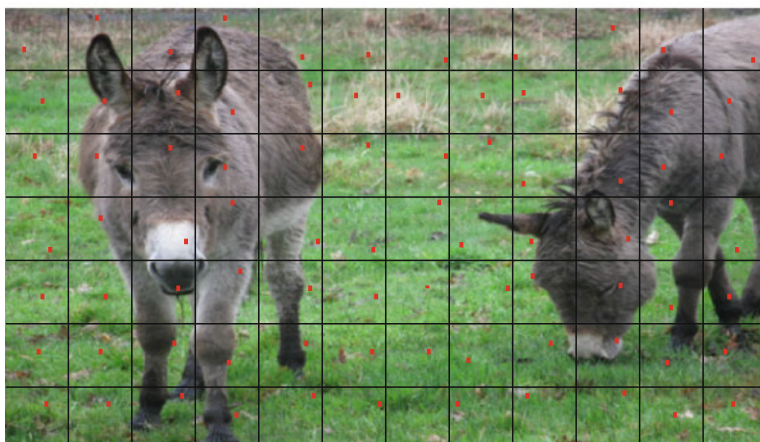
The patch-based approaches suffer from two drawbacks, which are the difficulty choosing a reliable metric to compare the patches and the spatial coherency in the border of two areas with different textures. We will see in the following how to overcome these limitations.

The patch-based approaches need some metrics in order to compare the patches. Unfortunately, there does not exist any perfect metric, each of them having its advantages and drawbacks. In most computer vision problems, the algorithms have to distinguish objects or textures with the same accuracy and the same sensitivity as human visual system. Metrics for texture comparison are based on numerical data. The link between this data and the human visual system is done by features that are vectors which describe the local statistic of the image.

The most simple metrics are based on the mean or the standard deviation of the patches, whereas some others use histograms, Fourier transform, SURF features (Bay et al. 2006), structure tensors, co-variance matrices, Gabor features, etc.



(a) Search of the candidates.



(b) Sub-sampling on a regular grid.

Fig. 7 For each pixel of the target image, the method compares the patch centered on the pixel with the ones of the gray-scale version of the source. Next, the method retains the color of the central pixel of the closest patch (see (a)). To speed up the algorithm, the search is not performed among all pixels, but only on a sub-sampling (see (b))

Based on various patch metrics, it is thus possible to get many exemplar-based colorization results. In the following, we focus on the fusion of such results to obtain only one final result.

Experimentally, the authors of Bugeau and Ta (2012) have used the following descriptors:

- The standard deviation on 5×5 and 3×3 patches
- The spectrum amplitude (FFT) on 7×7 , 9×9 , and 11×11 patches
- Difference in L^1 norm of the cumulative histograms on 7×7 , 9×9 , and 11×11 patches

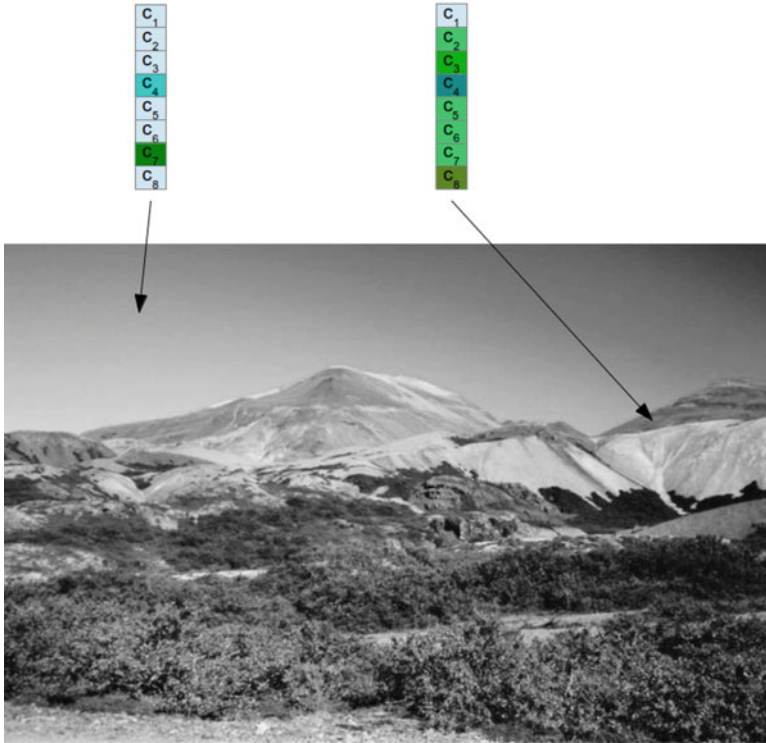


Fig. 8 Some methods of the literature begin with the search of C candidates per pixel (here $C = 8$)

These descriptors are used by the authors of Bugeau and Ta (2012) to extract eight color candidates for each pixel in the same way as done in Welsh et al. (2002). For each metric, the method retains the pixel of the source image corresponding to the closest patch with respect to this metric. After this step, for each pixel of the target image, eight pixels of the source image can match. To summarize, each pixel having its luminance and eight chrominance values coming from the matched pixels (see, e.g., Fig. 8), eight colors are available, called *color candidates*. In the work of Bugeau and Ta (2012), the colors are used directly, whereas in Pierre et al. (2014a) an oblique projection in the RGB color space is proposed in order to avoid some artificial modification of the hue due to gamut problems.

Some other metrics could be used. For instance, whereas the method of Charpiat et al. (2008) is not based on patch decomposition, it uses a local representation with SURF descriptors to predict color in each pixel. Let us mention that this method also requires numerous and complex steps.

With multiple color candidates coming from various descriptors, a choice has to be done among them. In the following we will consider a generic number of color candidates denoted by C . The aim of the methods described hereinafter consists of

the selection of one of the color candidates. Let us notice that the choice of an ideal metric based on metric learning has been proposed in Pierre et al. (2015b) but with rather worst results than the state of the art due to a lack of spatial regularization of the results. In order to retain only one color per pixel, the authors of Bugeau and Ta (2012) proposed to compute a median of the candidates based on an order between them computed with a standard PCA of the set of colors. This PCA is required because there is no natural order in the RGB space of colors. The method of L  zoray et al. (2005, 2007b) provides an order in the set of colors, but it requires some neighborhood information which is not available here.

Let us remark that the method of Bugeau and Ta (2012) does not use the spatial regularization or spatial coherency of the color to choose a color candidate. The authors of Jin et al. (2019) proposed an extension to exemplar-based colorization of Jung and Kang (2016) with color inference based on patch descriptors (DFT and variance of patches). A variational method similar to Pierre et al. (2015a) is proposed to regularize the final results.

A Variational Model for Image Colorization with Channel Coupling

In Pierre et al. (2015a), the authors have proposed a functional that selects a color among candidates extracted from a patch-based method, inspired by the method of Bugeau et al. (2014), in order to tackle some issues (numerical cost of numerical scheme, halo effects, etc.). Assume that C candidates are available in each pixel of a domain Ω and assume that two chrominance channels are available for each candidate. Let us denote for each pixel at position x the i -th candidate by $c_i(x)$, $u(x) = (U(x), V(x))$ stands for chrominances to compute, and $w(x) = \{w_i(x)\}$ with $i = 1, \dots, C$ for the candidate weights. Let us minimize the following functional with respect to (u, w) :

$$F(u, w) := TV_{\mathcal{C}}(u) + \frac{\lambda}{2} \int_{\Omega} \sum_{i=1}^C w_i \|u(x) - c_i(x)\|_2^2 dx + \chi_{\mathcal{R}}(u(x)) + \chi_{\Delta}(w(x)). \quad (27)$$

The central part of this model is based on the term

$$\int_{\Omega} \sum_{i=1}^C w_i(x) \|u(x) - c_i(x)\|_2^2 dx. \quad (28)$$

This term is a weighted average of some L^2 norms with respect to the candidates c_i . The weights w_i can be seen as a probability distribution of the c_i . For instance, if $w_1 = 1$ and $w_i = 0$ for $2 \leq i \leq C$, the minimum of F with respect to u is equal to the minimization of

$$TV_{\mathcal{C}}(u) + \frac{\lambda}{2} \int_{\Omega} \|u(x) - c_1(x)\|_2^2 dx + \chi_{\mathcal{R}}(u(x)). \quad (29)$$

To simplify the notations, the dependence of each value to the position x of the current pixel will be removed in the following. For instance, the second term of (27) will be denoted by $\int_{\Omega} \sum_{i=1}^C w_i \|u - c_i\|_2^2 dx$.

This model is a classical one with a fidelity-data term $\int_{\Omega} \sum_{i=1}^C w_i \|u - c_i\|_2^2$ and a regularization term $TV_{\mathcal{C}}(u)$ defined in Equation (6). Since the first step of the method extracts many candidates, we propose averaging the fidelity-data term issued from each candidate. This average is weighted by w_i . Thus, the term

$$\int_{\Omega} \sum_{i=1}^C w_i \|u - c_i\|_2^2 \quad (30)$$

connects the candidate color c_i to the color u that will be retained. The minimum of this term with respect to u is reached when u is equal to the weighted average of candidates c_i .

Since the average is weighted by w_i , these weights are constrained to be onto the probability simplex. This constraint is formalized by $\chi_{\Delta}(w)$ whose value is 0 if $w \in \Delta$ and $+\infty$ otherwise, with Δ defined as:

$$\Delta := \left\{ (w_1, \dots, w_C) \text{ s.t. } 0 \leq w_i \leq 1 \text{ and } \sum_{i=1}^C w_i = 1 \right\}. \quad (31)$$

In order to compute a suitable solution for the problem in (27), the authors of Pierre et al. (2015a) propose a primal-dual algorithm with alternating minimization of the terms depending of w . They also proposed numerical experiments showing the convergence of their algorithm. Let us note that this recent reference shows that the convergence of such numerical schemes can be demonstrated after smoothing of the total variation term. Among all the numerical schemes proposed in the references (Pierre et al. 2015a; Tan et al. 2019), we choose the methodology having the best convergence rate as well as a convergence proof. This scheme is given in Algorithm 2 in Tan et al. (2019). This algorithm is a block-coordinate forward-backward algorithm. To increase the speed-up of the convergence, Algorithm 2 of Tan et al. (2019) is initialized with the result of 500 iterations of the primal-dual algorithm of Pierre et al. (2015a). Whereas this algorithm has no guaranty of convergence, the authors of Tan et al. (2019) have experimentally observed that it numerically converges faster.

Unfortunately, the functional (27) is highly non-convex and it contains many critical points. More precisely, the functional is convex with respect to u with fixed w , and reversely, it is convex with respect to w for fixed u . Nevertheless, the functional is not convex with respect to the joint variables (u, w) . Thus, even if the

numerical scheme would converge to a local minimum, the solution of the problem depends on the initialization.

The dependence to the initialization implies an influence of the source image for exemplar-based colorization, and it does not enable a fully automatic image colorization within this paradigm. In the next section, we will show how the colorization from datasets can be used to tackle this last limitation.

Colorization from Dataset

The third colorization approach uses some large image databases (Zhang et al. 2016). Neural networks (convolutional neural networks, generative adversarial networks, autoencoder, recursive neural networks) have also been used successfully leading to a significant number of recent contributions. The survey proposed in this section is based on the paper (Mouzon et al. 2019). This literature can be divided into two categories of methods. The first evaluates the statistical distribution of colors for each pixel (Zhang et al. 2016; Royer et al. 2017; Chen et al. 2018). The network computes, for each pixel of the grayscale images, the probability distribution of the possible colors. The second takes a grayscale image as input and provides a color image as output, mostly in the form of chrominance channels (Iizuka et al. 2016; Larsson et al. 2016; Cao et al. 2017; Isola et al. 2017; Deshpande et al. 2017; Guadarrama et al. 2017; He et al. 2018; Su et al. 2018). Some methods use a mixture of both (e.g., Zhang et al. 2017).

Both techniques require image resizing that is either done by deconvolution layers or performed a posteriori with standard interpolation techniques.

In the case of Zhang et al. (2016), the network computes a probability distribution of the color on a down-sampled version of the original image. The choice of a color in each pixel at high resolution is made by linear interpolation without taking into account the grayscale image. Hence, the contours of chrominance and luminance may be not aligned, producing halo effects. Figure 9 shows some gray halo effects at the bottom of the cat that are visible on the red part, near the tail. On the other hand, in comparison to the other approaches of the state of the art, the method of Zhang et al. (2016) produces images which are shinier.



Fig. 9 Example of halo effects produced by the method of Zhang et al. (2016). Based on a variational model, the approach of Mouzon et al. (2019) is able to remove such artifacts

Below, the CNN described in Zhang et al. (2016) is presented in detail. The method of Zhang et al. (2016) is based on a discretization of the CIE Lab color space into $C = 313$ colors. This number of reference colors comes from the intersection gamut of the RGB color space and the discretization of the Lab space. The authors designed a CNN based on a VGG network (Simonyan and Zisserman 2015) in order to compute a statistical distribution of the C colors in each pixel. The input of the network is the L lightness channel of the Lab transform of an image of size 256×256 . The output is a distribution of probability over a set of 313 couples of a , b chrominance values for each pixel of a 64×64 size image. The quantification of the color space in 313 colors is computed from two assumptions. First, the colors are regularly spaced onto the CIE Lab color space. On this color space two colors are close with respect to the Euclidean norm when the human visual system feels them close. The second assumption that rules the set of colors is the respect of the RGB gamut. The colors have to be displayable onto a standard screen.

To train this CNN, the database ImageNet (Deng et al. 2009) is used without the grayscale images. The images are resized at size 256×256 and then transformed into the CIE Lab color space. The images are then resized at size 64×64 to compute the a and b channels. The loss function used is the cross-entropy between the luminance (a , b) of the training image and the distribution over the 313 original colors. Let us denote by Δ the probability simplex in $C = 313$ dimensions.

Denoting by $(\hat{w}_i(x))_{i=1..C} \in \Delta^N$ the probability distribution of dimension C in the N pixels of the 64×64 image (over a domain Ω), and denoting by $(w_i(x))$ the ground truth distribution computed with a soft-encoding scheme (see Zhang et al. 2016 for details), the loss function is given by:

$$L(\hat{w}, w) = - \sum_{x \in \Omega} \sum_{i=1}^C w_i(x) \log(\hat{w}_i(x)). \quad (32)$$

The forward propagation in the network provides a probability distribution over the C colors. In order to compute a colorization result, a choice among all these colors has to be performed. Basically, the authors of Zhang et al. (2016) proposed an annealed mean in each pixel, independently. After that, a resizing of the (a , b) channels at original size is done and recombined with brightness channel to obtain the color image.

Nevertheless, this recombination is done without taking into account any spatial consideration. In the next section, we will describe some approaches that couple some previously described algorithms.

Coupled Approaches

Neither the exemplar-based methods, nor the manual techniques, nor the deep learning approaches are able to colorize images without some defects. All of them having drawbacks or advantages, we propose to describe some coupling approaches

that rely on different types of methods in the literature. First, a framework to couple exemplar-based approach and manual colorization is described. A coupling of variational method with deep learning is then recalled.

Coupling Manual Approach with Exemplar-Based Colorization

A method can be considered interactive when the user can influence the result of the colorization process. Nevertheless, the interactivity can be difficult to reach. Indeed, if a method computes a result with a too long delay, the user cannot stand to an intermediary result in order to see the influence of his intervention. The results and the survey proposed in this section are based on the papers (Pierre et al. 2014b, 2015a) which have led to a software (Pierre et al. 2016).

Some of the exemplar-based methods enable some interaction with the user, for instance, the *swatches* approach of Welsh et al. (2002) in which the user distinguishes some parts of the image by drawing some rectangles on the source and target images where the textures are similar. The method then colorizes some parts of the target image with the specified parts of the source image. Finally, the method computes a solution for all the remaining uncolored pixels of the image based on the already colorized parts. The advantage of this framework is that the user can easily distinguish or associate the textures of the different images, which is difficult with an automatic method. At the opposite, the exemplar-based method is reliable to well colorize an image from its own parts, because the textures are more similar. With this method, a contextual information is added.

The framework of Chia et al. (2011) exploits the huge quantity of data available on the Internet. Nevertheless, the user has to manually segment and label the objects of the target image. Next, for each labeled object, the images with the same label are found on the Internet and used as source images. The image research is based on superpixel extraction (Comaniciu and Meer 2002) as well as graph-based optimization.

In the work of Ding et al. (2012), the scribbles are automatically generated and the user is invited to associate a color to each scribble. Then, the phases of the wavelet in the quaternion space are computed in order to propagate the colors along the lines of equal phase. Indeed, the wavelets in quaternion space are a measure of contours.

The method proposed in Pierre et al. (2014b) consists of a combination of the method of Bugeau and Ta (2012) and the one of Yatziv and Sapiro (2006). The approach uses a GPU implementation to compute a solution of model (27) that enables to colorize an image of size 370×600 in approximately 1 s. This computation time enables an extension of the exemplar-based approach of Pierre et al. (2014c) by including an interaction with the user, which leads to a software for colorization (Pierre et al. 2016).

The scribbles can be given in advance or added step by step by the user. When a source image is added, the first step consists of the extraction of C candidates as

in section “Patch-Based Methods” and the corresponding weights are initialed with the value $w = 1/C$.

The information given by the scribbles influences the weights and the candidate number. More precisely, for each pixel of the image, a new candidate is added for each scribble. When a candidate is introduced, its weight is initialized for the minimization process with a value depending on the geodesic distance in a similar way as Yatziv and Sapiro (2006).

The geodesic distance, denoted by D , is computed with the fast marching algorithm (Sethian 1999) with a potential equal to $\left(0.001 + \|\nabla u\|_2^2\right)^{-4}$ given by Chan and Vese (2001). D is normalized to get values between 0 and 1. The implementation of Peyré (2008) can be used to compute it.

The pixels having a low geodesic distance from a scribble get its color, whereas those having a high geodesic distance are not influenced by the user intervention. The w variable is composed of concatenation of uniform weights for the color candidates coming from the source image with the patch extraction and the weight coming from the geodesic distance. The values are then projected onto the probability simplex Δ with the algorithm of Chen and Ye (2011). The u variable is initialized with $\sum_i w_i c_i$ and the functional (27) is minimized using this initialization.

In Fig. 10, we show a first example of colorization using both manual and exemplar-based approaches. Figure 10a and b shows the source and target images. Figure 10c corresponds to exemplar-based colorization done without manual scribble. In this first result, the sky is not suitably colored since it appears brown instead of blue, as well as the door in ruins. Moreover, some blue blotches appear on the floor. Figure 10d shows the corrections done by the user by adding three scribbles on the exemplar-based result (Fig. 10c). Figure 10e illustrates the advantage of the combination of the methods. Indeed, the work provided by the user is of lower quality than the full manual colorization. It also shows that Model (27) is able to enhance contours.

Figure 11 shows the additional results and illustrate the advantage of using the joint model instead of using only the source image (fourth column) or only scribbles (fifth column). Colorization results in the last column in Fig. 11 are visually better than the ones computed from only one information source. This experiment shows

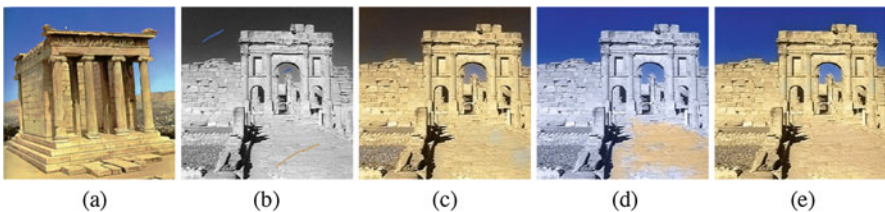


Fig. 10 Colorization using manual and exemplar-based approach. (a) Source image. (b) Target image with three scribbles. (c) Exemplar-based colorization. (d) Manual colorization. (e) Both

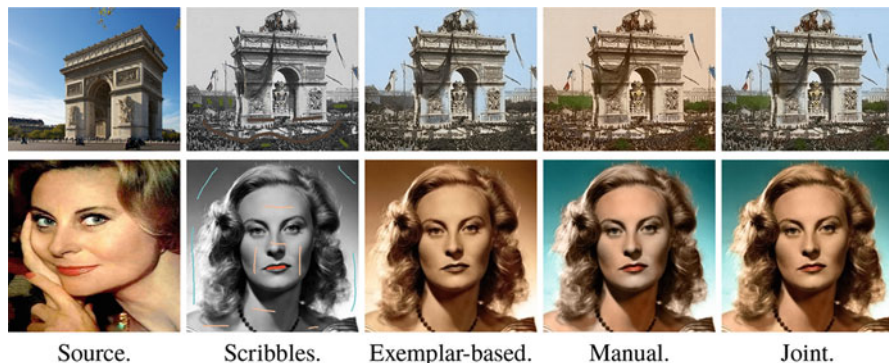


Fig. 11 Advantage of the joint approach, compared to manual and exemplar-based colorization. From left to right: source, target with scribbles added by the user, exemplar-based result, scribble-based results, and finally the joint approach

also that old photographs and faces are difficult to colorize with exemplar-based approaches since they require more scribbles. This statement has been done in Chen et al. (2004). Indeed, old pictures contain a lot of noise and textures. Face image contains smooth parts, for instance skin or background, with no textures. This kind of images is hard to colorize with assumption of texture similarities. Nevertheless, it is possible to compute a suitable result with the joint method, as well as morphing-based approach presented in section “[Morphing-Based Approach](#)”. Let us remark that the scribbles given by the user have naturally a local influence, but this influence can be also considered as global. For instance, on the last row in Fig. 11, the blue scribble in the arch also improves the color of the sky in the left-hand part of the image.

Coupling CNN with a Variational Approach

In the following, we recall the results given in the paper Mouzon et al. (2019) which consists of a coupling between a variational approach and the output of the CNN of Zhang et al. (2016). Next, we perform numerical comparisons with the original CNN approach of Zhang et al. (2016).

Coupling the CNN with a Variational Method

In image colorization, convolutional neural networks can be used to compute in each pixel a set of possible colors and their associated probabilities (Zhang et al. 2016). However, since the final choice is made without taking into account the regularity of the image, this leads to halo effects. To improve this, we first propose to adapt the functional of Pierre et al. (2015a) to the regularization of such results within the framework of colorization. The method of Pierre et al. (2015a) being able to choose between several color candidates in each pixel, it will be quite easy to use the color

distribution provided by the CNN described in Zhang et al. (2016). In addition, the numerical results of Pierre et al. (2015a) demonstrate the ability to remove halos, which is relevant to the limitations of Zhang et al. (2016). This functional will have to face two main problems: on the one hand, the transition from a low to a high resolution and, on the other hand, the maintenance of a higher saturation than the current methods.

In this section, a method to couple the prediction power of CNN with the precision of variational methods is described. To this aim, let us remark that the variable w of the functional (27) represents the ratio of each color candidate which is represented in the final result. This comes from the fact that, for a given vector $w \in \mathbb{R}^C$, the minimum of

$$\sum_{i=1}^C w_i \|u - c_i\| \quad (33)$$

with respect to u is given by

$$\sum_{i=1}^C w_i c_i. \quad (34)$$

Thus, it can be seen as a probability distribution of the colors in the desired color image, which has exactly the same purpose as the one of the CNN in Zhang et al. (2016).

Figure 12 shows an overview of Mouzon et al. (2019). First, the grayscale image, considered as the luminance L , is given as an input to the CNN. The output of the CNN is a probability distribution over 313 possible chromaticity at low resolution (64×64). In order to initialize the minimization algorithm, the output weights of the CNN can be used. The CNN provides a coarse-scale output that needs an up-sampling before producing a suitable output at original definition. Two ways can be considered. For the first one, the variational method can be used at coarse scale (low definition), and then an interpolation can be performed to recover a result at fine scale (high definition). For the second one, the probability distributions can be interpolated to get a high-definition array. In the following, the second approach will be preferred. Indeed, the interpolation of a color image produces a decrease of the saturation that makes the images drabber. By interpolating the probability distributions instead of the color images, the variational method will be able to compute a color for each pixel based on a coupling of the channels at high resolution. The given probability distribution is then used as the initialization value for the numerical scheme. As it was still proposed in Pierre et al. (2015a), the variable u is initialized with $\sum_{i=1}^C w_i c_i$. After the iterations of the functional, the result, denoted by (u^*, w^*) , provides some binary weights (see, e.g., Pierre et al. (2015a), Section 2.3.2) and a regularized result u^* that gives two chromaticity

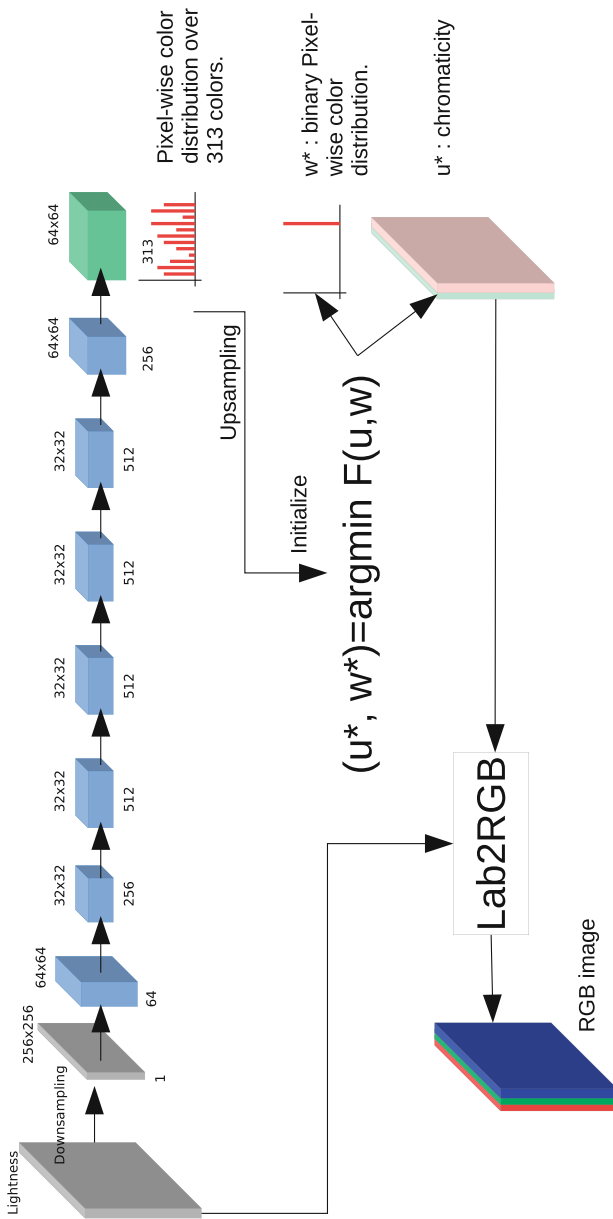


Fig. 12 Overview of method (Mouzon et al. 2019). A CNN computes color distribution on each pixel. A variational method selects then a color for each pixel based on a regularity hypothesis

channels, a and b , at initial definition. Recombined with the luminance L and transformed into the RGB space, that produces a color image.

Let us remark that the authors of Zhang et al. (2016) proposed to first produce the color image and then to resize it with bi-cubic interpolation. Unfortunately, up-sampling or down-sampling images with bi-linear or bi-cubic interpolations reduces the saturation of the colors and makes them drabber than the original. To avoid that, we propose here the opposite approach: we first up-sample the color distribution, and then we compute a color image at full definition by using it. Since the numerical scheme is used at full definition, the required memory of the algorithm for all the weights and the colors is a limitation to process high-resolution images on a standard PC. To tackle this issue, we propose to select some of the 313 colors. This selection is done with respect to the probability distribution of the colors, by choosing the ten highest modes.

This choice of ten modes has been done experimentally. For most images, eight or nine candidates are enough and taking more of them does not improve the result, but it increases the computational time. On the other hand, taking less candidates decreases the quality of the result on a significant number of images. Finally, the number of ten is a fair trade-off.

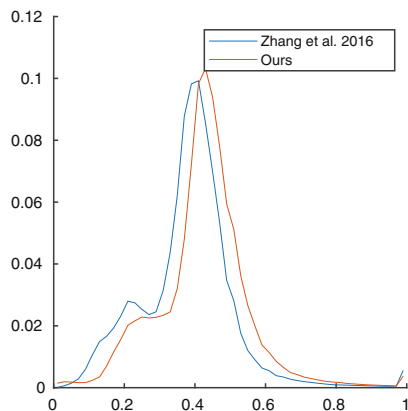
The training step of the CNN is done as in Zhang et al. (2016). The variational step is not taken into account during the training process. Indeed, the relation between the initialization of the weights and the result is not analytically described and the gradient back-propagation algorithms are not suitable for this problem. Thus, the training is done by feeding the CNN with a grayscale image as input and a color distribution as output. The variational step remains independent of the full framework during the training step. Its integration will be the purpose of future works.

In the next section, numerical results are presented.

Numerical Results

In this section we show a qualitative comparison between Zhang et al. (2016) and the framework of Mouzon et al. (2019). A lot of results provided by Zhang et al. (2016) are accurate and reliable. We show on these examples that the method of Mouzon et al. (2019) does not reduce the quality of the images. We then propose some comparisons with erroneous results of Zhang et al. (2016), which shows that the method of Mouzon et al. (2019) is reliable to fully automatically colorize images without artifacts and halo effects. A time comparison between the CNN inference computation and the variational step will be proposed to show that the regularization of the result is not a burden on the CNN approach. Finally, to show the limitation of CNN in image colorization, we show some results where neither the approach of Zhang et al. (2016) nor the framework of Mouzon et al. (2019) is able to produce some reliable results.

Figure 13 shows the colorization results of the method of Zhang et al. (2016). Whereas it is hard to see that the method of Mouzon et al. (2019) produces a



Histograms of saturation

Fig. 13 Results of Zhang et al. (2016) compared with Mouzon et al. (2019). The histogram of the saturation shows the second result is shinier than the first one. Indeed, the average value of the saturation is higher for the model of Mouzon et al. (2019) (0.4228) than the one of Zhang et al. (2016) (0.3802). (a) Original image. (b) Zhang et al. 2016. (c) Mouzon et al. 2019. (d) Histograms of saturation

shinier result than the result of Zhang et al. (2016) unless being a calibration expert, the histogram of the saturation is able to show the improvement. Indeed, since the histogram is right-shifted, it means that globally, the saturation is higher on the result of Mouzon et al. (2019). Quantitatively, the average of the saturation is equal to 0.4228 for the method of Mouzon et al. (2019), while it is equal to 0.3802 for the method of Zhang et al. (2016). This improvement comes from the fact that the method of Mouzon et al. (2019) selects one color among the ones given by the results of the CNN, whereas the method of Zhang et al. (2016) computes the annealed mean of them. The mean of the chrominances of the colors produces a decrease of the saturation and makes the colors drabber. By using a selection algorithm based on the image regularization, the method of Mouzon et al. (2019) is able to avoid this drawback.

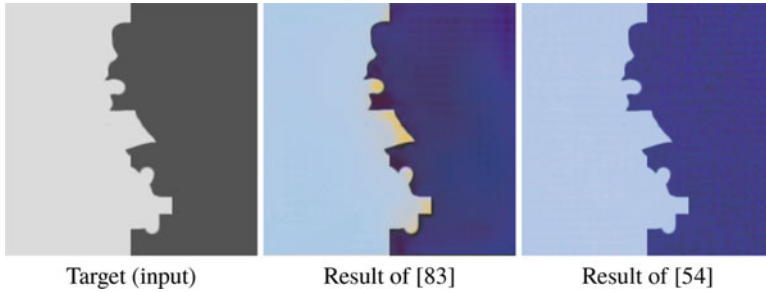


Fig. 14 Comparison of Mouzon et al. (2019) with Zhang et al. (2016). This example provides a proof of concept. The method of Mouzon et al. (2019) is able to remove the halo effects on the colorization result of Zhang et al. (2016)

The result in Fig. 14 is a proof of concept for the proposed framework. We can see a toy example which is automatically colorized by the method of Zhang et al. (2016). The result given by the method of Zhang et al. (2016) produces some halo effect near the only contour of the image, which is unnatural. The regularization of the result is able to remove this halo effect and to recover an image looking less artificial. This toy example contains only two constant parts. The aim of the variational method is to couple the contours of the chrominance channels and the ones of the luminance. The result produced with the method of Mouzon et al. (2019) contains no halo effect, showing the benefits of their framework.

In Fig. 15, we review some results and we compare them to the method of Zhang et al. (2016). For the lion, first line, a misalignment of the colors with the grayscale image is visible (a part of the lion is colorized in blue and a part of the sky is brown beige). This is a typical case of halo effect where the framework of Mouzon et al. (2019) is able to remove the artifacts. For the image of mountaineer, on the result of Zhang et al. (2016) some pink stains appear. With the method of Mouzon et al. (2019), the minimization of the total variation ensures the regularity of the image, and thus it removes these strains.

Figure 16 shows additional results. The first line is an old port card. Its colorization is reliable with the CNN, and, in addition, the variational approach makes it a little bit shinier. This example shows the ability of the approach of Mouzon et al. (2019) to colorize historical images. In the second example, most of the image is well colorized by the original method of Zhang et al. (2016). Nevertheless, the lighthouse and the right-side building contain some orange halos that are not reliable. With the variational method, the colors are convincing. Additional results are available in <http://www.fabienpierre.fr/colorization>.

The computational time of the CNN forward pass is about 1.5 s in GPU, whereas the minimization of the variational model (27) is about 15 s in Matlab on CPU.

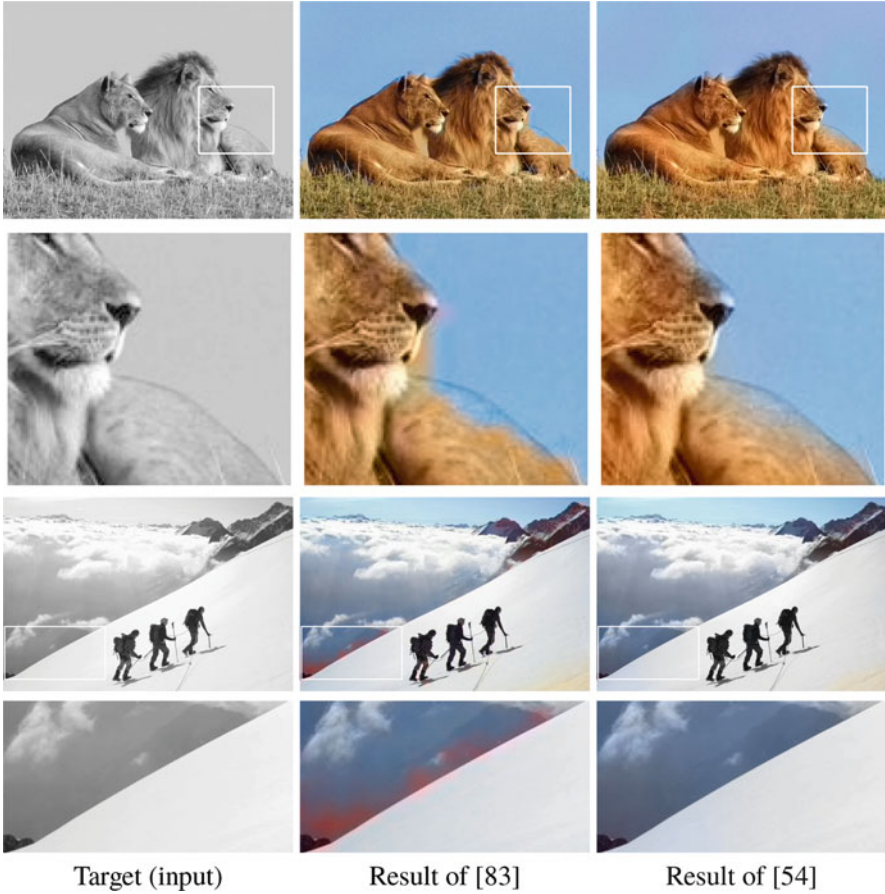


Fig. 15 Comparison between Mouzon et al. (2019) and Zhang et al. (2016)

In Pierre et al. (2017a), the authors provide a computation time almost equal to 1 s with unoptimized GPU implementation. Since the minimization scheme of Tan et al. (2019) is approximately the same, the computational time would be almost equal. Thus, the computational time of the approach of Mouzon et al. (2019) is not a burden in comparison with the method of Zhang et al. (2016).

In Fig. 17, a failure case is shown. In this case, since the minimization of the variational model strongly depends on its initialization, the method of Mouzon et al. (2019) is not able to recover its realistic colors. Actually, fully automatic colorization remains an open problem.

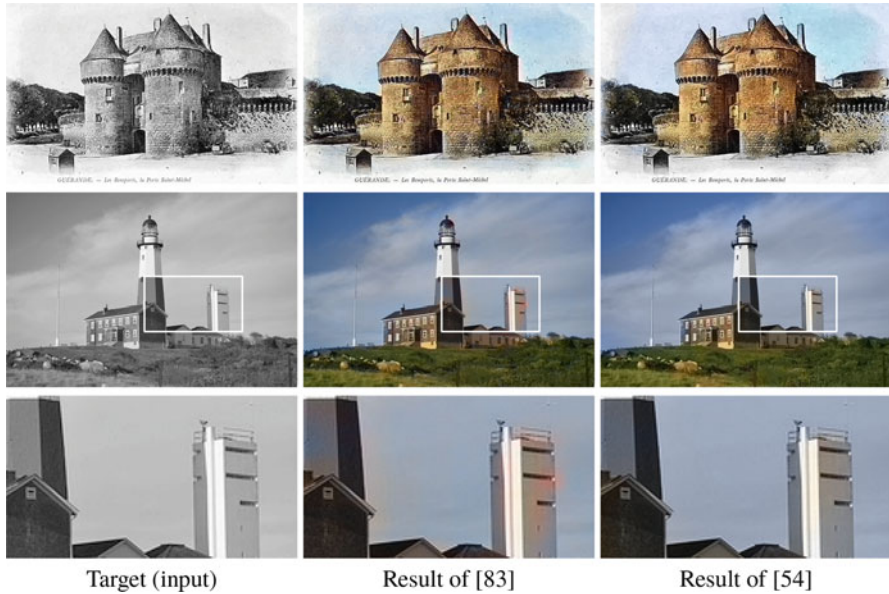


Fig. 16 Additional comparisons of Mouzon et al. (2019) with Zhang et al. (2016)

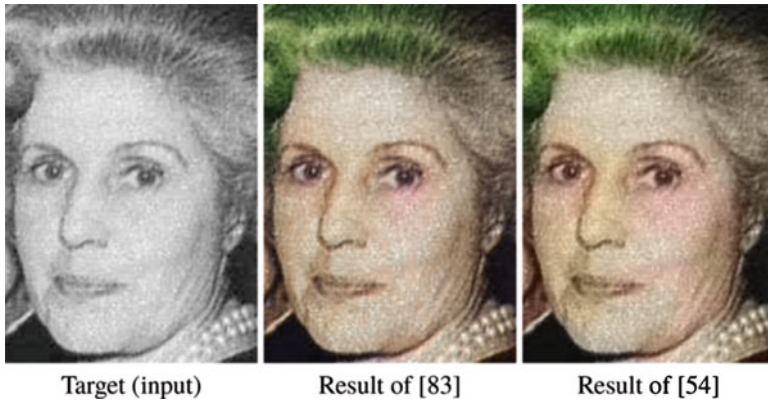


Fig. 17 Fail case. The prediction of the CNN is not able to recover a reliable color

Conclusion and Future Works

In this chapter, we have shown that image colorization has known a huge progress during the last 10 years by introducing a wide number of methods and approaches. Some extensions of these techniques have been proposed for video colorization but with limited number of frames. Future works could consider this application with more success. In this work, we have shown some limitations to colorization which

let the topic open for active research. Joint approaches have shown their efficiency, and a combination of deep learning with manual approaches could enhance the human system interface for image and video colorization.

Acknowledgments This study has been carried out with financial support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01).

References

- Abidi, B.R., Zheng, Y., Gribok, A.V., Abidi, M.A.: Improving weapon detection in single energy x-ray images through pseudocoloring. *IEEE Trans. Syst. Man Cybern. Part C* **36**(6), 784–796 (2006)
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012)
- Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: *European Conference on Computer Vision*, pp. 404–417. Springer (2006)
- Berkels, B., Effland, A., Rumpf, M.: Time discrete geodesic paths in the space of images. *SIAM J. Imaging Sci.* **8**(3), 1457–1488 (2015)
- Bugeau, A., Ta, V.T.: Patch-based image colorization. In: *IEEE International Conference on Pattern Recognition*, pp. 3058–3061 (2012)
- Bugeau, A., Ta, V.T., Papadakis, N.: Variational exemplar-based image colorization. *IEEE Trans. Image Proces.* **23**(1), 298–307 (2014)
- Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 151–166. Springer (2017)
- Caselles, V., Facciolo, G., Meinhardt, E.: Anisotropic cheeger sets and applications. *SIAM J. Imaging Sci.* **2**(4), 1211–1254 (2009)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Proces.* **10**(2), 266–277 (2001)
- Chan, T.F., Kang, S.H., Shen, J.: Total variation denoising and enhancement of color images based on the cb and hsv color models. *J. Vis. Commun. Image Represent.* **12**(4), 422–435 (2001)
- Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: *European Conference on Computer Vision*, pp. 126–139. Springer (2008)
- Chen, Y., Ye, X.: Projection onto a simplex. *arXiv preprint arXiv:1101.6081* (2011)
- Chen, T., Wang, Y., Schillings, V., Meinel, C.: Grayscale image matting and colorization. In: *Asian Conference on Computer Vision*, pp. 1164–1169 (2004)
- Chen, Y., Luo, Y., Ding, Y., Yu, B.: Automatic colorization of images from chinese black and white films based on cnn. In: *2018 IEEE International Conference on Audio, Language and Image Processing*, pp. 97–102 (2018)
- Chia, A.Y.S., Zhuo, S., Kumar, R.G., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S.: Semantic colorization with internet images. In: *ACM SIGGRAPH ASIA* (2011)
- Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002)
- Cui, M., Hu, J., Razdan, A., Wonka, P.: Color-to-gray conversion using isomap. *Vis. Comput.* **26**(11), 1349–1360 (2010)
- Deledalle, C.A., Papadakis, N., Salmon, J., Vaiter, S.: Clear: covariant least-square re-fitting with applications to image restoration. *SIAM J. Imaging Sci.* **10**(1), 243–284 (2017)

- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
- Deshpande, A., Lu, J., Yeh, M.C., Chong, M.J., Forsyth, D.A.: Learning diverse image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2877–2885 (2017)
- Di Blasi, G., Reforgiato, D.: Fast colorization of gray images. Eurographics Italian (2003). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.6839&rep=rep1&type=pdf>
- Ding, X., Xu, Y., Deng, L., Yang, X.: Colorization using quaternion algebra with automatic scribble generation. In: Advances in Multimedia Modeling (2012)
- Drew, M.S., Finlayson, G.D.: Improvement of colorization realism via the structure tensor. *Int. J. Image Graph.* **11**(04), 589–609 (2011)
- Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038 (1999)
- Fitschen, J.H., Nikolova, M., Pierre, F., Steidl, G.: A variational model for color assignment. In: Scale Space and Variational Methods in Computer Vision, pp. 437–448 (2015)
- Fornasier, M.: Nonlinear projection recovery in digital inpainting for color image restoration. *J. Math. Imaging Vis.* **24**(3), 359–373 (2006)
- Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Upper Saddle River, Pearson (2008)
- Guadarrama, S., Dahl, R., Bieber, D., Shlens, J., Norouzi, M., Murphy, K.: Pixcolor: pixel recursive colorization. In: British Machine Vision Conference (2017)
- Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: ACM International Conference on Multimedia, pp. 369–378 (2012)
- He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. *ACM Trans. Graph.* **37**(4), 47:1–47:16 (2018)
- Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: ACM Computer Graphics and Interactive Techniques, pp. 327–340 (2001)
- Heu, J.H., Hyun, D.Y., Kim, C.S., Lee, S.U.: Image and video colorization based on prioritized source propagation. In: IEEE International Conference on Image Processing, pp. 465–468 (2009)
- Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.* **35**(4), 1–11 (2016)
- Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Eurographics Symposium on Rendering, vol. 2. Citeseer (2005)
- Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
- Jin, Z., Zhou, C., Ng, M.K.: A coupled total variation model with curvature driven for image colorization. *Inverse Prob. Imaging* **10**(1930–8337), 1037 (2016). <https://doi.org/10.3934/ipi.2016031>
- Jin, Z., Min, L., Ng, M.K., Zheng, M.: Image colorization by fusion of color transfers based on DFT and variance features. *Comput. Math. Appl.* **77**, 2553–2567 (2019)
- Jung, M., Kang, M.: Variational image colorization models using higher-order Mumford–Shah regularizers. *J. Sci. Comput.* **68**(2), 864–888 (2016). <https://doi.org/10.1007/s10915-015-0162-9>
- Kang, S.H., March, R.: Variational models for image colorization via chromaticity and brightness decomposition. *IEEE Trans. Image Proces.* **16**(9), 2251–2261 (2007)
- Kawulok, M., Kawulok, J., Smolka, B.: Discriminative textural features for image and video colorization. *IEICE Trans. Inf. Syst.* **95-D**(7), 1722–1730 (2012)
- Kim, T.H., Lee, K.M., Lee, S.U.: Edge-preserving colorization using data-driven random walks with restart. In: IEEE International Conference on Image Processing, pp. 1661–1664 (2010)
- Kuhn, G.R., Oliveira, M.M., Fernandes, L.A.: An improved contrast enhancing approach for color-to-grayscale mappings. *Vis. Comput.* **24**(7–9), 505–514 (2008)

- Kuzovkin, D., Chamaret, C., Pouli, T.: Descriptor-based image colorization and regularization. In: Computational Color Imaging, pp. 59–68. Springer, Cham (2015)
- Lagodzinski, P., Smolka, B.: Digital image colorization based on probabilistic distance transformation. In: 50th International Symposium ELMAR, vol. 2, pp. 495–498 (2008)
- Lannaud, C.: Fallait-il coloriser la guerre? L'express (2009). Disponible en ligne sur http://www.lexpress.fr/culture/tele/fallait-il-coloriser-la-guerre_789380.html
- Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European Conference on Computer Vision, pp. 1–16. Springer (2016)
- Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM Transactions on Graphics, vol. 23–3, pp. 689–694 (2004)
- Lézoray, O., Meurie, C., Elmoataz, A.: A graph approach to color mathematical morphology. In: IEEE International Symposium on Signal Processing and Information Technology, pp. 856–861 (2005)
- Lézoray, O., Elmoataz, A., Bougleux, S.: Graph regularization for color image processing. *Comput. Vis. Image Underst.* **107**(1), 38–55 (2007a)
- Lézoray, O., Elmoataz, A., Meurie, C.: Mathematical morphology in any color space. In: IAPR/IEEE International Conference on Image Analysis and Processing, Computational Color Imaging Workshop (2007b)
- Lézoray, O., Ta, V.T., Elmoataz, A.: Nonlocal graph regularization for image colorization. In: IEEE International Conference on Pattern Recognition, pp. 1–4 (2008)
- Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.Q., Shum, H.Y.: Natural image colorization. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques, EGSR'07, pp. 309–320. Eurographics Association, Aire-la-Ville (2007). <https://doi.org/10.2312/EGWR/EGSR07/309-320>
- Mouzon, T., Pierre, F., Berger, M.O.: Joint CNN and variational model for fully-automatic image colorization. In: SSVM 2019 – Seventh International Conference on Scale Space and Variational Methods in Computer Vision, Hofgeismar (2019). <https://hal.archives-ouvertes.fr/hal-02059820>
- Nikolova, M., Steidl, G.: Fast hue and range preserving histogram specification: theory and new algorithms for color image enhancement. *IEEE Trans. Image Proces.* **23**(9), 4087–4100 (2014)
- Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990)
- Persch, J., Pierre, F., Steidl, G.: Exemplar-based face colorization using image morphing. *J. Imaging* **3**(4), 48 (2017)
- Peter, P., Kaufhold, L., Weickert, J.: Turning diffusion-based image colorization into efficient color compression. *IEEE Trans. Image Proces.* **26**(2), 860–869 (2017)
- Peyré, G.: Toolbox fast marching – a toolbox for fast marching and level sets computations (2008). <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6110&objectType=FILE>
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Hue constrained image colorization in the RGB space. Preprint (2014a). Disponible en ligne sur <https://hal.archives-ouvertes.fr/hal-00995724/document>
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: A unified model for image colorization. In: Color and Photometry in Computer Vision (ECCV Workshop), pp. 1–12 (2014b)
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T., Papadakis, N.: Exemplar-based colorization in RGB color space. In: IEEE International Conference on Image Processing, pp. 1–5 (2014c)
- Pierre, F., Aujol, J.F., Bugeau, A., Papadakis, N., Ta, V.T.: Luminance-chrominance model for image colorization. *SIAM J. Imaging Sci.* **8**(1), 536–563 (2015a)
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Combinaison linéaire optimale de métriques pour la colorisation d'images. In: XXVème colloque GRETSI, pp. 1–4 (2015b)
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Luminance-hue specification in the RGB space. In: Scale Space and Variational Methods in Computer Vision, pp. 413–424 (2015c)

- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Colociel. Dépôt Agence de Protection des Programmes No IDD.N.FR.001.080021.000.S.P.2016.000.2100 (2016). Disponible en ligne sur http://www.labri.fr/perso/fpierre/colociel_v1.zip
- Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Interactive video colorization within a variational framework. *SIAM J. Imaging Sci.* **10**(4), 2293–2325 (2017a) a
- Pierre, F., Aujol, J.F., Deledalle, C.A., Papadakis, N.: Luminance-guided chrominance denoising with debiased coupled total variation. In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 235–248. Springer (2017b)
- Quang, M.H., Kang, S.H., Le, T.M.: Image and video colorization using vector-valued reproducing kernel hilbert spaces. *J. Math. Imaging Vis.* **37**(1), 49–65 (2010)
- Ren, X., Malik, J.: Learning a classification model for segmentation. In: *IEEE International Conference on Computer Vision*, pp. 10–17 (2003)
- Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization. In: *British Machine Vision Conference* (2017)
- Sethian, J.A.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3. Cambridge University Press, Cambridge (1999)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations* (2015)
- Song, M., Tao, D., Chen, C., Bu, J., Yang, Y.: Color-to-gray based on chance of happening preservation. *Neurocomputing* **119**, 222–231 (2013)
- Su, Z., Liang, X., Guo, J., Gao, C., Luo, X.: An edge-refined vectorized deep colorization model for grayscale-to-color images. *Neurocomputing* **311**, 305–315 (2018)
- Sykora, D., Buriánek, J., Žára, J.: Unsupervised colorization of black-and-white cartoons. In: *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering*, pp. 121–127. ACM (2004)
- Tan, P., Pierre, F., Nikolova, M.: Inertial alternating generalized forward–backward splitting for image colorization. *J. Math. Imaging Vis.* **61**(5), 672–690 (2019)
- Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: *ACM Computer Graphics and Interactive Techniques*, pp. 479–488. Press/Addison-Wesley Publishing Co. (2000)
- Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: *ACM Transactions on Graphics*, vol. 21–3, pp. 277–280. ACM (2002)
- Williams, A., Barrus, S., Morley, R.K., Shirley, P.: An efficient and robust ray-box intersection algorithm. In: *ACM SIGGRAPH 2005 Courses*, p. 9 (2005)
- Wolfgang Baatz Massimo Fornasier, P.A.M., Schönlieb, C.B.: Inpainting of ancient austrian frescoes. In: Sarhangi, R., Séquin, C.H. (eds.) *Bridges Leeuwarden: Mathematics, Music, Art, Architecture, Culture*, pp. 163–170. Tarquin Publications, London (2008). Disponible en ligne sur <http://archive.bridgesmathart.org/2008/bridges2008-163.html>
- Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. *IEEE Trans. Image Proces.* **15**(5), 1120–1129 (2006)
- Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: *European Conference on Computer Vision*, pp. 1–16. Springer (2016)
- Zhang, R., Zhu, J.Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.* **9**(4), 119:1–119:11 (2017)
- Zheng, Y., Essock, E.A.: A local-coloring method for night-vision colorization utilizing image analysis and fusion. *Inf. Fusion* **9**(2), 186–199 (2008)



Numerical Solution for Sparse PDE Constrained Optimization

16

Xiaoliang Song and Bo Yu

Contents

Introduction	624
Finite Element Approximation and Error Estimates	632
An Inexact Heterogeneous ADMM Algorithm	642
An Inexact Heterogeneous ADMM Algorithm	642
Convergence Results of ihADMM	645
An Inexact Majorized Accelerated Block Coordinate Descent Method for (D_h)	652
An Inexact Block Symmetric Gauss-Seidel Iteration	653
Inexact Majorized Accelerate Block Coordinate Descent (imABCD) Method	656
A sGS-imABCD Algorithm for (D_h)	659
Numerical Results	663
Algorithmic Details	663
Examples	664
Conclusion	673
References	673

Abstract

In this chapter, elliptic PDE-constrained optimal control problems with L^1 -control cost (L^1 -EOCP) are considered. Motivated by the success of the first-order methods, we give an overview on two efficient first-order methods to solve L^1 -EOCP: inexact heterogeneous alternating direction method of multipliers (ihADMM) and an inexact symmetric Gauss-Seidel (sGS)-based 2-block majorized accelerated block coordinate descent (ABCD) method (sGS-imABCD). Different from the classical ADMM, the ihADMM adopts two

X. L. Song · B. Yu (✉)

School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, China

e-mail: songxiaoliang@dlut.edu.cn; yubo@dlut.edu.cn

different weighted inner products to define the augmented Lagrangian function in two subproblems, respectively. Benefiting from such different weighted techniques, two subproblems of ihADMM can be efficiently implemented. Furthermore, theoretical results on the global convergence as well as the iteration complexity results $o(1/k)$ for ihADMM are given. A common approach to solve the L^1 -EOCP is directly solving the primal problem. Based on the dual problem of L^1 -EOCP, which can be reformulated as a multi-block unconstrained convex composite minimization problem, an efficient inexact ABCD method is introduced for solving L^1 -EOCP. The design of this method combines an inexact 2-block majorized ABCD and the recent advances in the inexact sGS technique for solving a multi-block convex composite quadratic programming whose objective contains a nonsmooth term involving only the first block.

Keywords

PDE-constrained optimization · Sparsity · Finite element · ADMM · Symmetric Gauss-Seidel accelerated block coordinate descent

Introduction

We study the following linear-quadratic elliptic PDE-constrained optimal control problem with L^1 -control cost and piecewise box constraints on the control:

$$\left\{ \begin{array}{l} \min_{(y,u) \in Y \times U} J(y,u) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2 + \beta \|u\|_{L^1(\Omega)} \\ \text{s.t.} \quad Ly = u + y_r \text{ in } \Omega, \\ \quad \quad y = 0 \quad \quad \text{on } \Gamma, \\ \quad \quad u \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e. on } \Omega\} \subseteq U, \end{array} \right. \tag{P}$$

where $Y := H_0^1(\Omega)$, $U := L^2(\Omega)$, $\Omega \subseteq \mathbb{R}^n$ ($n = 2$ or 3) is a convex, open, and bounded domain with $C^{1,1}$ - or polygonal boundary Γ ; the desired state $y_d \in L^2(\Omega)$ and the source term $y_r \in L^2(\Omega)$ are given; and $a \leq 0 \leq b$ and $\alpha, \beta > 0$. Moreover, the operator L is a second-order linear elliptic differential operator. It is well-known that L^1 -norm could lead to sparse optimal control, i.e., the optimal control with small support. Such an optimal control problem (P) plays an important role for the placement of control devices (Stadler 2009). In some cases, it is difficult or undesirable to place control devices all over the control domain and one hopes to localize controllers in small and effective regions, and the L^1 -solution gives information about the optimal location of the control devices.

Through this chapter, let us suppose the elliptic PDEs involved in (P) which are of the form

$$\begin{aligned} Ly &= u + y_r && \text{in } \Omega, \\ y &= 0 && \text{on } \partial\Omega, \end{aligned} \tag{1}$$

satisfy the following assumption:

Assumption 1. *The linear second-order differential operator L is defined by*

$$(Ly)(x) := - \sum_{i,j=1}^n \partial_{x_j}(a_{ij}(x)y_{x_i}) + c_0(x)y(x), \tag{2}$$

where functions $a_{ij}(x), c_0(x) \in L^\infty(\Omega)$, $c_0 \geq 0$, and it is uniformly elliptic, i.e., $a_{ij}(x) = a_{ji}(x)$ and there is a constant $\theta > 0$ such that

$$\sum_{i,j=1}^n a_{ij}(x)\xi_i\xi_j \geq \theta\|\xi\|^2 \quad \text{for a.a. } x \in \Omega \text{ and } \forall \xi \in \mathbb{R}^n. \tag{3}$$

The weak formulation of (1) is given by

$$\text{Find } y \in H_0^1(\Omega) : a(y, v) = (u + y_r, v)_{L^2(\Omega)} \quad \forall v \in H_0^1(\Omega), \tag{4}$$

with the bilinear form

$$a(y, v) = \int_{\Omega} \left(\sum_{i,j=1}^n a_{ji}y_{x_i}v_{x_j} + c_0yv \right) dx, \tag{5}$$

or in short $Ay = B(u + y_r)$, where $A \in \mathcal{L}(Y, Y^*)$ is the operator induced by the bilinear form a , i.e., $Ay = a(y, \cdot)$ and $B \in \mathcal{L}(U, Y^*)$ is defined by $Bu = (u, \cdot)_{L^2(\Omega)}$. Since the bilinear form $a(\cdot, \cdot)$ is symmetric and U, Y are Hilbert spaces, we have $A^* \in \mathcal{L}(Y, Y^*) = A$ and $B^* \in \mathcal{L}(Y, U)$ with $B^*v = v$ for any $v \in Y$.

Remark 1. Although we assume that the Dirichlet boundary condition $y = 0$ holds, it should be noted that the assumption is not a restriction and our considerations can also carry over to the more general boundary conditions of Robin type:

$$\frac{\partial y}{\partial \nu} + \gamma y = g \quad \text{on } \partial\Omega,$$

where $g \in L^2(\partial\Omega)$ is given and $\gamma \in L^\infty(\partial\Omega)$ is nonnegative coefficient. Furthermore, it is assumed that the control satisfies $a \leq u \leq b$, where a and b have opposite signs. First, we should emphasize that this condition is required in practice, e.g., the placement of control devices. In addition, please also note that this condition is not a restriction from the point of view of the algorithm. If one has, e.g., $a > 0$ on Ω , the L^1 -norm in U_{ad} is in fact a linear function, and thus the problem can also be handled by our method.

Optimal control problems with $\alpha > 0$, $\beta = 0$ and their numerical realization have been studied intensively in recent papers; see, e.g., Hinze (2005), Falk (1973), Geveci (1979), Rösch (2006), Casas and Tröltzsch (2003), Meyer and Rösch (2004) and the references cited there. Let us first comment on known results on error estimates of control-constrained optimal control problems. Basic a priori error estimates were derived by Falk (1973) and Geveci (1979) where Falk considered distributed controls, while Geveci concentrated on the Neumann boundary controls. Both the authors proved optimal L^2 -error estimates $O(h)$ for piecewise constant approximations of the control variables. Convergence results for the approximations of the controls by piecewise linear, globally continuous elements can be found in Casas and Tröltzsch (2003), where Casas and Tröltzsch proved order $O(h)$ in the case of linear-quadratic control problems. Later Casas (2007) proved order $o(h)$ for the control problems governed by semilinear elliptic equations and quite general cost functions. In Rösch (2006) for the first time proved that the error order is $O(h^{\frac{3}{2}})$ under some special assumptions on the continuous solutions. However, his proof was just done in one dimension. All previous papers were devoted to the full discretization. Recently, a variational discretization concept is introduced by Hinze (2005). More precisely, the state variable and the state equation are discretized, but there is no discretization of the control. He showed that the control error is of order $O(h^2)$. In certain situations, the same convergence order can also be achieved by a special postprocessing procedure; see Meyer and Rösch (2004).

For the study of optimal control problems with sparsity promoting terms, as far as we know, the first paper devoted to this study is published by Stadler (2009), in which structural properties of the control variables were analyzed in the case of the linear-quadratic elliptic optimal control problem. In 2011, a priori and a posteriori error estimates were first given by Wachsmuth and Wachsmuth in (2011) for piecewise linear control discretizations, in which the convergence rate is obtained to be of order $O(h)$ under the L^2 norm. However, from the point of view of the algorithm, the resulting discretized L^1 -norm

$$\|u_h\|_{L^1(\Omega_h)} := \int_{\Omega_h} \left| \sum_{i=1}^{N_h} u_i \phi_i(x) \right| dx, \quad (6)$$

does not have a decoupled form with respect to the coefficients $\{u_i\}$, where $\{\phi_i(x)\}$ are the piecewise linear nodal basis functions. Hence, the authors introduced an alternative discretization of the L^1 -norm which relies on a nodal quadrature formula:

$$\|u_h\|_{L_h^1(\Omega)} := \sum_{i=1}^{N_h} |u_i| \int_{\Omega_h} \phi_i(x) dx. \quad (7)$$

Obviously, this quadrature incurs an additional error, although the authors proved that this approximation does not change the order of error estimates. In a sequence of papers (Casas et al. 2012a,b), for the non-convex case governed by a semi-linear elliptic equation, Casas et al. proved second-order necessary and sufficient

optimality conditions. Using the second-order sufficient optimality conditions, the authors provide error estimates of order h w.r.t. the L^∞ norm for three different choices of the control discretization. It should be pointed out that, for the piecewise linear control discretization case, a similar approximation technique to the one introduced in Wachsmuth and Wachsmuth (2011) is also used for the discretizations of the L^2 norm and L^1 norm of the control.

Apart from using L^1 -norm to induce sparsity, Clason and Kunisch in (2011) investigated elliptic control problems with measure-valued controls to promote the sparsity of the control. They discussed the existence and uniqueness of the corresponding dual problems. Subsequently, in 2012, Casas et al. in (2012) studied the optimality conditions and provided a priori finite element error estimates for the case of linear-quadratic elliptic control problems with a measure-valued control, in which the control measure was approximated by a linear combination of Dirac measures.

To numerically solve the problem (P), there are two possible ways. One is called *First discretize, then optimize*, and another approach is called *First optimize, then discretize* (Collis and Heinkenschloss 2002). Independently of where discretization is located, the resulting finite dimensional equations are quite large. Thus, both of these cases require us to consider proposing an efficient algorithm. In this chapter, we focus on the *First discretize, then optimize* approach to solve (P) and employ the piecewise linear finite elements to discretize (P).

Next, let us mention some existing numerical methods for solving problem (P). Since problem (P) is nonsmooth, thus applying semismooth Newton (SSN) methods is used to be a priority in consideration of their locally superlinear convergence. A special semismooth Newton method with the active set strategy, called the primal-dual active set (PDAS) method, is introduced in Bergounioux et al. (1999) for control-constrained elliptic optimal control problems. It is proved to have the locally superlinear convergence (see Ulbrich (2002), Ulbrich (2003), Hinze et al. (2009) for more details). Mesh-independence results for the SSN method were established in Hintermüller and Ulbrich (2004). Additionally, the authors in Porcelli et al. (2017) showed that a saddle point system with 2×2 block structure should be solved by employing some Krylov subspace methods with a good preconditioner at each iteration step of the SSN method. However, the 2×2 block linear system is obtained by reducing a 3×3 block linear system with bringing additional computation for linear system involving the mass matrix. Furthermore, the coefficient matrix of the Newton equation would change with every iteration due to the change of the active set. In this case, it is clear that forming a uniform preconditioner, which is used to precondition the Krylov subspace methods for solving the Newton equations, is difficult. For a survey of how to precondition saddle point problems, we refer to Herzog and Ekkehard (2010).

Undeniably, employing the SSN method can derive the solution with high precision. However, it should be mentioned that in general solving Newton equations is expensive.

Recently, for the finite dimensional large-scale optimization problem, some efficient first-order algorithms, such as iterative shrinkage/soft thresholding algorithms (ISTA) (Blumensath and Davies 2008), accelerated proximal gradient (APG)-based

method (Beck and Teboulle 2009), ADMM (Fazel et al. 2013; Chen and Toh 2017; Li et al. 2015, 2016), etc., have become the state-of-the-art algorithms. Thanks to the iteration complexity $O(1/k^2)$, a fast inexact proximal (FIP) method in function space, which is actually the APG method, was proposed to solve the problem (P) in Schindele and Borzì (2016). As we know, the efficiency of the FIP method depends on how close the step-length is to the Lipschitz constant. However, in general, choosing an appropriate step-length is difficult since the Lipschitz constant is usually not available analytically. Thus, this disadvantage largely limits the efficiency of APG method.

In this chapter, we will focus first on the ADMM algorithm. The classical ADMM was originally proposed by Glowinski and Marroco (1975) and Gabay and Mercier (1976), and it has found lots of efficient applications in a broad spectrum of areas. In particular, we refer to Boyd et al. (2011) for a review of the applications of ADMM in the areas of distributed optimization and statistical learning. We give a brief sketch of ADMM for the following finite dimensional linearly constrained convex programming problem with two blocks of functions and variables:

$$\begin{cases} \min f(u) + g(z) \\ \text{s.t. } A_1 u + A_2 z = c, \\ u \in U, z \in Z, \end{cases} \quad (8)$$

where $f(u) : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g(z) : \mathbb{R}^m \rightarrow \mathbb{R}$ are both closed, proper, and convex functions (but not necessary smooth); $A_1 \in \mathbb{R}^{p \times n}$, $A_2 \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$; $U \subset \mathbb{R}^n$ and $Z \subset \mathbb{R}^m$ are given closed, convex, and non-empty sets. Let

$$\mathcal{L}_\sigma(u, z, \lambda; \sigma) = f(u) + g(z) + \langle \lambda, A_1 u + A_2 z - c \rangle + \frac{\sigma}{2} \|A_1 u + A_2 z - c\|^2 \quad (9)$$

be the augmented Lagrangian function of (8) with the Lagrange multiplier $\lambda \in \mathbb{R}^p$ and the penalty parameter $\sigma > 0$. For a given $\tau \in \left(0, \frac{\sqrt{5}+1}{2}\right)$, the classical ADMM is described as follows:

$$\begin{cases} u^{k+1} = \arg \min_{u \in U} \mathcal{L}_\sigma(u, z^k, \lambda^k; \sigma), \\ z^{k+1} = \arg \min_{z \in Z} \mathcal{L}_\sigma(u^{k+1}, z, \lambda^k; \sigma), \\ \lambda^{k+1} = \lambda^k + \tau \rho (A_1 u^{k+1} + A_2 z^{k+1} - c). \end{cases} \quad (10)$$

Thanks to the separable structure of the objective function, each subproblem in (10) involves only one block of $f(u)$ and $g(z)$ and could be solved easily. Under some trivial assumptions, the classical ADMM for solving (8) has global convergence and sublinear convergence rate at least.

Motivated by the success of the finite dimensional ADMM algorithm, it is reasonable to consider extending the ADMM to infinite dimensional optimal control

problems, as well as the corresponding discretized problems. In 2016, the authors Elvetun and Nielsen (2014) adapted the split Bregman method (equivalent to the classical ADMM) to handle PDE-constrained optimization problems with total variation regularization. However, for the discretized problem, the authors did not take advantage of the inherent structure of problem and still used the classical ADMM to solve it. In this chapter, making full use of inherent structure of problem, we aim to design an appropriate ADMM-type algorithm to solve problem (P). In order to employ the ADMM algorithm and obtain a separable by adding an artificial variable z , we can separate the smooth and nonsmooth terms and equivalently reformulate problem (P) as

$$\left\{ \begin{array}{ll} \min_{(y,u,z) \in Y \times U \times U} & J(y, u, z) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{4} \|u\|_{L^2(\Omega)}^2 + \frac{\alpha}{4} \|z\|_{L^2(\Omega)}^2 \\ & + \beta \|z\|_{L^1(\Omega)} \\ \text{s.t.} & Ay = u + y_r \text{ in } \Omega, \\ & y = 0 \text{ on } \partial\Omega, \\ & u = z, \\ & z \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e on } \Omega\} \subseteq U. \end{array} \right. \quad (\tilde{P})$$

However, when the classical ADMM is directly used to solve (\overline{DP}_h) , i.e., the discrete version of (\tilde{P}) , there is no well-structure as in continuous case and the corresponding subproblems cannot be efficiently solved. Thus, making use of the inherent structure of (\overline{DP}_h) , a heterogeneous ADMM is proposed. Meanwhile, sometimes it is unnecessary to exactly compute the solution of each subproblem even if it is doable, especially at the early stage of the whole process. For example, if a subproblem is equivalent to solving a large-scale or ill-condition linear system, it is a natural idea to use the iterative methods such as some Krylov-based methods. Hence, taking the inexactness of the solutions of associated subproblems into account, a more practical inexact heterogeneous ADMM (ihADMM) is proposed. Different from the classical ADMM, we utilize two different weighted inner products to define the augmented Lagrangian function for two subproblems, respectively. Specifically, based on the M_h -weighted inner product, the augmented Lagrangian function with respect to the u -subproblem in k -th iteration is defined as

$$\mathcal{L}_\sigma(u, z^k; \lambda^k) = f(u) + g(z^k) + \langle \lambda, M_h(u - z^k) \rangle + \frac{\sigma}{2} \|u - z^k\|_{M_h}^2,$$

where M_h is the mass matrix. On the other hand, for the z -subproblem, based on the W_h -weighted inner product, the augmented Lagrangian function in k -th iteration is defined as

$$\mathcal{L}_\sigma(u^{k+1}, z; \lambda^k) = f(u^{k+1}) + g(z) + \langle \lambda, M_h(u^{k+1} - z) \rangle + \frac{\sigma}{2} \|u^{k+1} - z\|_{W_h}^2,$$

where the lumped mass matrix W_h is diagonal.

Benefiting from different weighted techniques, each subproblem of ihADMM for (\overline{DP}_h) can be efficiently solved. Specifically, the u -subproblem of ihADMM, which results in a large-scale linear system, is the main computation cost in whole algorithm. W_h -weighted technique makes z -subproblem have a decoupled form and admit a closed-form solution given by the soft thresholding operator and the projection operator onto the box constraint $[a, b]$. Moreover, global convergence and the iteration complexity result $o(1/k)$ in non-ergodic sense for our ihADMM will be proved. Taking the precision of discretized error into account, we should mention that using our ihADMM algorithm to solve problem (\overline{DP}_h) is highly enough and efficient in obtaining an approximate solution with moderate accuracy.

As far as we know, most of the aforementioned papers are devoted to solving the primal problem. Based on the special structure of the dual problem, we will also consider using the duality-based approach for (P) . The dual of problem (P) can be written, in its equivalent minimization form, as

$$\begin{aligned} \min \Phi(\lambda, \mu, p) := & \frac{1}{2} \|A^*p - y_d\|_{L^2(\Omega)}^2 + \frac{1}{2\alpha} \| -p + \lambda + \mu \|_{L^2(\Omega)}^2 \\ & + \langle p, y_r \rangle_{L^2(\Omega)} + \delta_{\beta B_\infty(0)}(\lambda) + \delta_{U_{ad}}^*(\mu) - \frac{1}{2} \|y_d\|_{L^2(\Omega)}^2, \end{aligned} \tag{D}$$

where $p \in H_0^1(\Omega)$, $\lambda, \mu \in L^2(\Omega)$, $B_\infty(0) := \{\lambda \in L^2(\Omega) : \|\lambda\|_{L^\infty(\Omega)} \leq 1\}$, and for any given non-empty, closed convex subset C of $L^2(\Omega)$, $\delta_C(\cdot)$ is the indicator function of C . Based on the L^2 -inner product, we define the conjugate of $\delta_C(\cdot)$ as follows:

$$\delta_C^*(w^*) = \sup_{w \in C} \langle w^*, w \rangle_{L^2(\Omega)}.$$

Although the duality-based approach has been introduced in Clason and Kunisch (2011) for elliptic control problems without control constraints in nonreflexive Banach spaces, the authors did not take advantage of the structure of the dual problem and still used semismooth Newton methods to solve the Moreau-Yosida regularization of the dual problem. In the chapter, in terms of the structure of problem (D) , we aim to design an algorithm which could efficiently and fast solve the dual problem (D) .

By setting $x = (\mu, \lambda, p)$, $x_0 = \mu$, and $x_1 = \lambda$, it is quite clear that our dual problem (D) belongs to a general class of multi-block convex optimization problems of the form

$$\min F(x_0, x) := \varphi(x_0) + \psi(x_1) + \phi(x_0, x), \tag{11}$$

where $x_0 \in X_0$, $x = (x_1, \dots, x_s) \in X := X_1 \times \dots \times X_s$ and each X_i is a finite dimensional real Euclidean space. The functions φ , ψ , and ϕ are three closed proper convex functions. Thanks to the structure of (11), in 2015, Chambolle and Dossa (2015) proposed the accelerated alternative descent (AAD) algorithm to solve the problem (11) in which the joint objective function ϕ was quadratic. But the disadvantage is that the AAD method does not take the inexactness of the solutions of the associated subproblems into account. As we know, in some case, it is either impossible or extremely expensive to exactly compute the solution of each subproblem even if it is doable, especially at the early stage of the whole process. For example, if a subproblem is equivalent to solving a large-scale or ill-condition linear system, it is a natural idea to use the iterative methods such as some Krylov-based methods. Hence, it is not suitable for the practical application. Subsequently, when ϕ is a general closed proper convex function and $\arg \min_{x_0} \varphi(x_0) + \phi(x_0, x)$ could be computed exactly, Sun et al. (2016) proposed an inexact accelerated block coordinate descent (iABCD) method to solve least squares semidefinite programming (LSSDP) via its dual. The basic idea of the iABCD method is firstly applying the Danskin-type theorem to reduce the two block nonsmooth terms into only one block and then using APG method to solve the reduced problem. More importantly, the powerful inexact symmetric Gauss-Seidel (sGS) decomposition technique developed in Li et al. (2015) is the key for designing the iABCD method. Additionally, the authors proved that the iABCD method has the $O(1/k^2)$ iteration complexity when the subproblems are solved approximately subject to certain inexactness criteria.

However, for the situation the subproblem with respect to block x_0 could not be solved exactly, one could not no longer use Danskin-type theorem to achieve the goal of reducing it into one block nonsmooth term. To overcome the above bottlenecks, in her PhD thesis (Cui 2016, Chapter 3), Cui proposed an inexact majorized accelerated block coordinate descent (imABCD) method for solving the following unconstrained convex optimization problems with coupled objective functions:

$$\min_{v,w} f(v) + g(w) + \phi(v, w). \quad (12)$$

Under suitable assumptions and certain inexactness criteria, the author can prove that the inexact mABCD method also enjoys the impressive $O(1/k^2)$ iteration complexity.

In this chapter, which is inspired by the success of the iABCD and imABCD methods, we combine their virtues and propose an inexact sGS-based majorized ABCD method (called sGS-imABCD) to solve problem (D). The design of this method combines an inexact 2-block majorized ABCD and the recent advances in the inexact sGS technique. Owing to the convergence results of imABCD method which are given in Cui (2016, Chapter 3), our proposed algorithm could be proven having the $O(1/k^2)$ iteration complexity as well. Moreover, some truly implementable inexactness criteria controlling the accuracy of the generated imABCD

subproblems are analyzed. Specifically, because of two nonsmooth subproblems having the closed-form solutions, it is easy to see that the main computation of our sGS-imABCD algorithm is in solving p -subproblems, which is equivalent to solving the 2×2 block saddle point linear system twice at each iteration. It should be pointed out that the coefficient matrix of the saddle point linear system is fixed. To efficiently solve the linear system, a preconditioned GMRES method is used which leads to the rapid convergence and the robustness with respect to the mesh size h . More importantly, at first glance, it appears that we would need to solve the linear system twice. In practice, in order to avoid this situation and improve the efficiency of our sGS-imABCD algorithm, we design a strategy to approximate the solution for the second linear system. Thus, when a residual error condition is satisfied, the linear system need only to be solved once instead of twice. We should emphasize that such a saving can be significant, especially in the middle and later stages of the whole algorithm. Thus, in terms of the amount of calculation and the discretized error, our sGS-imABCD algorithm is superior to the semismooth Newton method.

Finite Element Approximation and Error Estimates

The goal of this section is to study the approximation of problems (P) and (\tilde{P}) by finite elements.

To achieve our aim, we first consider a family of regular and quasi-uniform triangulations $\{\mathcal{T}_h\}_{h>0}$ of $\bar{\Omega}$. For each cell $T \in \mathcal{T}_h$, let us define the diameter of the set T by $\rho_T := \text{diam } T$ and define σ_T to be the diameter of the largest ball contained in T . The mesh size of the grid is defined by $h = \max_{T \in \mathcal{T}_h} \rho_T$. We suppose that the following regularity assumptions on the triangulation are satisfied which are standard in the context of error estimates.

Assumption 2. *There exist two positive constants κ and τ such that*

$$\frac{\rho_T}{\sigma_T} \leq \kappa \quad \text{and} \quad \frac{h}{\rho_T} \leq \tau,$$

hold for all $T \in \mathcal{T}_h$ and all $h > 0$. Let us define $\bar{\Omega}_h = \bigcup_{T \in \mathcal{T}_h} T$, and let $\Omega_h \subset \Omega$ and Γ_h denote its interior and its boundary, respectively. In the case that Ω is a convex polyhedral domain, we have $\Omega = \Omega_h$. In the case that Ω has a $C^{1,1}$ -boundary Γ , we assumed that $\bar{\Omega}_h$ is convex and that all boundary vertices of $\bar{\Omega}_h$ are contained in Γ , such that $|\Omega \setminus \Omega_h| \leq \hat{c}h^2$, where $|\cdot|$ denotes the measure of the set and $\hat{c} > 0$ is a constant.

On account of the homogeneous boundary condition of the state equation, we use

$$Y_h = \left\{ y_h \in C(\bar{\Omega}) \mid y_{h|T} \in \mathcal{P}_1 \text{ for all } T \in \mathcal{T}_h \text{ and } y_h = 0 \text{ in } \bar{\Omega} \setminus \Omega_h \right\}$$

as the discretized state space, where \mathcal{P}_1 denotes the space of polynomials of degree less than or equal to 1. For a given source term y_r and right-hand side $u \in L^2(\Omega)$, we denote by $y_h(u)$ the approximated state associated with u , which is the unique solution for the following discretized weak formulation:

$$\int_{\Omega_h} \left(\sum_{i,j=1}^n a_{ij} y_{h x_i} v_{h x_j} + c_0 y_h v_h \right) dx = \int_{\Omega_h} (u + y_r) v_h dx \quad \forall v_h \in Y_h. \quad (13)$$

Moreover, $y_h(u)$ can also be expressed by $y_h(u) = \mathcal{S}_h(u + y_r)$, in which \mathcal{S}_h is a discretized version of \mathcal{S} and an injective, self-adjoint operator. The following error estimates are well-known.

Lemma 1 (Ciarlet 1978, Theorem 4.4.6). *For a given $u \in L^2(\Omega)$, let y and $y_h(u)$ be the unique solution of (4) and (13), respectively. Then there exists a constant $c_1 > 0$ independent of h , u , and y_r such that*

$$\|y - y_h(u)\|_{L^2(\Omega)} + h \|\nabla y - \nabla y_h(u)\|_{L^2(\Omega)} \leq c_1 h^2 (\|u\|_{L^2(\Omega)} + \|y_r\|_{L^2(\Omega)}). \quad (14)$$

In particular, this implies $\|\mathcal{S} - \mathcal{S}_h\|_{L^2 \rightarrow L^2} \leq c_1 h^2$ and $\|\mathcal{S} - \mathcal{S}_h\|_{L^2 \rightarrow H^1} \leq c_1 h$.

Considering the homogeneous boundary condition of the adjoint state equation (1), we use

$$U_h = \left\{ u_h \in C(\bar{\Omega}) \mid u_{h|T} \in \mathcal{P}_1 \text{ for all } T \in \mathcal{T}_h \text{ and } u_h = 0 \text{ in } \bar{\Omega} \setminus \Omega_h \right\},$$

as the discretized space of the control u and artificial variable z .

For a given regular and quasi-uniform triangulation \mathcal{T}_h with nodes $\{x_i\}_{i=1}^{N_h}$, let $\{\phi_i(x)\}_{i=1}^{N_h}$ be a set of nodal basis functions associated with nodes $\{x_i\}_{i=1}^{N_h}$, where the basis functions satisfy the following properties:

$$\phi_i(x) \geq 0, \quad \|\phi_i(x)\|_{\infty} = 1 \quad \forall i = 1, 2, \dots, N_h, \quad \sum_{i=1}^{N_h} \phi_i(x) = 1. \quad (15)$$

The elements $z_h \in U_h$, $u_h \in U_h$, and $y_h \in Y_h$ can be represented in the following forms, respectively:

$$u_h = \sum_{i=1}^{N_h} u_i \phi_i(x), \quad z_h = \sum_{i=1}^{N_h} z_i \phi_i(x), \quad y_h = \sum_{i=1}^{N_h} y_i \phi_i(x),$$

and $u_h(x_i) = u_i$, $z_h(x_i) = z_i$, and $y_h(x_i) = y_i$ hold.

Let $U_{ad,h}$ denote the discretized feasible set, which is defined by

$$U_{ad,h} := U_h \cap U_{ad} = \left\{ z_h = \sum_{i=1}^{N_h} z_i \phi_i(x) \mid a \leq z_i \leq b, \forall i = 1, \dots, N_h \right\} \subset U_{ad}.$$

Following the approach of Carstensen (1999), for the error analysis further below, let us introduce a quasi-interpolation operator $\Pi_h : L^1(\Omega_h) \rightarrow U_h$ which provides interpolation estimates. For an arbitrary $w \in L^1(\Omega)$, the operator Π_h is constructed as follows:

$$\Pi_h w = \sum_{i=1}^{N_h} \pi_i(w) \phi_i(x), \quad \pi_i(w) = \frac{\int_{\Omega_h} w(x) \phi_i(x) dx}{\int_{\Omega_h} \phi_i(x) dx}. \tag{16}$$

And we know that

$$w \in U_{ad} \Rightarrow \Pi_h w \in U_{ad,h}, \quad \text{for all } w \in L^1(\Omega). \tag{17}$$

Based on the assumption on the mesh and the control discretization, we extend $\Pi_h w$ to Ω by taking $\Pi_h w = w$ for every $x \in \Omega \setminus \Omega_h$ and have the following estimates of the interpolation error. For the detailed proofs, we refer to Carstensen (1999) and de Los Reyes et al. (2008).

Lemma 2. *There is a constant c_2 independent of h such that*

$$h \|z - \Pi_h z\|_{L^2(\Omega)} + \|z - \Pi_h z\|_{H^{-1}(\Omega)} \leq c_2 h^2 \|z\|_{H^1(\Omega)},$$

holds for all $z \in H^1(\Omega)$.

Now, we can consider a discretized version of problem (\tilde{P}) as

$$\left\{ \begin{array}{l} \min J_h(y_h, u_h, z_h) = \frac{1}{2} \|y_h - y_d\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{4} \|u_h\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{4} \|z_h\|_{L^2(\Omega_h)}^2 \\ \quad + \beta \|z_h\|_{L^1(\Omega_h)} \\ \text{s.t.} \quad y_h = \mathcal{S}_h(u_h + y_r), \\ \quad u_h = z_h, \\ \quad z_h \in U_{ad,h}, \end{array} \right. \tag{\tilde{P}_h}$$

where

$$\|z_h\|_{L^2(\Omega_h)}^2 = \int_{\Omega_h} \left(\sum_{i=1}^{N_h} z_i \phi_i(x) \right)^2 dx, \tag{18}$$

$$\|z_h\|_{L^1(\Omega_h)} = \int_{\Omega_h} \left| \sum_{i=1}^{N_h} z_i \phi_i(x) \right| dx. \tag{19}$$

This implies, for problem (P), we have the following discretized version:

$$\left\{ \begin{aligned} \min_{(y_h, u_h, z_h) \in Y_h \times U_h \times U_h} \quad & J_h(y_h, u_h, z_h) = \frac{1}{2} \|y_h - y_d\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{2} \|u_h\|_{L^2(\Omega_h)}^2 \\ & + \beta \|u_h\|_{L^1(\Omega_h)} \\ \text{s.t.} \quad & y_h = \mathcal{S}_h(u_h + y_r), \\ & u_h \in U_{ad,h}. \end{aligned} \right. \tag{P}_h$$

For problem (P_h), in Wachsmuth and Wachsmuth (2011), the authors gave the following error estimates results.

Theorem 1 (Wachsmuth and Wachsmuth 2011, Proposition 4.3). *Let (y, u) be the optimal solution of problem (P), and (y_h, u_h) be the optimal solution of problem (P_h). For every h₀ > 0, α₀ > 0, there is a constant C > 0 such that for all 0 < α ≤ α₀, 0 < h ≤ h₀ it holds*

$$\|u - u_h\|_{L^2(\Omega)} \leq C(\alpha^{-1}h + \alpha^{-\frac{3}{2}}h^2), \tag{20}$$

where C is a constant independent of h and α.

However, the resulting discretized problem (\tilde{P}_h) is not in a decoupled form as the finite dimensional l^1 -regularization optimization problem usually does, since (18) and (19) do not have a decoupled form. Thus, if we directly apply ADMM algorithm to solve the discretized problem, then the z -subproblem cannot have a closed-form solution. Thus, directly solving (\tilde{P}_h), it cannot make full use of the advantages of ADMM. In order to overcome this bottleneck, we introduce the nodal quadrature formulas to approximately discretized the L^2 -norm and L^1 -norm. Let

$$\|z_h\|_{L_h^2(\Omega_h)} := \left(\sum_{i=1}^{N_h} (z_i)^2 \int_{\Omega_h} \phi_i(x) dx \right)^{\frac{1}{2}}, \tag{21}$$

$$\|z_h\|_{L_h^1(\Omega_h)} := \sum_{i=1}^{N_h} |z_i| \int_{\Omega_h} \phi_i(x) dx, \tag{22}$$

and call them L_h^2 - and L_h^1 -norm, respectively.

It is obvious that the L_h^2 -norm and the L_h^1 -norm can be considered as a weighted l^2 -norm and a weighted l^1 -norm of the coefficient of z_h , respectively. Both of them are norms on U_h . In addition, the L_h^2 -norm is a norm induced by the following inner product:

$$\langle z_h, v_h \rangle_{L_h^2(\Omega_h)} = \sum_{i=1}^{N_h} (z_i v_i) \int_{\Omega_h} \phi_i(x) dx \quad \text{for } z_h, v_h \in U_h. \tag{23}$$

More importantly, the following properties hold.

Proposition 1 (Wathen 1987, Table 1). $\forall z_h \in U_h$, the following inequalities hold:

$$\|z_h\|_{L^2(\Omega_h)}^2 \leq \|z_h\|_{L_h^2(\Omega_h)}^2 \leq c \|z_h\|_{L^2(\Omega_h)}^2, \quad \text{where } c = \begin{cases} 4 & \text{if } n = 2, \\ 5 & \text{if } n = 3. \end{cases} \tag{24}$$

$$\int_{\Omega_h} \left| \sum_{i=1}^n z_i \phi_i(x) \right| dx \leq \|z_h\|_{L_h^1(\Omega_h)}. \tag{25}$$

Thus, based on (22) and (21), we derive a new discretized optimal control problems

$$\left\{ \begin{array}{l} \min J_h(y_h, u_h, z_h) = \frac{1}{2} \|y_h - y_d\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{4} \|u_h\|_{L^2(\Omega_h)}^2 \\ \quad + \frac{\alpha}{4} \|z_h\|_{L_h^2(\Omega_h)}^2 + \beta \|z_h\|_{L_h^1(\Omega_h)} \\ \text{s.t.} \quad y_h = \mathcal{S}_h u_h, \\ \quad \quad u_h = z_h, \\ \quad \quad z_h \in U_{ad,h}. \end{array} \right. \quad (\widetilde{DP}_h)$$

It should be mentioned that the approximate L_h^1 was already used in Wachsmuth and Wachsmuth (2011, Section 4.4). However, different from their discretization schemes, in this chapter, in order to keep the separability of the discrete L^2 -norm with respect to z , we use (21) to approximately discretize it. In addition, although these nodal quadrature formulas incur additional discrete errors, it will be proven that these approximation steps will not change the order of error estimates as shown in (20); see Theorem 1.

To give the error estimates, we first introduce the Karush-Kuhn-Tucker (KKT) conditions. It is clear that problem (\widetilde{P}) is continuous and strongly convex. Therefore, the existence and uniqueness of solution of (\widetilde{P}) are obvious.

Theorem 2 (First-Order Optimality Condition). *Under Assumption 1, (y^*, u^*, z^*) is the optimal solution of $(\tilde{\mathbf{P}})$, if and only if there exists adjoint state $p^* \in H_0^1(\Omega)$ and Lagrange multiplier $\lambda^* \in L^2(\Omega)$, such that the following conditions hold in the weak sense:*

$$y^* = \mathcal{S}(u^* + y_r), \quad (26a)$$

$$p^* = \mathcal{S}^*(y^* - y_d), \quad (26b)$$

$$\frac{\alpha}{2}u^* + p^* + \lambda^* = 0, \quad (26c)$$

$$u^* = z^*, \quad (26d)$$

$$z^* \in U_{ad}, \quad (26e)$$

$$\left\langle \frac{\alpha}{2}z^* - \lambda^*, \tilde{z} - z^* \right\rangle_{L^2(\Omega)} + \beta(\|\tilde{z}\|_{L^1(\Omega)} - \|z^*\|_{L^1(\Omega)}) \geq 0, \quad \forall \tilde{z} \in U_{ad}. \quad (26f)$$

Moreover, we have

$$u^* = \mathbf{P}_{U_{ad}} \left(\frac{1}{\alpha} \text{soft}(-p^*, \beta) \right), \quad (27)$$

where the projection operator $\mathbf{P}_{U_{ad}}(\cdot)$ and the soft thresholding operator $\text{soft}(\cdot, \cdot)$ are defined as follows, respectively:

$$\begin{aligned} \mathbf{P}_{U_{ad}}(v(x)) &:= \max\{a, \min\{v(x), b\}\}, \\ \text{soft}(v(x), \beta) &:= \text{sgn}(v(x)) \circ \max(|v(x)| - \beta, 0). \end{aligned} \quad (28)$$

In addition, the optimal control u has the regularity $u \in H^1(\Omega)$.

Analogous to the continuous problem $(\tilde{\mathbf{P}})$, the discretized problem $(\tilde{\mathbf{DP}}_h)$ is also a strictly convex problem, which is uniquely solvable. We derive the following first-order optimality conditions, which are necessary and sufficient for the optimal solution of $(\tilde{\mathbf{DP}}_h)$.

Theorem 3 (Discrete First-order Optimality Condition). *(u_h, z_h, y_h) is the optimal solution of $(\tilde{\mathbf{DP}}_h)$, if and only if there exist an adjoint state p_h and a Lagrange multiplier λ_h , such that the following conditions are satisfied:*

$$y_h = \mathcal{S}_h(u_h + y_r), \quad (29a)$$

$$p_h = \mathcal{S}_h^*(y_h - y_d), \quad (29b)$$

$$\frac{\alpha}{2}u_h + p_h + \lambda_h = 0, \quad (29c)$$

$$u_h = z_h, \tag{29d}$$

$$z_h \in U_{ad,h}, \tag{29e}$$

$$\begin{aligned} \left\langle \frac{\alpha}{2} z_h, \tilde{z}_h - z_h \right\rangle_{L^2_h(\Omega_h)} - \langle \lambda_h, \tilde{z}_h - z_h \rangle_{L^2(\Omega_h)} \\ + \beta \left(\|\tilde{z}_h\|_{L^1_h(\Omega_h)} - \|z\|_{L^1_h(\Omega_h)} \right) \geq 0, \end{aligned} \tag{29f}$$

$$\forall \tilde{z}_h \in U_{ad,h}.$$

Now, let us start to do error estimation. Let (y, u, z) be the optimal solution of problem (\tilde{P}) , and (y_h, u_h, z_h) be the optimal solution of problem (\tilde{DP}_h) . We have the following results.

Theorem 4. *Let (y, u, z) be the optimal solution of problem (\tilde{P}) , and (y_h, u_h, z_h) be the optimal solution of problem (\tilde{DP}_h) . For any $h > 0$ small enough and $\alpha_0 > 0$, there is a constant C such that for all $0 < \alpha \leq \alpha_0$,*

$$\frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega)}^2 + \frac{1}{2} \|y - y_h\|_{L^2(\Omega)}^2 \leq C(h^2 + \alpha h^2 + \alpha^{-1} h^2 + h^3 + \alpha^{-1} h^4 + \alpha^{-2} h^4),$$

where C is a constant independent of h and α .

Proof. Due to the optimality of z and z_h , z and z_h satisfy (26f) and (29f), respectively. Let us use the test function $z_h \in U_{ad,h} \subset U_{ad}$ in (26f) and the test function $\tilde{z}_h := \Pi_h z \in U_{ad,h}$ in (29f); thus, we have

$$\left\langle \frac{\alpha}{2} z - \lambda, z_h - z \right\rangle_{L^2(\Omega)} + \beta \left(\|z_h\|_{L^1(\Omega)} - \|z\|_{L^1(\Omega)} \right) \geq 0, \tag{30}$$

$$\left\langle \frac{\alpha}{2} z_h, \tilde{z}_h - z_h \right\rangle_{L^2_h(\Omega_h)} - \langle \lambda_h, \tilde{z}_h - z_h \rangle_{L^2(\Omega_h)} + \beta \left(\|\tilde{z}_h\|_{L^1_h(\Omega_h)} - \|z_h\|_{L^1_h(\Omega_h)} \right) \geq 0. \tag{31}$$

Because $z_h = 0$ on $\bar{\Omega} \setminus \Omega_h$, the integrals over Ω can be replaced by integrals over Ω_h in (30), and it can be rewritten as

$$\begin{aligned} \left\langle \frac{\alpha}{2} z - \lambda, z - z_h \right\rangle_{L^2(\Omega_h)} + \beta \left(\|z\|_{L^1(\Omega_h)} - \|z_h\|_{L^1(\Omega_h)} \right) &\leq \left\langle \lambda - \frac{\alpha}{2} z, z \right\rangle_{L^2(\Omega \setminus \Omega_h)} \\ &- \beta \|z\|_{L^1(\Omega \setminus \Omega_h)} \leq \langle \lambda, z \rangle_{L^2(\Omega \setminus \Omega_h)} \leq ch^2, \end{aligned} \tag{32}$$

where the last inequality follows from the boundedness of λ and z and the assumption $|\Omega \setminus \Omega_h| \leq \hat{c}h^2$.

By the definition of the quasi-interpolation operator in (16) and (24) in Proposition 1, we have

$$\langle z_h, \tilde{z}_h - z_h \rangle_{L_h^2(\Omega_h)} = \langle z_h, \tilde{z}_h \rangle_{L_h^2(\Omega_h)} - \|z_h\|_{L_h^2(\Omega_h)}^2 \leq \langle z_h, z - z_h \rangle_{L^2(\Omega_h)}. \quad (33)$$

Thus, (31) can be rewritten as

$$\begin{aligned} & \left\langle -\frac{\alpha}{2}z_h + \lambda_h, z - z_h \right\rangle_{L^2(\Omega_h)} + \langle \lambda_h, \tilde{z}_h - z \rangle_{L^2(\Omega_h)} \\ & \quad - \beta \left(\|\tilde{z}_h\|_{L_h^1(\Omega_h)} - \|z_h\|_{L_h^1(\Omega_h)} \right) \leq 0. \end{aligned} \quad (34)$$

Adding up and rearranging (32) and (34), we obtain

$$\begin{aligned} \frac{\alpha}{2} \|z - z_h\|_{L^2(\Omega_h)}^2 & \leq \langle \lambda - \lambda_h, z - z_h \rangle_{L^2(\Omega_h)} - \langle \lambda_h, \tilde{z}_h - z \rangle_{L^2(\Omega_h)} \\ & \quad + \beta \left(\|z_h\|_{L^1(\Omega_h)} - \|z\|_{L^1(\Omega_h)} + \|\tilde{z}_h\|_{L_h^1(\Omega_h)} - \|z_h\|_{L_h^1(\Omega_h)} \right) + ch^2 \\ & = \underbrace{\left\langle \frac{\alpha}{2}(u_h - u) + p_h - p, z - z_h \right\rangle_{L^2(\Omega_h)}}_{I_1} + \underbrace{\left\langle \frac{\alpha}{2}u_h + p_h, \tilde{z}_h - z \right\rangle_{L^2(\Omega_h)}}_{I_2} \\ & \quad + \underbrace{\beta \left(\|z_h\|_{L^1(\Omega_h)} - \|z\|_{L^1(\Omega_h)} + \|\tilde{z}_h\|_{L_h^1(\Omega_h)} - \|z_h\|_{L_h^1(\Omega_h)} \right)}_{I_3} + ch^2, \end{aligned} \quad (35)$$

where the second inequality follows from (26c) and (29c).

Next, we first estimate the third term I_3 . By (25) in Proposition 1, we have $\|z_h\|_{L^1(\Omega_h)} \leq \|z_h\|_{L_h^1(\Omega_h)}$. And following from the definition of $\tilde{z}_h = \Pi_h(z)$ and the non-negativity and partition of unity of the nodal basis functions, we get

$$\|\tilde{z}_h\|_{L_h^1(\Omega_h)} = \|\Pi_h(z)\|_{L_h^1(\Omega_h)} = \sum_{i=1}^{N_h} \left| \frac{\int_{\Omega_h} z(x)\phi_i dx}{\int_{\Omega_h} \phi_i dx} \right| \int_{\Omega_h} \phi_i dx \leq \|z\|_{L^1(\Omega_h)}. \quad (36)$$

Thus, we have $I_3 \leq 0$.

For the terms I_1 and I_2 , from $u = z$, $u_h = z_h$, we get

$$\begin{aligned} I_1 + I_2 & = -\frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega_h)}^2 + \langle p_h - p, \tilde{z}_h - z_h \rangle_{L^2(\Omega_h)} \\ & \quad + \left\langle \frac{\alpha}{2}u + p, \tilde{z}_h - z \right\rangle_{L^2(\Omega_h)} + \frac{\alpha}{2} \langle u_h - u, \tilde{z}_h - z \rangle_{L^2(\Omega_h)}. \end{aligned}$$

Then (35) can be rewritten as

$$\begin{aligned}
 \frac{\alpha}{2} \|z - z_h\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega_h)}^2 &\leq \underbrace{\langle p_h - p, \tilde{z}_h - z_h \rangle_{L^2(\Omega_h)}}_{I_4} + \underbrace{\left\langle \frac{\alpha}{2} u + p, \tilde{z}_h - z \right\rangle_{L^2(\Omega_h)}}_{I_5} \\
 &\quad + \underbrace{\frac{\alpha}{2} \langle u_h - u, \tilde{z}_h - z \rangle_{L^2(\Omega_h)}}_{I_6} + ch^2.
 \end{aligned}
 \tag{37}$$

For the term I_4 , let $\tilde{p}_h = \mathcal{S}_h^*(y - y_d)$, and we have

$$\begin{aligned}
 I_4 &= \langle p_h - \tilde{p}_h + \tilde{p}_h - p, \tilde{z}_h - z_h \rangle_{L^2(\Omega_h)} \\
 &= -\|y - y_h\|_{L^2(\Omega_h)}^2 + \underbrace{\langle y_h - y, (\mathcal{S}_h - \mathcal{S})(\tilde{z}_h + y_r) - \mathcal{S}(z - \tilde{z}_h) \rangle_{L^2(\Omega_h)}}_{I_7} \\
 &\quad + \underbrace{\langle y - y_d, (\mathcal{S}_h - \mathcal{S})(\tilde{z}_h - z_h) \rangle_{L^2(\Omega_h)}}_{I_8}.
 \end{aligned}$$

Consequently,

$$\frac{\alpha}{2} \|z - z_h\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega_h)}^2 + \|y - y_h\|_{L^2(\Omega_h)}^2 \leq I_5 + I_6 + I_7 + I_8 + ch^2.
 \tag{38}$$

In order to further estimate (38), we will discuss each of these items from I_5 to I_8 in turn. Firstly, from the regularity of the optimal control u , i.e., $u \in H^1(\Omega)$, and (27), we know that

$$\|u\|_{H^1(\Omega)} \leq \frac{1}{\alpha} \|p\|_{H^1(\Omega)} + \left(\frac{\beta}{\alpha} + |a| + b \right) \mathcal{M}(\Omega),
 \tag{39}$$

where $\mathcal{M}(\Omega)$ denotes the measure of the Ω . Then we have

$$\left\| \frac{\alpha}{2} u + p \right\|_{H^1(\Omega)} \leq \frac{3}{2} \|p\|_{H^1(\Omega)} + \frac{1}{2} (\beta + \alpha|a| + \alpha b) \mathcal{M}(\Omega).$$

Moreover, due to the boundedness of the optimal control u , the state y , the adjoint state p , and the operator \mathcal{S} , we can choose a large enough constant $L > 0$ independent of α, h and a constant α_0 , such that for all $0 < \alpha \leq \alpha_0$ and $h > 0$, the following inequation holds:

$$\begin{aligned}
 \frac{3}{2} \|p\|_{H^1(\Omega)} + (\beta + \alpha a + \alpha b) \mathcal{M}(\Omega) + \|y - y_d\|_{L^2(\Omega)} + \|y_r\|_{L^2(\Omega)} \\
 + \|\mathcal{S}\|_{\mathcal{L}(H^{-1}, L^2)} + \sup_{u_h \in U_{ad,h}} \|u_h\| \leq L.
 \end{aligned}
 \tag{40}$$

From (40) and $u = z$, we have $\|z\|_{H^1(\Omega)} \leq \alpha^{-1}L$. Thus, for the term I_5 , utilizing Lemma 2, we have

$$I_5 \leq \frac{\alpha}{2} \|u + p\|_{H^1(\Omega_h)} \|\tilde{z}_h - z\|_{H^{-1}(\Omega_h)} \leq c_2 L \|z\|_{H^1(\Omega_h)} h^2 \leq c_2 L^2 \alpha^{-1} h^2. \quad (41)$$

For terms I_6 and I_7 , using Hölder's inequality, Lemma 1, and Lemma 2, we have

$$I_6 \leq \frac{\alpha}{4} \|u_h - u\|_{L^2(\Omega_h)}^2 + \frac{\alpha}{4} \|\tilde{z}_h - z\|_{L^2(\Omega_h)}^2 \leq \frac{\alpha}{4} \|u_h - u\|_{L^2(\Omega_h)}^2 + \frac{c_2^2 L^2 \alpha^{-1}}{4} h^2, \quad (42)$$

and

$$\begin{aligned} I_7 &\leq \frac{1}{2} \|y - y_h\|_{L^2(\Omega_h)}^2 + 2\|\mathbf{S}_h - \mathbf{S}\|_{\mathcal{L}(L^2, L^2)}^2 (\|\tilde{z}_h\|_{L^2(\Omega_h)}^2 + \|y_r\|_{L^2(\Omega_h)}^2) \\ &\quad + \|\mathbf{S}\|_{\mathcal{L}(H^{-1}, L^2)} \|z - \tilde{z}_h\|_{H^{-1}(\Omega_h)}^2 \\ &\leq \frac{1}{2} \|y - y_h\|_{L^2(\Omega_h)}^2 + 2c_1^2 L^2 h^4 + c_2^2 L^3 \alpha^{-2} h^4. \end{aligned} \quad (43)$$

Finally, about the term I_8 , we have

$$\begin{aligned} I_8 &\leq \|y - y_d\|_{L^2(\Omega_h)} \|\mathbf{S}_h - \mathbf{S}\|_{\mathcal{L}(L^2, L^2)} (\|\tilde{z}_h - z\|_{L^2(\Omega_h)} + \|z - z_h\|_{L^2(\Omega_h)}) \\ &\leq c_1 L h^2 (c_2 L \alpha^{-1} h + \|z - z_h\|_{L^2(\Omega_h)}) \\ &\leq \frac{\alpha}{4} \|z - z_h\|_{L^2(\Omega_h)}^2 + c_1 c_2 \alpha^{-1} L^2 h^3 + 4c_1^2 L^2 \alpha^{-1} h^4. \end{aligned} \quad (44)$$

Substituting (41), (42), (43), and (44) into (38) and rearranging, we get

$$\frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega_h)}^2 + \frac{1}{2} \|y - y_h\|_{L^2(\Omega_h)}^2 \leq C(h^2 + \alpha^{-1}h^2 + \alpha^{-1}h^3 + \alpha^{-1}h^4 + \alpha^{-2}h^4),$$

where $C > 0$ is a properly chosen constant. Using again the assumption $|\Omega \setminus \Omega_h| \leq ch^2$, we can get

$$\frac{\alpha}{2} \|u - u_h\|_{L^2(\Omega)}^2 + \frac{1}{2} \|y - y_h\|_{L^2(\Omega)}^2 \leq C(h^2 + \alpha h^2 + \alpha^{-1}h^2 + h^3 + \alpha^{-1}h^4 + \alpha^{-2}h^4).$$

Corollary 1. *Let (y, u, z) be the optimal solution of problem $(\tilde{\mathbf{P}})$, and (y_h, u_h, z_h) be the optimal solution of problem $(\tilde{\mathbf{DP}}_h)$. For every $h_0 > 0$, $\alpha_0 > 0$, there is a constant $C > 0$ such that for all $0 < \alpha \leq \alpha_0$, $0 < h \leq h_0$ it holds*

$$\|u - u_h\|_{L^2(\Omega)} \leq C(\alpha^{-1}h + \alpha^{-\frac{3}{2}}h^2),$$

where C is a constant independent of h and α . □

An Inexact Heterogeneous ADMM Algorithm

In this section, we will introduce the ihADMM algorithm with the aim of solving (\overline{DP}_h) to moderate accuracy. Firstly, let us define following stiffness and mass matrices:

$$K_h = (a_h(\phi_i, \phi_j))_{i,j=1}^{N_h}, \quad M_h = \left(\int_{\Omega_h} \phi_i \phi_j dx \right)_{i,j=1}^{N_h}, \quad \square$$

where the bilinear form $a_h(\cdot, \cdot)$ is defined as

$$a_h(y, v) = \int_{\Omega_h} \left(\sum_{i,j=1}^n a_{ji} y_{x_i} v_{x_i} + c_0 y v \right) dx.$$

Due to the quadrature formulas (21) and (22), a lumped mass matrix $W_h = \text{diag} \left(\int_{\Omega_h} \phi_i(x) dx \right)_{i=1}^{N_h}$ is introduced. Moreover, by (24) in Proposition 1, we have the following results about the mass matrix M_h and the lump mass matrix W_h .

Proposition 2. $\forall z \in \mathbb{R}^{N_h}$, the following inequalities hold:

$$\|z\|_{M_h}^2 \leq \|z\|_{W_h}^2 \leq c \|z\|_{M_h}^2, \quad \text{where } c = \begin{cases} 4 & \text{if } n = 2, \\ 5 & \text{if } n = 3. \end{cases}$$

An Inexact Heterogeneous ADMM Algorithm

Denoting by $y_{d,h} := \sum_{i=1}^{N_h} y_d^i \phi_i(x)$ and $y_{c,h} := \sum_{i=1}^{N_h} y_c^i \phi_i(x)$ the L^2 -projection of y_d and y_r onto Y_h , respectively, and identifying discretized functions with their coefficient vectors, we can rewrite the problem (\overline{DP}_h) as a matrix-vector form:

$$\left\{ \begin{array}{ll} \min_{(y,u,z) \in \mathbb{R}^{3N_h}} & \frac{1}{2} \|y - y_d\|_{M_h}^2 + \frac{\alpha}{4} \|u\|_{M_h}^2 + \frac{\alpha}{4} \|z\|_{W_h}^2 + \|W_h z\|_1 \\ \text{s.t.} & K_h y = M_h(u + y_r), \\ & u = z, \\ & z \in [a, b]^{N_h}. \end{array} \right. \quad (\overline{DP}_h)$$

By Assumption 1, we have the stiffness matrix K_h is a symmetric positive definite matrix. Then problem (\overline{DP}_h) can be rewritten as the following reduced form:

$$\begin{cases} \min_{(u,z) \in \mathbb{R}^{2N_h}} & f(u) + g(z) \\ \text{s.t.} & u = z. \end{cases} \quad (\overline{\text{RDP}}_h)$$

where

$$\begin{aligned} f(u) &= \frac{1}{2} \|K_h^{-1} M_h(u + y_r) - y_d\|_{M_h}^2 + \frac{\alpha}{4} \|u\|_{M_h}^2, \\ g(z) &= \frac{\alpha}{4} \|z\|_{W_h}^2 + \beta \|W_h z\|_1 + \delta_{[a,b]^{N_h}}(z). \end{aligned} \quad (45)$$

To solve $(\overline{\text{RDP}}_h)$ by using ADMM-type algorithm, we first introduce the augmented Lagrangian function for $(\overline{\text{RDP}}_h)$. According to three possible choices of norms (\mathbb{R}^{N_h} norm, W_h -weighted norm, and M_h -weighted norm), for the augmented Lagrangian function, there are three versions as follows: for given $\sigma > 0$,

$$\mathcal{L}_\sigma^1(u, z; \lambda) := f(u) + g(z) + \langle \lambda, u - z \rangle + \frac{\sigma}{2} \|u - z\|^2, \quad (46)$$

$$\mathcal{L}_\sigma^2(u, z; \lambda) := f(u) + g(z) + \langle \lambda, M_h(u - z) \rangle + \frac{\sigma}{2} \|u - z\|_{W_h}^2, \quad (47)$$

$$\mathcal{L}_\sigma^3(u, z; \lambda) := f(u) + g(z) + \langle \lambda, M_h(u - z) \rangle + \frac{\sigma}{2} \|u - z\|_{M_h}^2. \quad (48)$$

Then based on these three versions of augmented Lagrangian function, we give the following four versions of ADMM-type algorithm for $(\overline{\text{RDP}}_h)$ at k -th iteration: for given $\tau > 0$ and $\sigma > 0$,

$$\begin{cases} u^{k+1} = \arg \min_u f(u) + \langle \lambda^k, u - z^k \rangle + \sigma/2 \|u - z^k\|^2, \\ z^{k+1} = \arg \min_z g(z) + \langle \lambda^k, u^{k+1} - z \rangle + \sigma/2 \|u^{k+1} - z\|^2, \\ \lambda^{k+1} = \lambda^k + \tau \sigma (u^{k+1} - z^{k+1}). \end{cases} \quad (\text{ADMM1})$$

$$\begin{cases} u^{k+1} = \arg \min_u f(u) + \langle \lambda^k, M_h(u - z^k) \rangle + \sigma/2 \|u - z^k\|_{W_h}^2, \\ z^{k+1} = \arg \min_z g(z) + \langle \lambda^k, M_h(u^{k+1} - z) \rangle + \sigma/2 \|u^{k+1} - z\|_{W_h}^2, \\ \lambda^{k+1} = \lambda^k + \tau \sigma (u^{k+1} - z^{k+1}). \end{cases} \quad (\text{ADMM2})$$

$$\begin{cases} u^{k+1} = \arg \min_u f(u) + \langle \lambda^k, M_h(u - z^k) \rangle + \sigma/2 \|u - z^k\|_{M_h}^2, \\ z^{k+1} = \arg \min_z g(z) + \langle \lambda^k, M_h(u^{k+1} - z) \rangle + \sigma/2 \|u^{k+1} - z\|_{M_h}^2, \\ \lambda^{k+1} = \lambda^k + \tau \sigma (u^{k+1} - z^{k+1}). \end{cases} \tag{ADMM3}$$

$$\begin{cases} u^{k+1} = \arg \min_u f(u) + \langle \lambda^k, M_h(u - z^k) \rangle + \sigma/2 \|u - z^k\|_{M_h}^2, \\ z^{k+1} = \arg \min_z g(z) + \langle \lambda^k, M_h(u^{k+1} - z) \rangle + \sigma/2 \|u^{k+1} - z\|_{W_h}^2, \\ \lambda^{k+1} = \lambda^k + \tau \sigma (u^{k+1} - z^{k+1}). \end{cases} \tag{ADMM4}$$

As one may know, (ADMM1) is actually the classical ADMM for $(\overline{\text{RDP}}_h)$. The remaining three ADMM-type algorithms are proposed based on the structure of $(\overline{\text{RDP}}_h)$. Now, let us start to analyze and compare the advantages and disadvantages of the four algorithms. Firstly, we focus on the z -subproblem in each algorithm. Since both identity matrix I and lumped mass matrix W_h are diagonal, it is clear that all the z -subproblems in (ADMM1), (ADMM2), and (ADMM4) have a closed form solution, except for the z -subproblem in (ADMM3). Specifically, for z -subproblem in (ADMM1), the closed-form solution could be given by

$$z^k = P_{U_{ad}} \left(\left(\frac{\alpha}{2} W_h + \sigma I \right)^{-1} W_h \text{soft}(W_h^{-1}(\sigma u^{k+1} + \lambda^k), \beta) \right). \tag{49}$$

Similarly, for z -subproblems in (ADMM2) and (ADMM4), the closed-form solutions could be given by

$$z^{k+1} = P_{U_{ad}} \left(\frac{1}{\sigma + 0.5\alpha} \text{soft} \left(\sigma u^{k+1} + W_h^{-1} M_h \lambda^k, \beta \right) \right), \tag{50}$$

Next, let us analyze the structure of u -subproblem in each algorithm. For (ADMM1), the first subproblem at k -th iteration is equivalent to solving the following linear system:

$$\begin{bmatrix} M_h & 0 & K_h \\ 0 & \frac{\alpha}{2} M_h + \sigma I & -M_h \\ K_h & -M_h & 0 \end{bmatrix} \begin{bmatrix} y^{k+1} \\ u^{k+1} \\ p^{k+1} \end{bmatrix} = \begin{bmatrix} M_h y_d \\ \sigma z^k - \lambda^k \\ M_h y_r \end{bmatrix}. \tag{51}$$

Similarly, the u -subproblem in (ADMM2) can be converted into the following linear system:

$$\begin{bmatrix} M_h & 0 & K_h \\ 0 & \frac{\alpha}{2} M_h + \sigma W_h & -M_h \\ K_h & -M_h & 0 \end{bmatrix} \begin{bmatrix} y^{k+1} \\ u^{k+1} \\ p^{k+1} \end{bmatrix} = \begin{bmatrix} M_h y_d \\ \sigma W_h z^k - M_h \lambda^k \\ M_h y_r \end{bmatrix}. \quad (52)$$

However, the u -subproblem in both (ADMM3) and (ADMM4) can be rewritten as

$$\begin{bmatrix} M_h & 0 & K_h \\ 0 & (0.5\alpha + \sigma)M_h & -M_h \\ K_h & -M_h & 0 \end{bmatrix} \begin{bmatrix} y^{k+1} \\ u^{k+1} \\ p^{k+1} \end{bmatrix} = \begin{bmatrix} M_h y_d \\ M_h(\sigma z^k - \lambda^k) \\ M_h y_r \end{bmatrix}. \quad (53)$$

In (53), since $p^{k+1} = (0.5\alpha + \sigma)u^{k+1} - \sigma z^k + \lambda^k$, it is obvious that (53) can be reduced into the following system by eliminating the variable p without any computational cost:

$$\begin{bmatrix} \frac{1}{0.5\alpha + \sigma} M_h & K_h \\ -K_h & M_h \end{bmatrix} \begin{bmatrix} y^{k+1} \\ u^{k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{0.5\alpha + \sigma} (K_h(\sigma z^k - \lambda^k) + M_h y_d) \\ -M_h y_r \end{bmatrix}, \quad (54)$$

while, reduced forms of (51) and (52), both involve the inversion of M_h .

For abovementioned reasons, we prefer to use (ADMM4), which is called the heterogeneous ADMM (hADMM). However, in general, it is expensive and unnecessary to exactly compute the solution of saddle point system (54) even if it is doable, especially at the early stage of the whole process. Based on the structure of (54), it is a natural idea to use the iterative methods such as some Krylov-based methods. Hence, taking the inexactness of the solution of u -subproblem into account, a more practical inexact heterogeneous ADMM (ihADMM) algorithm is proposed.

Due to the inexactness of the proposed algorithm, we first introduce an error tolerance. Throughout this chapter, let $\{\epsilon_k\}$ be a summable sequence of nonnegative numbers, and define

$$C_1 := \sum_{k=0}^{\infty} \epsilon_k \leq \infty, \quad C_2 := \sum_{k=0}^{\infty} \epsilon_k^2 \leq \infty. \quad (55)$$

The details of our ihADMM algorithm is shown in Algorithm 1 to solve (\overline{DP}_h) .

Convergence Results of ihADMM

For the ihADMM (Algorithm 1), in this section, we establish the global convergence and the iteration complexity results in non-ergodic sense for the sequence generated by Algorithm 1.

Algorithm 1 Inexact heterogeneous ADMM algorithm for (\overline{DP}_h)

Input : $(z^0, u^0, \lambda^0) \in \text{dom}(\delta_{[a,b]}(\cdot)) \times \mathbb{R}^n \times \mathbb{R}^n$ and parameters $\sigma > 0, \tau > 0$. let $\{\epsilon_k\}$ be a summable sequence of nonnegative numbers, and define

$$C_1 := \sum_{k=0}^{\infty} \epsilon_k \leq \infty, \quad C_2 := \sum_{k=0}^{\infty} \epsilon_k^2 \leq \infty.$$

Set $k = 1$.

Output : u^k, z^k, λ^k

Step 1 Find an minimizer (inexact)

$$u^{k+1} = \arg \min f(u) + (M_h \lambda^k, u - z^k) + \frac{\sigma}{2} \|u - z^k\|_{M_h}^2 - \langle \delta^k, u \rangle,$$

where the error vector δ^k satisfies $\|\delta^k\|_2 \leq \epsilon_k$

Step 2 Compute z^k as follows:

$$z^{k+1} = \arg \min g(z) + (M_h \lambda^k, u^{k+1} - z) + \frac{\sigma}{2} \|u^{k+1} - z\|_{W_h}^2$$

Step 3 Compute

$$\lambda^{k+1} = \lambda^k + \tau \sigma (u^{k+1} - z^{k+1}).$$

Step 4 If a termination criterion is not met, set $k := k + 1$ and go to Step 1

Before giving the proof of Theorem 5, we first provide a lemma, which is useful for analyzing the non-ergodic iteration complexity of ihADMM and introduced in Chen and Toh (2017).

Lemma 3. *If a sequence $\{a_i\} \in \mathbb{R}$ satisfies the following conditions:*

$$a_i \geq 0 \text{ for any } i \geq 0 \quad \text{and} \quad \sum_{i=0}^{\infty} a_i = \bar{a} < \infty.$$

Then we have $\min_{i=1, \dots, k} \{a_i\} \leq \frac{\bar{a}}{k}$, and $\lim_{k \rightarrow \infty} \{k \cdot \min_{i=1, \dots, k} \{a_i\}\} = 0$. □

For the convenience of the iteration complexity analysis below, we define the function $R_h : (u, z, \lambda) \rightarrow [0, \infty)$ by

$$R_h(u, z, \lambda) = \|M_h \lambda + \nabla f(u)\|^2 + \text{dist}^2(0, -M_h \lambda + \partial g(z)) + \|u - z\|^2. \quad (56)$$

By the definitions of $f(u)$ and $g(z)$ in (45), it is obvious that $f(u)$ and $g(z)$ are both closed, proper, and convex functions. Since M_h and K_h are symmetric positive

definite matrices, we know the gradient operator ∇f is strongly monotone, and we have

$$\langle \nabla f(u_1) - \nabla f(u_2), u_1 - u_2 \rangle = \|u_1 - u_2\|_{\Sigma_f}^2, \quad (57)$$

where $\Sigma_f = \frac{\alpha}{2}M_h + M_h K_h^{-1} M_h K_h^{-1} M_h$ is symmetric positive definite. Moreover, the subdifferential operator ∂g is a maximal monotone operators, e.g.,

$$\langle \varphi_1 - \varphi_2, z_1 - z_2 \rangle \geq \frac{\alpha}{2} \|z_1 - z_2\|_{W_h}^2 \quad \forall \varphi_1 \in \partial g(z_1), \varphi_2 \in \partial g(z_2). \quad (58)$$

For the subsequent convergence analysis, we denote

$$\bar{u}^{k+1} := \arg \min f(u) + \langle M_h \lambda^k, u - z^k \rangle + \frac{\sigma}{2} \|u - z^k\|_{M_h}^2, \quad (59)$$

$$\bar{z}^{k+1} := \text{P}_{U_{ad}} \left(\frac{1}{\sigma + 0.5\alpha} \text{soft} \left(\sigma \bar{u}^{k+1} + W_h^{-1} M_h \lambda^k, \beta \right) \right), \quad (60)$$

which are the exact solutions at the $(k+1)$ -th iteration in Algorithm 1. The following results show the gap between (u^{k+1}, z^{k+1}) and $(\bar{u}^{k+1}, \bar{z}^{k+1})$ in terms of the given error tolerance $\|\delta^k\|_2 \leq \epsilon_k$.

Lemma 4. *Let $\{(u^{k+1}, z^{k+1})\}$ be the sequence generated by Algorithm 1, and $\{\bar{u}^{k+1}\}, \{\bar{z}^{k+1}\}$ be defined in (59) and (60). Then for any $k \geq 0$, we have*

$$\|u^{k+1} - \bar{u}^{k+1}\| = \|(\sigma M_h + \Sigma_f)^{-1} \delta^k\| \leq \rho \epsilon_k, \quad (61)$$

$$\|z^{k+1} - \bar{z}^{k+1}\| \leq \frac{\sigma}{\sigma + 0.5\alpha} \|u^{k+1} - \bar{u}^{k+1}\| \leq \frac{\rho\sigma}{\sigma + 0.5\alpha} \epsilon_k, \quad (62)$$

where $\rho := \|(\sigma M_h + \Sigma_f)^{-1}\|$. □

Proof. By the optimality conditions at point (u^{k+1}, z^{k+1}) and $(\bar{u}^{k+1}, \bar{z}^{k+1})$, we have

$$\Sigma_f u^{k+1} - M_h K_h^{-1} M_h y_d + M_h \lambda^k + \sigma M_h (u^{k+1} - z^k) - \delta^k = 0,$$

$$\Sigma_f \bar{u}^{k+1} - M_h K_h^{-1} M_h y_d + M_h \lambda^k + \sigma M_h (\bar{u}^{k+1} - z^k) = 0;$$

thus,

$$u^{k+1} - \bar{u}^{k+1} = (\sigma M_h + \Sigma_f)^{-1} \delta^k$$

which implies (61). From (50) and (60), and the fact that the projection operator $\Pi_{[a,b]}(\cdot)$ and soft thresholding operator $\text{soft}(\cdot, \cdot)$ are nonexpansive, we get

$$\|z^{k+1} - \bar{z}^{k+1}\| \leq \frac{\sigma}{\sigma + 0.5\alpha} \|u^{k+1} - \bar{u}^{k+1}\|,$$

which implies (62). The proof is completed.

Next, for $k \geq 0$, we define

$$\begin{aligned} r^k &= u^k - z^k, & \bar{r}^k &= \bar{u}^k - \bar{z}^k \\ \tilde{\lambda}^{k+1} &= \lambda^k + \sigma r^{k+1}, & \bar{\lambda}^{k+1} &= \lambda^k + \tau \sigma \bar{r}^{k+1}, & \hat{\lambda}^{k+1} &= \lambda^k + \sigma \bar{r}^{k+1}, \end{aligned}$$

and give two inequalities which is essential for establishing both the global convergence and the iteration complexity of our ihADMM. For the details of the proof, one can see in Appendix.

Proposition 3. *Let $\{(u^k, z^k, \lambda^k)\}$ be the sequence generated by Algorithm 1 and (u^*, z^*, λ^*) be the KKT point of problem (RDP_h). Then for $k \geq 0$, we have*

$$\begin{aligned} \langle \delta^k, u^{k+1} - u^* \rangle &+ \frac{1}{2\tau\sigma} \|\lambda^k - \lambda^*\|_{M_h}^2 + \frac{\sigma}{2} \|z^k - z^*\|_{M_h}^2 \\ &- \frac{1}{2\tau\sigma} \|\lambda^{k+1} - \lambda^*\|_{M_h}^2 - \frac{\sigma}{2} \|z^{k+1} - z^*\|_{M_h}^2 \geq \|u^{k+1} - u^*\|_T^2 \\ &+ \frac{\sigma}{2} \|z^{k+1} - z^*\|_{2W_h - M_h}^2 + \frac{\sigma}{2} \|r^{k+1}\|_{W_h - \tau M_h}^2 + \frac{\sigma}{2} \|u^{k+1} - z^k\|_{M_h}^2, \end{aligned} \tag{63}$$

where $T := \Sigma_f - \frac{\sigma}{2}(W_h - M_h)$. □

Proposition 4. *Let $\{(u^k, z^k, \lambda^k)\}$ be the sequence generated by Algorithm 1, (u^*, z^*, λ^*) be the KKT point of the problem (RDP_h) and $\{\bar{u}^k\}$ and $\{\bar{z}^k\}$ be two sequences defined in (59) and (60), respectively. Then for $k \geq 0$, we have*

$$\begin{aligned} &\frac{1}{2\tau\sigma} \|\lambda^k - \lambda^*\|_{M_h}^2 + \frac{\sigma}{2} \|z^k - z^*\|_{M_h}^2 - \frac{1}{2\tau\sigma} \|\bar{\lambda}^{k+1} - \lambda^*\|_{M_h}^2 - \frac{\sigma}{2} \|\bar{z}^{k+1} - z^*\|_{M_h}^2 \\ \geq &\|\bar{u}^{k+1} - u^*\|_T^2 + \frac{\sigma}{2} \|\bar{z}^{k+1} - z^*\|_{2W_h - M_h}^2 + \frac{\sigma}{2} \|\bar{r}^{k+1}\|_{W_h - \tau M_h}^2 + \frac{\sigma}{2} \|\bar{u}^{k+1} - z^k\|_{M_h}^2, \end{aligned} \tag{64}$$

where $T := \Sigma_f - \frac{\sigma}{2}(W_h - M_h)$. □

Then based on former results, we have the following convergence results.

Theorem 5. *Let $(y^*, u^*, z^*, p^*, \lambda^*)$ be the KKT point of (DP_h), then the sequence $\{(u^k, z^k, \lambda^k)\}$ is generated by Algorithm 1 with the associated state $\{y^k\}$ and adjoint state $\{p^k\}$, and then for any $\tau \in (0, 1]$ and $\sigma \in (0, \frac{1}{4}\alpha]$, we have*

$$\lim_{k \rightarrow \infty} \{\|u^k - u^*\| + \|z^k - z^*\| + \|\lambda^k - \lambda^*\|\} = 0 \tag{65}$$

$$\lim_{k \rightarrow \infty} \{\|y^k - y^*\| + \|p^k - p^*\|\} = 0 \quad (66)$$

Moreover, there exists a constant C only depending on the initial point (u^0, z^0, λ^0) and the optimal solution (u^*, z^*, λ^*) such that for $k \geq 1$,

$$\min_{1 \leq i \leq k} \{R_h(u^i, z^i, \lambda^i)\} \leq \frac{C}{k}, \quad \lim_{k \rightarrow \infty} \left(k \times \min_{1 \leq i \leq k} \{R_h(u^i, z^i, \lambda^i)\} \right) = 0. \quad (67)$$

where $R_h(\cdot)$ is defined as in (56).

Proof. It is easy to see that (u^*, z^*) is the unique optimal solution of discrete problem (RDP_h) if and only if there exists a Lagrangian multiplier λ^* such that the following Karush-Kuhn-Tucker (KKT) conditions hold:

$$-M_h \lambda^* = \nabla f(u^*), \quad (68a)$$

$$M_h \lambda^* \in \partial g(z^*), \quad (68b)$$

$$u^* = z^*. \quad (68c)$$

In the inexact heterogeneous ADMM iteration scheme, the optimality conditions for (u^{k+1}, z^{k+1}) are

$$\delta^k - (M_h \lambda^k + \sigma M_h (u^{k+1} - z^k)) = \nabla f(u^{k+1}), \quad (69a)$$

$$M_h \lambda^k + \sigma W_h (u^{k+1} - z^{k+1}) \in \partial g(z^{k+1}). \quad (69b)$$

Next, let us first prove the **global convergence of iteration sequences**, e.g., establish the proof of (65) and (66).

The first step is to show that $\{(u^k, z^k, \lambda^k)\}$ is bounded. We define the following sequence θ^k and $\bar{\theta}^k$ with:

$$\theta^k = \left(\frac{1}{\sqrt{2\tau\sigma}} M_h^{\frac{1}{2}} (\lambda^k - \lambda^*), \sqrt{\frac{\sigma}{2}} M_h^{\frac{1}{2}} (z^k - z^*) \right), \quad (70)$$

$$\bar{\theta}^k = \left(\frac{1}{\sqrt{2\tau\sigma}} M_h^{\frac{1}{2}} (\bar{\lambda}^k - \lambda^*), \sqrt{\frac{\sigma}{2}} M_h^{\frac{1}{2}} (\bar{z}^k - z^*) \right).$$

According to Proposition 1, for any $\tau \in (0, 1]$ and $\sigma \in (0, \frac{1}{4}\alpha]$ for, we have $\Sigma_f - \frac{\sigma}{2}(W_h - M_h) \succ 0$, and $W_h - \tau M_h \succ 0$. Then, by Proposition 4, we get $\|\bar{\theta}^{k+1}\|^2 \leq \|\theta^k\|^2$. As a result, we have

$$\|\theta^{k+1}\| \leq \|\bar{\theta}^{k+1}\| + \|\bar{\theta}^{k+1} - \theta^{k+1}\| = \|\theta^k\| + \|\bar{\theta}^{k+1} - \theta^{k+1}\|. \quad (71)$$

Employing Lemma 4, we get

$$\begin{aligned} \|\bar{\theta}^{k+1} - \theta^{k+1}\|^2 &= \frac{1}{2\tau\sigma} \|\bar{\lambda}^{k+1} - \lambda^{k+1}\|_{M_h}^2 + \frac{\sigma}{2} \|\bar{z}^{k+1} - z^{k+1}\|_{M_h}^2 \\ &\leq (2\tau + 1/2)\sigma \|M_h\| \rho^2 \epsilon_k^2 \leq 5/2\sigma \|M_h\| \rho^2 \epsilon_k^2, \end{aligned} \tag{72}$$

which implies $\|\bar{\theta}^{k+1} - \theta^{k+1}\| \leq \sqrt{5/2\sigma \|M_h\|} \rho \epsilon_k$. Hence, for any $k \geq 0$, we have

$$\begin{aligned} \|\theta^{k+1}\| &\leq \|\theta^k\| + \sqrt{5/2\sigma \|M_h\|} \rho \epsilon_k \leq \|\theta^0\| + \sqrt{5/2\sigma \|M_h\|} \rho \sum_{k=0}^{\infty} \epsilon_k \\ &= \|\theta^0\| + \sqrt{5/2\sigma \|M_h\|} \rho C_1 \equiv \bar{\rho}. \end{aligned} \tag{73}$$

From $\|\bar{\theta}^{k+1}\| \leq \|\theta^k\|$, for any $k \geq 0$, we also have $\|\bar{\theta}^{k+1}\| \leq \bar{\rho}$. Therefore, the sequences $\{\theta^k\}$ and $\{\bar{\theta}^k\}$ are bounded. From the definition of $\{\theta^k\}$ and the fact that $M_h > 0$, we can see that the sequences $\{\lambda^k\}$ and $\{z^k\}$ are bounded. Moreover, from updating technique of λ^k , we know $\{u^k\}$ is also bounded. Thus, due to the boundedness of the sequence $\{(u^k, z^k, \lambda^k)\}$, we know the sequence has a subsequence $\{(u^{k_i}, z^{k_i}, \lambda^{k_i})\}$ which converges to an accumulation point $(\bar{u}, \bar{z}, \bar{\lambda})$. Next we should show that $(\bar{u}, \bar{z}, \bar{\lambda})$ is a KKT point and equal to (u^*, z^*, λ^*) .

Again employing Proposition 4, we can derive

$$\begin{aligned} &\sum_{k=0}^{\infty} \left(\|\bar{u}^{k+1} - u^*\|_T^2 + \frac{\sigma}{2} \|\bar{z}^{k+1} - z^*\|_{2W_h - M_h}^2 + \frac{\sigma}{2} \|\bar{r}^{k+1}\|_{W_h - \tau M_h}^2 + \frac{\sigma}{2} \|\bar{u}^{k+1} - z^k\|_{M_h}^2 \right) \\ &\leq \sum_{k=0}^{\infty} (\|\theta^k\|^2 - \|\theta^{k+1}\|^2 + \|\theta^{k+1}\|^2 - \|\bar{\theta}^{k+1}\|^2) \leq \|\theta^0\|^2 + 2\bar{\rho}\sqrt{5/2\sigma \|M_h\|} \rho C_1 < \infty. \end{aligned} \tag{74}$$

Note that $T > 0$, $W_h - M_h > 0$, $W_h - \tau M_h > 0$ and $M_h > 0$, then we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\bar{u}^{k+1} - u^*\| &= 0, & \lim_{k \rightarrow \infty} \|\bar{z}^{k+1} - z^*\| &= 0, \\ \lim_{k \rightarrow \infty} \|\bar{r}^{k+1}\| &= 0, & \lim_{k \rightarrow \infty} \|\bar{u}^{k+1} - z^k\| &= 0. \end{aligned} \tag{75}$$

From the Lemma 4, we can get

$$\begin{aligned} \|u^{k+1} - u^*\| &\leq \|\bar{u}^{k+1} - u^*\| + \|u^{k+1} - \bar{u}^{k+1}\| \leq \|\bar{u}^{k+1} - u^*\| + \rho \epsilon_k, \\ \|z^{k+1} - z^*\| &\leq \|\bar{z}^{k+1} - z^*\| + \|z^{k+1} - \bar{z}^{k+1}\| \leq \|\bar{z}^{k+1} - z^*\| + \rho \epsilon_k. \end{aligned} \tag{76}$$

From the fact that $\lim_{k \rightarrow \infty} \epsilon_k = 0$ and (75), by taking the limit of both sides of (76), we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \|u^{k+1} - u^*\| &= 0, & \lim_{k \rightarrow \infty} \|z^{k+1} - z^*\| &= 0, \\ \lim_{k \rightarrow \infty} \|r^{k+1}\| &= 0, & \lim_{k \rightarrow \infty} \|u^{k+1} - z^k\| &= 0. \end{aligned} \quad (77)$$

Now taking limits for $k_i \rightarrow \infty$ on both sides of (69a), we have

$$\lim_{k_i \rightarrow \infty} (\delta^{k_i} - (M_h \bar{\lambda}^{k_i} + \sigma M_h (u^{k_i+1} - z^{k_i}))) = \lim_{k_i \rightarrow \infty} \nabla f(u^{k_i+1}),$$

which results in $-M_h \bar{\lambda} = \nabla f(u^*)$. Then from (68a), we know $\bar{\lambda} = \lambda^*$. At last, to complete the proof, we need to show that λ^* is the limit of the sequence of $\{\lambda^k\}$.

From (73), we have for any $k > k_i$, $\|\theta^{k+1}\| \leq \|\theta^{k_i}\| + \sqrt{5/2\sigma} \|M_h\| \rho \sum_{j=k_i}^k \epsilon_j$. Since

$$\lim_{k_i \rightarrow \infty} \|\theta^{k_i}\| = 0 \text{ and } \sum_{k=0}^{\infty} \epsilon_k < \infty, \text{ we have that } \lim_{k \rightarrow \infty} \|\theta^k\| = 0, \text{ which implies}$$

$\lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^*\| = 0$. Hence, we have proved the convergence of the sequence $\{(u^{k+1}, z^{k+1}, \lambda^{k+1})\}$, which completes the proof of (65). For the proof of (66), it is easy to show by the definition of the sequence $\{(y^k, p^k)\}$; here we omit it.

At last, we establish the proof of (67), e.g., **the iteration complexity results in non-ergodic sense for the sequence generated by the ihADMM.**

Firstly, by the optimality condition (69a) and (69b) for (u^{k+1}, z^{k+1}) , we have

$$\delta^k + (\tau - 1)\sigma M_h r^{k+1} - \sigma M_h (z^{k+1} - z^k) = M_h \lambda^{k+1} + \nabla f(u^{k+1}), \quad (78a)$$

$$\sigma (W_h - \tau M_h) r^{k+1} \in -M_h \lambda^{k+1} + \partial g(z^{k+1}). \quad (78b)$$

By the definition of R_h and denoting $w^{k+1} := (u^{k+1}, z^{k+1}, \lambda^{k+1})$, we derive

$$\begin{aligned} R_h(w^{k+1}) &= \|M_h \lambda^{k+1} + \nabla f(u^{k+1})\|^2 + \text{dist}^2(0, -M_h \lambda^{k+1} + \partial g(z^{k+1})) \\ &\quad + \|u^{k+1} - z^{k+1}\|^2 \leq 2\|\delta^k\|^2 + \eta \|r^{k+1}\|^2 + 4\sigma^2 \|M_h\| \|u^{k+1} - z^k\|_{M_h}^2, \end{aligned} \quad (79)$$

where $\eta := 2(\tau - 1)^2 \sigma^2 \|M_h\|^2 + 2\sigma^2 \|M_h\|^2 + \sigma^2 \|W_h - \tau M_h\|^2 + 1$.

In order to get an upper bound for $R_h(w^{k+1})$, we will use (63) in Proposition 3. First, by the definition of θ^k and (73), for any $k \geq 0$ we can easily have

$$\|\lambda^k - \lambda^*\| \leq \bar{\rho} \sqrt{\frac{2\tau\sigma}{\|M_h^{-1}\|}}, \quad \|z^k - z^*\| \leq \bar{\rho} \sqrt{\frac{2}{\sigma \|M_h^{-1}\|}}. \quad \square$$

Next, we should give an upper bound for $\langle \delta^k, u^{k+1} - u^* \rangle$:

$$\begin{aligned} \langle \delta^k, u^{k+1} - u^* \rangle &\leq \|\delta^k\| (\|u^{k+1} - z^{k+1}\| + \|z^{k+1} - z^*\|) \\ &\leq \left(\left(1 + \frac{2}{\sqrt{\tau}} \right) \frac{2\sqrt{2}\bar{\rho}}{\sqrt{\tau\sigma}\|M_h^{-1}\|} \right) \|\delta^k\| \equiv \bar{\eta}\|\delta^k\|. \end{aligned} \tag{80}$$

Then by (63) in Proposition 3, we have

$$\begin{aligned} \sum_{k=0}^{\infty} \left(\frac{\sigma}{2} \|r^{k+1}\|_{W_h - \tau M_h}^2 + \frac{\sigma}{2} \|u^{k+1} - z^k\|_{M_h}^2 \right) &\leq \sum_{k=0}^{\infty} (\|\theta^k\| - \|\theta^{k+1}\|) + \sum_{k=0}^{\infty} \langle \delta^k, u^{k+1} - u^* \rangle \\ &\leq \|\theta^0\| + \bar{\eta} \sum_{k=0}^{\infty} \|\delta^k\| \leq \|\theta^0\| + \bar{\eta} \sum_{k=0}^{\infty} \epsilon^k = \|\theta^0\| + \bar{\eta}C_1. \end{aligned} \tag{81}$$

Hence,

$$\sum_{k=0}^{\infty} \|r^{k+1}\|^2 \leq \frac{2(\|\theta^0\| + \bar{\eta}C_1)}{\sigma\|(W_h - \tau M_h)^{-1}\|}, \quad \sum_{k=0}^{\infty} \|u^{k+1} - z^k\|_{M_h}^2 \leq \frac{2(\|\theta^0\| + \bar{\eta}C_1)}{\sigma}. \tag{82}$$

By substituting (82) to (79), we have

$$\begin{aligned} \sum_{k=0}^{\infty} R_h(w^{k+1}) &\leq 2 \sum_{k=0}^{\infty} \|\delta^k\|^2 + \eta \sum_{k=0}^{\infty} \|r^{k+1}\|^2 + 4\sigma^2\|M_h\| \sum_{k=0}^{\infty} \|u^{k+1} - z^k\|_{M_h}^2 \\ &\leq C := 2C_2 + \eta \frac{2(\|\theta^0\| + \bar{\eta}C_1)}{\sigma\|(W_h - \tau M_h)^{-1}\|} + 4\sigma^2\|M_h\| \frac{2(\|\theta^0\| + \bar{\eta}C_1)}{\sigma} \end{aligned} \tag{83}$$

Thus, by Lemma 3, we know (67) holds. Therefore, combining the obtained global convergence results, we complete the whole proof of the Theorem 5. \square

An Inexact Majorized Accelerated Block Coordinate Descent Method for (D_h)

In this section, we consider solving problem (P) by a duality-based approach. Thus, for the purpose of numerical implementation, we first give the finite element discretizations of (D) as follows:

$$\begin{aligned} \min_{\mu, \lambda, p \in \mathbb{R}^{N_h}} \Phi_h(\mu, \lambda, p) &:= \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|\lambda + \mu - p\|_{M_h}^2 + \langle M_h y_r, p \rangle \\ &\quad + \delta_{[-\beta, \beta]}(\lambda) + \delta_{[a, b]}^*(M_h \mu) - \frac{1}{2} \|y_d\|_{M_h}^2. \end{aligned} \tag{D_h}$$

It is clear that problem (D_h) is a convex composite minimization problem whose objective is the sum of a coupled quadratic function involving three blocks of variables and two separable nonsmooth functions involving only the first and second block, respectively. In the following sections, benefiting from the structure of (D_h) , we aim to propose an efficient and fast algorithm to solve it.

An Inexact Block Symmetric Gauss-Seidel Iteration

We first introduce the symmetric Gauss-Seidel (sGS) technique proposed recently by Li, Sun, and Toh (Li et al. 2016). It is a powerful tool to solve a convex minimization problem whose objective is the sum of a multi-block quadratic function and a nonsmooth function involving only the first block, which plays an important role in our subsequent algorithms designs for solving the PDE-constraints optimization problems.

Let $s \geq 2$ be a given integer and $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_s$ where each \mathcal{X}_i is a real finite dimensional Euclidean space. The sGS technique aims to solve the following unconstrained nonsmooth convex optimization problem approximately:

$$\min \phi(x_1) + \frac{1}{2} \langle x, \mathcal{H}x \rangle - \langle r, x \rangle, \quad (84)$$

where $x \equiv (x_1, \dots, x_s) \in \mathcal{X}$ with $x_i \in \mathcal{X}_i, i = 1, \dots, s, \phi : \mathcal{X}_1 \rightarrow (-\infty, +\infty]$ is a closed proper convex function, $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{X}$ is a given self-adjoint positive semidefinite linear operator, and $r \equiv (r_1, \dots, r_s) \in \mathcal{X}$ is a given vector.

For notational convenience, we denote the quadratic function in (84) as

$$h(x) := \frac{1}{2} \langle x, \mathcal{H}x \rangle - \langle r, x \rangle, \quad (85)$$

and the block decomposition of the operator \mathcal{H} as

$$\mathcal{H}x := \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} & \cdots & \mathcal{H}_{1s} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} & \cdots & \mathcal{H}_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}_{1s}^* & \mathcal{H}_{2s}^* & \cdots & \mathcal{H}_{ss} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_s \end{pmatrix}, \quad (86)$$

where $\mathcal{H}_{ii} : \mathcal{X}_i \rightarrow \mathcal{X}_i, i = 1, \dots, s$ are self-adjoint positive semidefinite linear operators and $\mathcal{H}_{ij} : \mathcal{X}_j \rightarrow \mathcal{X}_i, i = 1, \dots, s-1, j > i$ are linear maps whose adjoints are given by \mathcal{H}_{ij}^* . Here, we assume that $\mathcal{H}_{ii} > 0, \forall i = 1, \dots, s$. Then, we consider a splitting of \mathcal{H} :

$$\mathcal{H} = \mathcal{D} + \mathcal{U} + \mathcal{U}^*, \quad (87)$$

where

$$\mathcal{U} := \begin{pmatrix} 0 & \mathcal{H}_{12} & \cdots & \mathcal{H}_{1s} \\ & \ddots & \cdots & \mathcal{H}_{2s} \\ & & \ddots & \mathcal{H}_{(s-1)s} \\ & & & 0 \end{pmatrix}, \tag{88}$$

denotes the strict upper triangular part of \mathcal{H} and $\mathcal{D} := \text{Diag}(\mathcal{H}_{11}, \dots, \mathcal{H}_{ss}) > 0$ is the diagonal of \mathcal{H} . For later discussions, we also define the following self-adjoint positive semidefinite linear operator:

$$\text{sGS}(\mathcal{H}) := \mathcal{T} = \mathcal{U}\mathcal{D}^{-1}\mathcal{U}^*. \tag{89}$$

For any $x \in \mathcal{X}$, we define

$$x_{\leq i} := (x_1, x_2, \dots, x_i), \quad x_{\geq i} := (x_i, x_{i+1}, \dots, x_s), \quad i = 1, \dots, s,$$

with the convention $x_{\leq 0} = x_{\geq 0} = \emptyset$. Moreover, in order to solve problem (84) inexactly, we introduce the following two error tolerance vectors:

$$\delta' := (\delta'_1, \dots, \delta'_s), \quad \delta := (\delta_1, \dots, \delta_s),$$

with $\delta'_1 = \delta_1$. Define

$$\Delta(\delta', \delta) = \delta + \mathcal{U}\mathcal{D}^{-1}(\delta - \delta'). \tag{90}$$

Given $\bar{x} \in \mathcal{X}$, we consider solving the following problem:

$$x^+ := \arg \min_x \left\{ \phi(x_1) + h(x) + \frac{1}{2} \|x - \bar{x}\|_{\mathcal{T}}^2 - \langle x, \Delta(\delta', \delta) \rangle \right\}, \tag{91}$$

where $\Delta(\delta', \delta)$ could be regarded as the error term. Then, the following sGS decomposition theorem, which is established by Li et al. in (2015), shows that computing x^+ in (91) is equivalent to computing in an inexact block symmetric Gauss-Seidel-type sequential updating of the variables x_1, \dots, x_s .

Theorem 6 (Li et al. 2015, Theorem 2.1). *Assume that the self-adjoint linear operators \mathcal{H}_i are positive definite for all $i = 1, \dots, s$. Then, it holds that*

$$\mathcal{H} + \mathcal{T} = (\mathcal{D} + \mathcal{U})\mathcal{D}^{-1}(\mathcal{D} + \mathcal{U}^*) > 0. \tag{92}$$

Furthermore, given $\bar{x} \in \mathcal{X}$, for $i = s, \dots, 2$, suppose we have computed $x_i^+ \in \mathcal{X}_i$ defined as follows:

$$\begin{aligned}
x'_i &:= \arg \min_{x_i \in \mathcal{X}_i} \phi(\bar{x}_1) + h(\bar{x}_{\leq i-1}, x_i, x'_{\geq i+1}) - \langle \delta'_i, x_i \rangle \\
&= \mathcal{H}_{ii}^{-1} \left(r_i + \delta'_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* \bar{x}_j - \sum_{j=i+1}^s \mathcal{H}_{ij} x'_j \right),
\end{aligned} \tag{93}$$

then the optimal solution x^+ defined by (91) can be obtained exactly via

$$\begin{cases}
x_1^+ = \arg \min_{x_1 \in \mathcal{X}_1} \phi(x_1) + h(x_1, x'_{\geq 2}) - \langle \delta_1, x_1 \rangle, \\
x_i^+ = \arg \min_{x_i \in \mathcal{X}_i} \phi(x_1^+) + h(x_{\leq i-1}^+, x_i, x'_{\geq i+1}) - \langle \delta_i, x_i \rangle \\
= \mathcal{H}_{ii}^{-1} \left(r_i + \delta_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* x_j^+ - \sum_{j=i+1}^s \mathcal{H}_{ij} x'_j \right), \quad i = 2, \dots, s.
\end{cases} \tag{94}$$

Remark 2. (a). In (93) and (94), x'_i and x_i^+ should be regarded as inexact solutions to the corresponding minimization problems without the linear error terms $\langle \delta'_i, x_i \rangle$ and $\langle \delta_i, x_i \rangle$. Once these approximate solutions have been computed, they would generate the error vectors δ'_i and δ_i as follows:

$$\begin{aligned}
\delta'_i &= \mathcal{H}_{ii} x'_i - \left(r_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* \bar{x}_j - \sum_{j=i+1}^s \mathcal{H}_{ij} x'_j \right), \quad i = s, \dots, 2, \\
\delta_1 &\in \partial \phi(x_1^+) + \mathcal{H}_{11} x_1^+ - \left(r_1 - \sum_{j=2}^s \mathcal{H}_{1j} x'_j \right), \\
\delta_i &= \mathcal{H}_{ii} x_i^+ - \left(r_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* x_j^+ - \sum_{j=i+1}^s \mathcal{H}_{ij} x'_j \right), \quad i = 2, \dots, s.
\end{aligned}$$

With the above known error vectors, we have that x'_i and x_i^+ are the exact solutions to the minimization problems in (93) and (94), respectively.

(b). In actual implementations, assuming that for $i = s, \dots, 2$, we have computed x'_i in the backward GS sweep for solving (93), then when solving the subproblems in the forward GS sweep in (94) for $i = 2, \dots, s$, we may try to estimate x_i^+ by using x'_i , and in this case the corresponding error vector δ_i would be given by

$$\delta_i = \delta'_i + \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* (x'_j - \bar{x}_j).$$

In practice, we may accept such an approximate solution $x_i^+ = x_i'$ for $i = 2, \dots, s$, if the corresponding error vector satisfies an admissible condition such as $\|\delta_i\| \leq c\|\delta_i'\|$ for some constant $c > 1$, say $c = 10$. □

In order to estimate the error term $\Delta(\delta', \delta)$ in (90), we have the following proposition.

Proposition 5 (Li et al. 2015, Proposition 2.1). *Suppose that $\widehat{\mathcal{H}} = \mathcal{H} + \mathcal{T}$ is positive definite. Let $\xi = \|\widehat{\mathcal{H}}^{-1/2} \Delta(\delta', \delta)\|$. It holds that*

$$\xi \leq \|\mathcal{D}^{-1/2}(\delta - \delta')\| + \|\widehat{\mathcal{H}}^{-1/2} \delta'\|. \tag{95}$$

Obviously, by choosing $v = \mu$ and $w = (\lambda, p)$ and taking

$$f(v) = \delta_{[a,b]}^*(M_h \mu), \tag{96}$$

$$g(w) = \delta_{[-\beta,\beta]}(\lambda), \tag{97}$$

$$\phi(v, w) = \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|\lambda + \mu - p\|_{M_h}^2 + \langle M_h y_r, p \rangle - \frac{1}{2} \|y_d\|_{M_h}^2, \tag{98}$$

(D_h) belongs to a general class of unconstrained, multi-block convex optimization problems with coupled objective function, that is,

$$\min_{v,w} \theta(v, w) := f(v) + g(w) + \phi(v, w), \tag{99}$$

where $f : \mathcal{V} \rightarrow (-\infty, +\infty]$ and $g : \mathcal{W} \rightarrow (-\infty, +\infty]$ are two convex functions (possibly nonsmooth), $\phi : \mathcal{V} \times \mathcal{W} \rightarrow (-\infty, +\infty]$ is a smooth convex function, and \mathcal{V}, \mathcal{W} are real finite dimensional Hilbert spaces.

Inexact Majorized Accelerate Block Coordinate Descent (imABCD) Method

It is well-known that taking the inexactness of the solutions of associated subproblems into account is important for the numerical implementation. Thus, let us give a brief sketch of the inexact majorized accelerate block coordinate descent (imABCD) method which is proposed by Cui in (2016, Chapter 3) for the case ϕ being a general smooth function. To deal with the general model (99), we need some more conditions and assumptions on ϕ .

Assumption 3. *The convex function $\phi : \mathcal{V} \times \mathcal{W} \rightarrow (-\infty, +\infty]$ is continuously differentiable with Lipschitz continuous gradient.* □

Let us denote $z := (v, w) \in \mathcal{V} \times \mathcal{W}$. In Hiriart-Urruty et al. 1984, Theorem 2.3, Hiriart-Urruty and Nguyen provide a second-order mean value theorem for ϕ , which states that for any z' and z in $\mathcal{V} \times \mathcal{W}$, there exists $z'' \in [z', z]$ and a self-adjoint positive semidefinite operator $\mathcal{G} \in \partial^2\phi(z'')$ such that

$$\phi(z) = \phi(z') + \langle \nabla\phi(z'), z - z' \rangle + \frac{1}{2}\|z' - z\|_{\mathcal{G}}^2,$$

where $\partial^2\phi(z'')$ denotes the Clarke's generalized Hessian at given z'' and $[z', z]$ denotes the line segment connecting z' and z . Under Assumption 3, it is obvious that there exist two self-adjoint positive semidefinite linear operators Q and $\widehat{Q} : \mathcal{V} \times \mathcal{W} \rightarrow \mathcal{V} \times \mathcal{W}$ such that for any $z \in \mathcal{V} \times \mathcal{W}$,

$$Q \preceq \mathcal{G} \preceq \widehat{Q}, \quad \forall \mathcal{G} \in \partial^2\phi(z).$$

Thus, for any $z, z' \in \mathcal{V} \times \mathcal{W}$, it holds

$$\phi(z) \geq \phi(z') + \langle \nabla\phi(z'), z - z' \rangle + \frac{1}{2}\|z' - z\|_Q^2,$$

and

$$\phi(z) \leq \widehat{\phi}(z; z') := \phi(z') + \langle \nabla\phi(z'), z - z' \rangle + \frac{1}{2}\|z' - z\|_{\widehat{Q}}^2.$$

Furthermore, we decompose the operators Q and \widehat{Q} into the following block structures:

$$Q_z := \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12}^* & Q_{22} \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}, \quad \widehat{Q}_z := \begin{pmatrix} \widehat{Q}_{11} & \widehat{Q}_{12} \\ \widehat{Q}_{12}^* & \widehat{Q}_{22} \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}, \quad \forall z = (v, w) \in \mathcal{U} \times \mathcal{V},$$

and assume Q and \widehat{Q} satisfy the following conditions.

Assumption 4 (Cui 2016, Assumption 3.1). *There exist two self-adjoint positive semidefinite linear operators $\mathcal{D}_1 : \mathcal{U} \rightarrow \mathcal{U}$ and $\mathcal{D}_2 : \mathcal{V} \rightarrow \mathcal{V}$ such that*

$$\widehat{Q} := Q + \text{Diag}(\mathcal{D}_1, \mathcal{D}_2).$$

Furthermore, \widehat{Q} satisfies that $\widehat{Q}_{11} > 0$ and $\widehat{Q}_{22} > 0$. □

Remark 3. It is important to note that Assumption 4 is a realistic assumption in practice. For example, when ϕ is a quadratic function, we could choose $Q = \mathcal{G} = \nabla^2\phi$. If we have $Q_{11} > 0$ and $Q_{22} > 0$, then Assumption 4 holds automatically. We should point out that ϕ is a quadratic function for many problems in the practical application, such as the SDP relaxation of a binary integer nonconvex quadratic

(BIQ) programming, the SDP relaxation for computing lower bounds for quadratic assignment problems (QAPs), and so on, and one can refer to Sun et al. (2016). Fortunately, it should be noted that the function ϕ defined in (98) for our problem (D_h) is quadratic and thus we can choose $Q = \nabla^2 \phi$.

We can now present the inexact majorized ABCD algorithm for the general problem (99) as follows.

Algorithm 2 (An inexact majorized ABCD algorithm for (99))

Input: $(v^1, w^1) = (\tilde{v}^0, \tilde{w}^0) \in \text{dom}(f) \times \text{dom}(g)$. Let $\{\epsilon_k\}$ be a summable sequence of nonnegative numbers, and set $t_1 = 1, k = 1$.

Output: $(\tilde{v}^k, \tilde{w}^k)$

Iterate until convergence:

Step 1 Choose error tolerance $\delta_v^k \in \mathcal{U}, \delta_w^k \in \mathcal{V}$ such that

$$\max\{\delta_v^k, \delta_w^k\} \leq \epsilon_k.$$

Compute

$$\begin{cases} \tilde{v}^k = \arg \min_{v \in \mathcal{V}} \{f(v) + \hat{\phi}(v, w^k; v^k, w^k) - \langle \delta_v^k, v \rangle\}, \\ \tilde{w}^k = \arg \min_{w \in \mathcal{W}} \{g(w) + \hat{\phi}(\tilde{v}^k, w; v^k, w^k) - \langle \delta_w^k, w \rangle\}. \end{cases}$$

Step 2 Set $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ and $\beta_k = \frac{t_k - 1}{t_{k+1}}$, compute

$$v^{k+1} = \tilde{v}^k + \beta_k(\tilde{v}^k - \tilde{v}^{k-1}), \quad w^{k+1} = \tilde{w}^k + \beta_k(\tilde{w}^k - \tilde{w}^{k-1}).$$

Here we state the convergence result without proving. For the detailed proof, one could see Cui (2016, Chapter 3). This theorem builds a solid foundation for our subsequent proposed algorithm.

Theorem 7 (Cui 2016, Theorem 3.2). *Suppose that Assumption 4 holds and the solution set Ω of the problem (99) is non-empty. Let $z^* = (v^*, w^*) \in \Omega$. Assume that $\sum_{k=1}^{\infty} k\epsilon_k < \infty$. Then the sequence $\{\tilde{z}^k\} := \{(\tilde{v}^k, \tilde{w}^k)\}$ generated by the Algorithm 2 satisfies that*

$$\theta(\tilde{z}^k) - \theta(z^*) \leq \frac{2\|\tilde{z}^0 - z^*\|_{\mathcal{S}}^2 + c_0}{(k + 1)^2}, \quad \forall k \geq 1,$$

where c_0 is a constant number and $\mathcal{S} := \text{Diag}(\mathcal{D}_1, \mathcal{D}_2 + \mathcal{Q}_{22})$. □

A sGS-imABCD Algorithm for (D_h)

Now, we can apply Algorithm 2 to our problem (D_h) , where μ is taken as one block, and (λ, p) are taken as the other one. Let us denote $z = (\mu, \lambda, p)$. Since ϕ defined in (98) for (D_h) is quadratic, we can take

$$Q := \frac{1}{\alpha} \begin{pmatrix} M_h & M_h & -M_h \\ M_h & M_h & -M_h \\ -M_h & -M_h & M_h + \alpha K_h M_h^{-1} K_h \end{pmatrix}, \quad (100)$$

where

$$Q_{11} := \frac{1}{\alpha} M_h, \quad Q_{22} := \frac{1}{\alpha} \begin{pmatrix} M_h & -M_h \\ -M_h & M_h + \alpha K_h M_h^{-1} K_h \end{pmatrix}.$$

Additionally, we assume that there exist two self-adjoint positive semidefinite operators \mathcal{D}_1 and \mathcal{D}_2 , such that Assumption 4 holds. It implies that we should majorize $\phi(\mu, \lambda, p)$ at $z' = (\mu', \lambda', p')$ as

$$\phi(z) \leq \hat{\phi}(z; z') = \phi(z) + \frac{1}{2} \|\mu - \mu'\|_{\mathcal{D}_1}^2 + \frac{1}{2} \left\| \begin{pmatrix} \lambda \\ p \end{pmatrix} - \begin{pmatrix} \lambda' \\ p' \end{pmatrix} \right\|_{\mathcal{D}_2}^2. \quad (101)$$

Thus, the framework of imABCD for (D_h) is given below.

Algorithm 3 (imABCD algorithm for (D_h))

Input: $(\mu^1, \lambda^1, p^1) = (\tilde{\mu}^0, \tilde{\lambda}^0, \tilde{p}^0) \in \text{dom}(\delta_{[a,b]}^*) \times [-\beta, \beta] \times \mathbb{R}^{N_h}$. Set $k = 1, t_1 = 1$.

Output: $(\tilde{\mu}^k, \tilde{\lambda}^k, \tilde{p}^k)$

Iterate until convergence

Step 1 Compute

$$\tilde{\mu}^k = \arg \min_{\delta_{[a,b]}^*} (M_h \mu) + \phi(\mu, \lambda^k, p^k) + \frac{1}{2} \|\mu - \mu^k\|_{\mathcal{D}_1}^2 - \langle \delta_{\mu}^k, \mu \rangle,$$

$$(\tilde{\lambda}^k, \tilde{p}^k) = \arg \min_{\delta_{[-\beta, \beta]}} \delta(\tilde{\mu}^k, \lambda, p) + \frac{1}{2} \left\| \begin{pmatrix} \lambda \\ p \end{pmatrix} - \begin{pmatrix} \lambda^k \\ p^k \end{pmatrix} \right\|_{\mathcal{D}_2}^2 - \langle \delta_{\lambda}^k, \lambda \rangle - \langle \delta_p^k, p \rangle.$$

Step 2 Set $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ and $\beta_k = \frac{t_k - 1}{t_{k+1}}$, compute

$$\mu^{k+1} = \tilde{\mu}^k + \beta_k (\tilde{\mu}^k - \tilde{\mu}^{k-1}), \quad p^{k+1} = \tilde{p}^k + \beta_k (\tilde{p}^k - \tilde{p}^{k-1}), \quad \lambda^{k+1} = \tilde{\lambda}^k + \beta_k (\tilde{\lambda}^k - \tilde{\lambda}^{k-1}).$$

Next, another key issue that should be considered is how to choose the operators \mathcal{D}_1 and \mathcal{D}_2 . As we know, choosing the appropriate and effective operators \mathcal{D}_1 and \mathcal{D}_2 is an important thing from the perspective of both theory analysis and numerical implementation. Note that for numerical efficiency, the general principle is that both \mathcal{D}_1 and \mathcal{D}_2 should be chosen as small as possible such that $\tilde{\mu}^k$ and $(\tilde{\lambda}^k, \tilde{p}^k)$ could take larger step-lengths while the corresponding subproblems still could be solved relatively easily.

First, for the proximal term $\frac{1}{2}\|\mu - \mu^k\|_{\mathcal{D}_1}^2$, in order to make the subproblem of the block μ having an analytical solution, and from Proposition (1), we choose

$$\mathcal{D}_1 := \frac{1}{\alpha}c_n M_h W_h^{-1} M_h - \frac{1}{\alpha}M_h, \quad \text{where } c_n = \begin{cases} 4 & \text{if } n = 2, \\ 5 & \text{if } n = 3. \end{cases}$$

Next, we will focus on how to choose the operator \mathcal{D}_2 . If we ignore the proximal term $\frac{1}{2}\left\|\begin{pmatrix} \lambda \\ p \end{pmatrix} - \begin{pmatrix} \lambda^k \\ p^k \end{pmatrix}\right\|_{\mathcal{D}_2}^2$ and the error terms, it is obvious that the subproblem of the block (λ, p) belongs to the form (84), which can be rewritten as

$$\min_{\delta_{[-\beta, \beta]}}(\lambda) + \frac{1}{2}\left\langle \begin{pmatrix} \lambda \\ p \end{pmatrix}, \mathcal{H} \begin{pmatrix} \lambda \\ p \end{pmatrix} \right\rangle - \left\langle r, \begin{pmatrix} \lambda \\ p \end{pmatrix} \right\rangle, \tag{102}$$

where $\mathcal{H} = \mathcal{Q}_{22} = \frac{1}{\alpha} \begin{pmatrix} M_h & -M_h \\ -M_h & M_h + \alpha K_h M_h^{-1} K_h \end{pmatrix}$ and

$r = \begin{pmatrix} \frac{1}{\alpha}M_h \tilde{\mu}^k \\ M_h y_r - K_h y_d - \frac{1}{\alpha}M_h \tilde{\mu}^k \end{pmatrix}$. Since the objective function of (102) is the sum of a two-block quadratic function and a nonsmooth function involving only the first block, thus the inexact sGS technique, which is introduced in Section , can be used to solve (102) . To achieve our goal, we choose

$$\tilde{\mathcal{D}}_2 = \text{sGS}(\mathcal{Q}_{22}) = \frac{1}{\alpha} \begin{pmatrix} M_h (M_h + \alpha K_h M_h^{-1} K_h)^{-1} M_h & 0 \\ 0 & 0 \end{pmatrix}.$$

Then according to Theorem 6, we can solve the (λ, p) -subproblem by the following procedure:

$$\begin{cases} \hat{p}^k = \arg \min \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|p - \lambda^k - \tilde{\mu}^k + \alpha y_r\|_{M_h}^2 - \langle \hat{\delta}_p^k, p \rangle, \\ \tilde{\lambda}^k = \arg \min \frac{1}{2\alpha} \|\lambda - (\hat{p}^k - \tilde{\mu}^k)\|_{M_h}^2 + \delta_{[-\beta, \beta]}(\lambda), \\ \tilde{p}^k = \arg \min \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|p - \tilde{\lambda}^k - \tilde{\mu}^k + \alpha y_r\|_{M_h}^2 - \langle \delta_p^k, p \rangle. \end{cases} \quad (103)$$

However, it is easy to see that the λ -subproblem is coupled about the variable λ since the mass matrix M_h is not diagonal; thus, there is no closed-form solution for λ . To overcome this difficulty, we can take advantage of the relationship between the mass matrix M_h and the lumped mass matrix W_h and add a proximal term $\frac{1}{2\alpha} \|\lambda - \lambda^k\|_{W_h - M_h}^2$ to the λ -subproblem. Fortunately, we have

$$\text{sGS}(\mathcal{Q}_{22}) = \text{sGS} \left(\mathcal{Q}_{22} + \frac{1}{\alpha} \begin{bmatrix} W_h - M_h & 0 \\ 0 & 0 \end{bmatrix} \right),$$

which implies that the proximal term $\frac{1}{2\alpha} \|\lambda - \lambda^k\|_{W_h - M_h}^2$ has no influence on the sGS technique. Thus, we can choose \mathcal{D}_2 as follows:

$$\mathcal{D}_2 = \text{sGS}(\mathcal{Q}_{22}) + \frac{1}{\alpha} \begin{pmatrix} W_h - M_h & 0 \\ 0 & 0 \end{pmatrix}.$$

Based on the choice of \mathcal{D}_1 and \mathcal{D}_2 , we get the majorized Hessian matrix $\hat{\mathcal{Q}}$ as follows:

$$\hat{\mathcal{Q}} = \mathcal{Q} + \frac{1}{\alpha} \begin{pmatrix} c_n M_h W_h^{-1} M_h - M_h & 0 & 0 \\ 0 & M_h (M_h + \alpha K_h M_h^{-1} K_h)^{-1} M_h + W_h - M_h & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (104)$$

Then, according to the choice of \mathcal{D}_1 and \mathcal{D}_2 , we give the detailed framework of our inexact sGS based majorized ABCD method (called sGS-imABCD) for (\mathbf{D}_h) as follows.

Based on Theorem 7, we can show our Algorithm 4 (sGS-imABCD) also has the following $O(1/k^2)$ iteration complexity.

Theorem 8. Assume that $\sum_{i=k}^{\infty} k\epsilon_k < \infty$. Let $\{\tilde{z}^k\} := \{(\tilde{\mu}^k, \tilde{\lambda}^k, \tilde{p}^k)\}$ be the sequence generated by Algorithm 4. Then we have

$$\Phi_h(\tilde{z}^k) - \Phi_h(z^*) \leq \frac{2\|\tilde{z}^0 - z^*\|_S^2 + c_0}{(k + 1)^2}, \quad \forall k \geq 1,$$

where c_0 is a constant number, $S := \text{Diag}(\mathcal{D}_1, \mathcal{D}_2 + \mathcal{Q}_{22})$, and $\Phi_h(\cdot)$ is the objective function of the dual problem (D_h) . □

Algorithm 4 (sGS-imABCD algorithm for (D_h))

Input: $(\mu^1, \lambda^1, p^1) = (\tilde{\mu}^0, \tilde{\lambda}^0, \tilde{p}^0) \in \text{dom}(\delta_{[a,b]}^*) \times [-\beta, \beta] \times \mathbb{R}^{N_h}$. Let $\{\epsilon_k\}$ be a nonincreasing sequence of nonnegative numbers such that $\sum_{k=1}^{\infty} k\epsilon_k < \infty$. Set $k = 1, t_1 = 1$.

Output: $(\tilde{\mu}^k, \tilde{\lambda}^k, \tilde{p}^k)$

Iterate until convergence

Step 1 Choose error tolerance $\delta_\mu^k, \hat{\delta}_p^k, \delta_p^k$ such that

$$\max\{\|\delta_\mu^k\|, \|\hat{\delta}_p^k\|, \|\delta_p^k\|\} \leq \epsilon_k.$$

Compute

$$\tilde{\mu}^k = \arg \min \frac{1}{2\alpha} \|\mu - (p^k - \lambda^k)\|_{M_h}^2 + \delta_{[a,b]}^*(M_h \mu) + \frac{1}{2} \|\mu - \mu^k\|_{\mathcal{D}_1}^2 - \langle \delta_\mu^k, \mu \rangle,$$

$$\hat{p}^k = \arg \min \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|p - \lambda^k - \tilde{\mu}^k + \alpha y_r\|_{M_h}^2 - \langle \hat{\delta}_p^k, p \rangle,$$

$$\tilde{\lambda}^k = \arg \min \frac{1}{2\alpha} \|\lambda - (\hat{p}^k - \tilde{\mu}^k)\|_{M_h}^2 + \delta_{[-\beta, \beta]}(\lambda) + \frac{1}{2\alpha} \|\lambda - \lambda^k\|_{W_h - M_h}^2,$$

$$\tilde{p}^k = \arg \min \frac{1}{2} \|K_h p - M_h y_d\|_{M_h^{-1}}^2 + \frac{1}{2\alpha} \|p - \tilde{\lambda}^k - \tilde{\mu}^k + \alpha y_r\|_{M_h}^2 - \langle \delta_p^k, p \rangle.$$

Step 2 Set $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$ and $\beta_k = \frac{t_k - 1}{t_{k+1}}$, compute

$$\mu^{k+1} = \tilde{\mu}^k + \beta_k(\tilde{\mu}^k - \tilde{\mu}^{k-1}), \quad p^{k+1} = \tilde{p}^k + \beta_k(\tilde{p}^k - \tilde{p}^{k-1}), \quad \lambda^{k+1} = \tilde{\lambda}^k + \beta_k(\tilde{\lambda}^k - \tilde{\lambda}^{k-1}).$$

Proof. By Proposition 1, we know that $c_n M_h W_h^{-1} M_h - M_h > 0$, $M_h(M_h + \alpha K_h M_h^{-1} K_h)^{-1} M_h > 0$, $W_h - M_h > 0$. Moreover, since stiffness and mass matrices are symmetric positive definite matrices, it is noticed that Assumption 4 is valid for our \hat{Q} which is defined in (104). Thus, according to Theorem 7, we can establish the convergence of Algorithm 4. □

Remark 4. Let $\tau_h = 2\|\tilde{z}^0 - z^*\|_S^2 + c_0$. It is obvious that τ_h is independent of the parameter β , whereas it depends on the parameter α and will increase with the decrease of α .

Numerical Results

In this section, we will first use Example 1 and Example 2 to evaluate the numerical behavior of the ihADMM and use Example 3 and Example 4 to evaluate the numerical behavior of the sGS-imABCD.

Algorithmic Details

We begin by describing the algorithmic details which are common to all examples.

Discretization. The discretization was carried out by using piecewise linear and continuous finite elements. The assembly of mass and the stiffness matrices, as well as the lump mass matrix, was left to the iFEM software package. To present the finite element error estimate results, it is convenient to introduce the experimental order of convergence (EOC), which for some positive error functional $E(h)$ with $h > 0$ is defined as follows: Given two grid sizes $h_1 \neq h_2$, let

$$\text{EOC} := \frac{\log E(h_1) - \log E(h_2)}{\log h_1 - \log h_2}. \quad (105)$$

It follows from this definition that if $E(h) = O(h^\gamma)$, then $\text{EOC} \approx \gamma$. The error functional $E(\cdot)$ investigated in the present section is given by $E_2(h) := \|u - u_h\|_{L^2(\Omega)}$.

Initialization. For all numerical examples, we choose $u = 0$ as initialization u^0 for all algorithms.

In Example 1 and Example 2, for comparison with ihADMM, we will also show the numerical results obtained by the classical ADMM and the APG algorithm, and the PDAS with line search. For the classical ADMM and our ihADMM, the penalty parameter σ was chosen as $\sigma = 0.1\alpha$. About the step-length τ , we choose $\tau = 1.618$ for the classical ADMM, and $\tau = 1$ for our ihADMM. For the PDAS method, the parameter in the active set strategy was chosen as $c = 1$. For the APG method, we estimate an approximation for the Lipschitz constant L with a backtracking method. In the numerical experiments, we measure the accuracy of an approximate optimal solution by using the corresponding K-K-T residual error for each algorithm. For the purpose of showing the efficiency of our ihADMM, we report the numerical results obtained by running the classical ADMM and the APG method to compare with the results obtained by our ihADMM. In this case, we terminate all the algorithms when $\eta < 10^{-6}$ with the maximum number of iterations set at 500.

In Example 3 and Example 4, for comparison with sGS-imABCD, we will also show the numerical results obtained by the ihADMM and APG methods for $(\overline{\text{DP}}_h)$. For the ihADMM method, the step-length τ for Lagrangian multipliers λ was chosen as $\tau = 1$, and the penalty parameter σ was chosen as $\sigma = 0.1\alpha$. For

the APG method, we estimate an approximation to the Lipschitz constant L with a backtracking method with $\eta = 1.4$ and $L^0 = 10^{-8}$. In the numerical experiments, we terminate all the algorithms when the corresponding relative residual $\eta < 10^{-7}$.

Examples

Example 1.

$$\left\{ \begin{array}{ll} \min_{(y,u) \in H_0^1(\Omega) \times L^2(\Omega)} & J(y, u) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2 + \beta \|u\|_{L^1(\Omega)} \\ \text{s.t.} & -\Delta y = u + y_c \quad \text{in } \Omega, \\ & y = 0 \quad \text{on } \partial\Omega, \\ & u \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e on } \Omega\}. \end{array} \right.$$

Here, we consider the problem with control $u \in L^2(\Omega)$ on the unit square $\Omega = (0, 1)^2$ with $\alpha = 0.5$, $\beta = 0.5$, $a = -0.5$, and $b = 0.5$. It is a constructed problem; thus, we set $y^* = \sin(\pi x_1) \sin(\pi x_2)$ and $p^* = 2\beta \sin(2\pi x_1) \exp(0.5x_1) \sin(4\pi x_2)$. Then through $u^* = \Pi_{U_{ad}} \left(\frac{1}{\alpha} \text{soft}(-p^*, \beta) \right)$, $y_c = y^* - \mathcal{S}u^*$, and $y_d = \mathcal{S}^{-*} p^* + y^*$, we can construct the example for which we know the exact solution.

The error of the control u w.r.t the L^2 -norm and the EOC for control are presented in Table 1. They also confirm that indeed the convergence rate is of order $O(h)$. Numerical results for the accuracy of solution, number of iterations, and CPU time obtained by our ihADMM, classical ADMM, and APG methods are shown in Table 1. As a result from Table 1, we can see that our proposed ihADMM method is an efficient algorithm to solve problem (\overline{DP}_h) to medium accuracy. Moreover, it is obvious that our ihADMM outperforms the classical ADMM and the APG method in terms of CPU time, especially when the discretization is in a fine level. It is worth noting that although the APG method requires less number of iterations when the termination condition is satisfied, the APG method spends much time on backtracking step with the aim of finding an appropriate approximation for the Lipschitz constant. This is the reason that our ihADMM has better performance than the APG method in actual numerical implementation. Furthermore, the numerical results in terms of iteration numbers illustrate the mesh-independent performance of the ihADMM and the APG method, except for the classical ADMM.

Table 1 Example 1: The convergence behavior of our ihADMM, classical ADMM, and APG for (DP_h) . In the table, #dofs stands for the number of degrees of freedom for the control variable on each grid level

h	#dofs	E_2	EOC	Index	ihADMM	Classical ADMM	APG
2^{-3}	49	0.2925	-	iter	27	32	13
				residual η	7.15e-07	7.55e-07	6.88e-07
				CPU time/s	0.19	0.23	0.18
2^{-4}	225	0.1127	1.3759	iter	31	44	13
				residual η	9.77e-07	9.91e-07	8.23e-07
				CPU times/s	0.37	0.66	0.32
2^{-5}	961	0.0457	1.3390	iter	31	58	12
				residual η	7.41e-07	8.11e-07	7.58e-07
				CPU time/s	1.02	2.32	1.00
2^{-6}	3969	0.0161	1.3944	iter	32	76	14
				residual η	7.26e-07	8.10e-07	7.88e-07
				CPU time/s	4.18	9.12	4.25
2^{-7}	16129	0.0058	1.4132	iter	31	94	14
				residual η	5.33e-07	7.85e-07	4.45e-07
				CPU time/s	17.72	65.82	26.25
2^{-8}	65025	0.0019	1.4503	iter	32	127	13
				residual η	6.88e-07	8.93e-07	7.47e-07
				CPU time/s	70.45	312.65	80.81
2^{-9}	261121	0.0007	1.4542	iter	31	255	13
				residual η	7.43e-07	7.96e-07	6.33e-07
				CPU time/s	525.28	4845.31	620.55

Example 2.

$$\left\{ \begin{array}{l} \min_{(y,u) \in Y \times U} J(y, u) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2 + \beta \|u\|_{L^1(\Omega)} \\ \text{s.t.} \quad -\Delta y = u, \quad \text{in } \Omega = (0, 1) \times (0, 1) \\ \quad \quad y = 0, \quad \text{on } \partial\Omega \\ \quad \quad u \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e on } \Omega\}, \end{array} \right.$$

where the desired state $y_d = \frac{1}{6} \sin(2\pi x) \exp(2x) \sin(2\pi y)$ and the parameters $\alpha = 10^{-5}$, $\beta = 10^{-3}$, $a = -30$, and $b = 30$. In addition, the exact solutions of the problem are unknown. Instead, we use the numerical solutions computed on a grid with $h^* = 2^{-10}$ as reference solutions.

The error of the control u w.r.t the L^2 norm with respect to the solution on the finest grid ($h^* = 2^{-10}$) and the experimental order of convergence (EOC) for control are presented in Table 2. They confirm the linear rate of convergence w.r.t. h .

Numerical results for the accuracy of solution, number of iterations, and CPU time obtained by our ihADMM, classical ADMM, and APG methods are also shown in Table 2. Experiment results show that the ADMM has evident advantage over the classical ADMM and the APG method in computing time. Furthermore, the numerical results in terms of iteration numbers also illustrate the mesh-independent performance of our ihADMM. These results demonstrate that our ihADMM is highly efficient in obtaining an approximate solution with moderate accuracy.

Table 2 Example 2: The convergence behavior of ihADMM, classical ADMM, and APG for (DP_h)

h	#dofs	E_2	EOC	Index	ihADMM	Classical ADMM	APG
2^{-3}	49	6.6122	-	iter	40	48	18
				residual η	8.22e-07	8.65e-07	7.96e-07
				CPU time/s	0.30	0.51	0.24
2^{-4}	225	2.6314	1.3293	iter	41	56	18
				residual η	7.22e-07	8.01e-07	7.58e-07
				CPU times/s	0.45	0.71	0.44
2^{-5}	961	1.2825	1.1831	iter	40	69	19
				residual η	8.12e-07	8.01e-07	7.90e-07
				CPU time/s	1.60	3.05	1.58
2^{-6}	3969	0.7514	1.0458	iter	42	85	18
				residual η	6.11e-07	7.80e-07	6.45e-07
				CPU time/s	7.25	14.62	7.45
2^{-7}	16129	0.2930	1.1240	iter	40	108	18
				residual η	6.35e-07	7.11e-07	5.62e-07
				CPU time/s	33.85	101.36	34.39
2^{-8}	65025	0.1357	1.1213	iter	41	132	19
				residual η	7.55e-07	7.83e-07	7.57e-07
				CPU time/s	158.62	508.65	165.75
2^{-9}	261121	0.0958	1.0181	iter	42	278	18
				residual η	5.25e-07	5.56e-07	4.85e-07
				CPU time/s	1781.98	11788.52	1860.11
2^{-10}	1046529	-	-	iter	41	500	19
				residual η	8.78e-07	Error	8.47e-07
				CPU time/s	42033.79	Error	44131.27

Example 3.

$$\left\{ \begin{array}{l} \min_{(y,u) \in H_0^1(\Omega) \times L^2(\Omega)} J(y, u) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2 + \beta \|u\|_{L^1(\Omega)} \\ \text{s.t.} \quad -\Delta y = u + y_r \quad \text{in } \Omega, \\ \quad \quad \quad y = 0 \quad \text{on } \partial\Omega, \\ \quad \quad \quad u \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e on } \Omega\}. \end{array} \right.$$

Here, we consider the problem with control $u \in L^2(\Omega)$ on the unit square $\Omega = (0, 1)^2$ with $\alpha = 0.5$, $\beta = 0.5$, $a = -0.5$, and $b = 0.5$. It is a constructed problem; thus, we set $y^* = \sin(2\pi x_1) \exp(0.5x_1) \sin(4\pi x_2)$ and $p^* = 2\beta \sin(2\pi x_1) \exp(0.5x_1) \sin(4\pi x_2)$.

The error of the control u w.r.t the L^2 norm and the experimental order of convergence (EOC) for control are presented in Tables 3 and 5. They also confirm that indeed the convergence rate is of order $O(h)$. Comparing the error results from Tables 3 and 5, it is obvious to see that solving the dual problem (D_h) could get better error results than that from solving (\overline{DP}_h) .

Numerical results for the accuracy of solution, number of iterations, and CPU time obtained by our proposed sGS-imABCD method for (D_h) are also shown in Table 3. As a result we obtain from Table 3, one can see that our proposed sGS-imABCD method is an efficient algorithm to solve problem (D_h) to high accuracy. It should be pointed out that iter. \tilde{p} -block denotes the iterations of \tilde{p} in Table 3. It is clear that p -subproblem almost always not be computed twice, which demonstrates the efficiency of our strategy to predict the solution of \tilde{p} -subproblem. Furthermore, the numerical results in terms of iteration numbers illustrate the mesh-independent performance of our proposed sGS-imABCD method. Additionally, in Table 4, we list the numbers of iteration steps and the relative residual errors of PMHSS-preconditioned GMRES method for the \hat{p} -subproblem on mesh $h = 2^{-7}$

Table 3 Example 3: The performance of sGS-imABCD for (D_h) . In the table, #dofs stands for the number of degrees of freedom for the control variable on each grid level

h	#dofs	iter.sGS-imABCD	iter. \tilde{p} -block	residual η	CPU time/s	E_2	EOC
2^{-3}	49	13	4	6.60e-08	0.14	0.1784	-
2^{-4}	225	13	4	6.32e-08	0.20	0.0967	0.8834
2^{-5}	961	12	3	7.38e-08	0.33	0.0399	1.0803
2^{-6}	3969	13	3	9.78e-08	2.04	0.0155	1.1749
2^{-7}	16129	12	3	6.66e-08	8.25	0.0052	1.2754
2^{-8}	65025	10	3	7.05e-08	52.15	0.0017	1.3388
2^{-9}	261121	9	2	5.19e-08	312.82	0.0006	1.3617

Table 4 Example 3: The convergence behavior of GMRES for $\hat{\rho}$ -block subproblem

h	iter.sGS-imABCD	iter.GMRES of $\hat{\rho}$ -block	Relative residual error of GMRES
2^{-7}	1	8	1.30e-07
	2	4	1.07e-07
	3	4	5.26e-08
	4	4	1.56e-08
	5	4	2.05e-09
	6	4	1.58e-09
	7	4	1.23e-09
	8	4	1.29e-10
	9	2	1.16e-10
	10	2	1.07e-10
	11	2	5.98e-11
	12	2	1.30e-11
2^{-8}	1	8	6.31e-08
	2	4	2.18e-08
	3	4	8.43e-09
	4	4	3.18e-09
	5	4	1.07e-09
	6	4	5.53e-10
	7	4	5.25e-11
	8	4	5.90e-12
	9	2	4.86e-12
	10	2	4.18e-12

and $h = 2^{-8}$. From Table 4, we can see that the number of iteration steps of the PMHSS-preconditioned GMRES method is roughly independent of the mesh size h .

As a comparison, numerical results obtained by the our proposed sGS-imABCD method for (D_h) and the iwADMM and APG methods for (\overline{DP}_h) are shown in Table 5. As a result from Table 5, it can be observed that our sGS-imABCD is faster and more efficient than the iwADMM and APG methods in terms of the iterations and CPU times.

At last, in order to show the robustness of our proposed sGS-imABCD method with respect to the parameters α and β , we also test the same problem with different values of α and β on mesh $h = 2^{-8}$. The results are presented in Table 6. From Table 6, it is obvious to see that our method could solve problem (D_h) to high accuracy for all tested values of α and β within 50 iterations. More importantly, from the results, we can see that when α is fixed, the number of iteration steps of the sGS-imABCD method remains nearly constant for β ranging from 0.005 to 1. However, for a fixed β , as α increases from 0.005 to 0.5, the number of iteration steps of the sGS-imABCD method changes drastically. These observations indicate that the sGS-imABCD method shows the β -independent convergence property, whereas it does not have the same convergence property with respect to the parameter α .

Table 5 Example 3: The convergence behavior of sGS-imABCD for (D_h) , ihADMM, and APG for (DP_h) . In the table, #dofs stands for the number of degrees of freedom for the control variable on each grid level. $E_2 = \min\{E_2(sGS - imABCD), E_2(ihADMM), E_2(APG)\}$

h	#dofs	E_2	EOC	Index of performance	sGS-imABCD	ihADMM	APG
2^{-3}	49	0.2925	-	iter	13	32	16
				residual η	6.25e-08	6.33e-08	3.51e-08
				CPU time/s	0.16	0.23	0.22
2^{-4}	225	0.1127	1.3759	iter	12	36	18
				residual η	6.34e-08	8.91e-08	7.23e-08
				CPU times/s	0.24	0.44	0.45
2^{-5}	961	0.0457	1.3390	iter	13	40	16
				residual η	7.10e-08	7.42e-08	8.88e-08
				CPU time/s	0.47	1.17	2.98
2^{-6}	3969	0.0161	1.3944	iter	14	44	16
				residual η	4.05e-08	9.10e-08	6.60e-08
				CPU time/s	2.62	6.04	4.86
2^{-7}	16129	0.0058	1.4132	iter	12	50	16
				residual η	6.43e-08	9.80e-08	8.45e-08
				CPU time/s	10.22	29.53	30.63
2^{-8}	65025	0.0019	1.4503	iter	10	53	17
				residual η	7.05e-08	8.93e-08	8.88e-08
				CPU time/s	60.45	160.24	92.60
2^{-9}	261121	0.0007	1.4542	iter	10	54	18
				residual η	5.21e-08	7.96e-08	3.24e-08
				CPU time/s	395.78	915.71	859.22

It should be pointed out that the numerical results are also consistent with the theoretical conclusion based on Theorem 8.

Example 4.

$$\left\{ \begin{array}{l} \min_{(y,u) \in Y \times U} J(y, u) = \frac{1}{2} \|y - y_d\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2 + \beta \|u\|_{L^1(\Omega)} \\ \text{s.t.} \quad -\Delta y = u \quad \text{in } \Omega = (0, 1) \times (0, 1), \\ \quad \quad y = 0 \quad \text{on } \partial\Omega, \\ \quad \quad u \in U_{ad} = \{v(x) | a \leq v(x) \leq b, \text{ a.e on } \Omega\}, \end{array} \right.$$

where the desired state $y_d = \frac{1}{6} \sin(2\pi x) \exp(2x) \sin(2\pi y)$ and the parameters $\alpha = 10^{-5}$, $\beta = 10^{-3}$, $a = -30$, and $b = 30$. In addition, the exact solution of

Table 6 Example 3: The performance of sGS-imABCD for (D_h) with different values of α and β

h	α	β	iter.sGS-imABCD	residual error η about K-K-T
2^{-8}	0.005	0.005	49	7.59e-08
		0.05	48	8.86e-08
		0.5	46	6.76e-08
		1	48	5.49e-08
	0.05	0.005	23	8.74e-08
		0.05	25	7.26e-08
		0.5	22	5.77e-08
		1	23	7.63e-08
	0.5	0.005	12	6.51e-08
		0.05	11	8.80e-08
		0.5	10	7.05e-08
		1	12	8.53e-08

Table 7 Example 4: The performance of sGS-imABCD for (D_h) . In the table, #dofs stands for the number of degrees of freedom for the control variable on each grid level

h	#dofs	iter. sGS-imABCD	No. \tilde{p} -block	residual η	CPU time/s	E_2	EOC
2^{-3}	49	37	12	8.67e-08	0.64	5.5408	–
2^{-4}	225	30	10	7.32e-08	0.65	2.4426	1.1817
2^{-5}	961	22	8	8.38e-08	0.73	1.1504	1.1340
2^{-6}	3969	22	7	6.83e-08	4.65	0.4380	1.2203
2^{-7}	16129	16	5	6.46e-08	16.60	0.1774	1.2413
2^{-8}	65025	15	3	6.36e-08	105.70	0.1309	1.0807
2^{-9}	261121	15	3	5.65e-08	1158.62	0.0406	1.1821
2^{-10}	1046529	16	3	4.50e-08	24008.07	–	–

the problem is unknown. In this case, using a numerical solution as the reference solution is a common method. For more details, one can see Hinze et al. (2009). In our practice implementation, we use the numerical solution computed on a grid with $h^* = 2^{-10}$ as the reference solution. It should be emphasized that choosing the solution that computed on mesh $h^* = 2^{-10}$ is reliable. As shown below, when $h^* = 2^{-10}$, the scale of data is 1046529.

In Table 7, we report the numerical results obtained by our proposed sGS-imABCD method for solving (D_h) . As a result, one can see that our proposed sGS-imABCD method is an efficient algorithm to solve problem (D_h) to high accuracy. In addition, the errors of the control u with respect to the solution on the finest grid ($h^* = 2^{-10}$) and the results of EOC for control are also presented in Table 7, which confirm the error estimate result as shown in Theorem 1. For the sake of comparison, in Table 9, we report the numerical results obtained by

Table 8 Example 4: The convergence behavior of GMRES for \hat{p} -block subproblem

h	iter.sGS-imABCD	iter.GMRES of \hat{p} -block	Relative residual error of GMRES
2^{-7}	1	7	1.54e-04
	2	7	1.12e-05
	3	8	7.25e-06
	4	8	3.95e-06
	5	8	3.85e-06
	6	8	2.66e-06
	7	8	3.33e-06
	8	8	2.60e-06
	9	8	1.86e-06
	10	8	1.15e-06
	11	8	1.28e-06
	12	7	8.68e-07
	13	7	9.26e-07
	14	7	5.17e-07
	15	7	7.76e-07
	16	7	7.39e-07
2^{-8}	1	7	1.50e-04
	2	7	1.11e-05
	3	8	7.23e-06
	4	8	9.61e-06
	5	9	5.56e-06
	6	10	7.37e-07
	7	8	3.98e-06
	8	8	2.34e-06
	9	8	1.96e-06
	10	8	1.15e-06
	11	8	1.27e-06
	12	7	8.36e-07
	13	7	8.16e-07
	14	7	4.38e-07
	15	7	7.61e-07

sGS-imABCD method for solving (D_h) and iwADMM and APG methods for (\overline{DP}_h) . Comparing the error results from Tables 7 and 9, we can see that directly solving (D_h) can get better error results than that from solving (D_h) and (\overline{DP}_h) . Obviously, this conclusion shows the efficiency of our dual-based approach which can avoid the additional error caused by the approximation of L^1 -norm. Furthermore, from Table 7, the numerical results in terms of iteration numbers illustrate the mesh-independent performance of our proposed sGS-imABCD method.

In addition, in Table 8, numbers of iteration steps and the relative residual errors of PMHSS-preconditioned GMRES method for the \hat{p} -subproblem on mesh $h = 2^{-7}$

Table 9 Example 4: The convergence behavior of sGS-imABCD, ihADMM, and APG for (\overline{DP}_h)

h	#dofs	E_2	EOC	Index of performance	sGS-imABCD	ihADMM	APG
2^{-3}	49	6.6122	-	iter	40	56	44
				residual η	6.06e-08	8.36e-08	9.92e-08
				CPU time/s	0.72	0.42	0.60
2^{-4}	225	2.6314	1.3293	iter	16	55	39
				residual η	9.94e-08	9.14e-08	9.74e-08
				CPU times/s	0.48	0.62	1.03
2^{-5}	961	1.2825	1.1831	iter	21	51	29
				residual η	5.36e-08	8.59e-08	8.31e-06
				CPU time/s	0.99	1.707	3.84
2^{-6}	3969	0.7514	1.0458	iter	22	46	29
				residual η	9.91e-08	6.83e-08	9.38e-08
				CPU time/s	4.95	8.34	11.94
2^{-7}	16129	0.29304	1.1240	iter	20	46	24
				residual η	9.89e-08	5.85e-08	9.36e-08
				CPU time/s	20.83	38.93	45.85
2^{-8}	65025	0.1357	1.1213	iter	20	48	20
				residual η	4.99e-08	8.39e-08	9.05e-08
				CPU time/s	143.88	219.27	181.11
2^{-9}	261121	0.0958	1.0181	iter	18	50	20
				residual η	9.05e-08	7.04e-08	8.84e-08
				CPU time/s	1272.25	2227.48	1959.11

and $h = 2^{-8}$ are presented, which shows that the PMHSS-preconditioned GMRES method is roughly independent of the mesh size h .

As a result from Table 9, it can be also observed that our sGS-imABCD is faster and more efficient than the iwADMM and APG methods in terms of the iteration numbers and CPU times. The numerical performance of our proposed sGS-imABCD method clearly demonstrates the importance of our method.

Finally, to show the influence of the parameters α and β on our proposed sGS-imABCD method, we also test Example 4 with different values of α and β on mesh $h = 2^{-8}$. The results are presented in Table 10. From Table 10, it is obvious to see that our proposed sGS-imABCD method is independent of the parameter β . However, its convergence rate depends on α . It also confirms the convergence results of Theorem 8.

Table 10 Example 4: The performance of sGS-imABCD for (D_h) with different values of α and β

h	α	β	iter.sGS-imABCD	residual error η about K-K-T
2^{-8}	10^{-6}	0.0005	26	8.37e-08
		0.001	27	8.40e-08
		0.005	26	9.77e-08
		0.008	28	2.47e-08
	10^{-5}	0.0005	13	5.44e-08
		0.001	15	6.36e-08
		0.005	14	8.60e-08
		0.008	13	8.17e-08
	10^{-4}	0.0005	5	9.84e-08
		0.001	4	3.71e-08
		0.005	5	9.23e-08
		0.008	5	5.22e-08

Conclusion

In this chapter, elliptic PDE-constrained optimal control problems with L^1 -control cost (L^1 -EOCP) are considered. By taking advantage of inherent structures of the problem, we introduce an inexact heterogeneous ADMM (ihADMM) to solve discretized problems. Furthermore, theoretical results on the global convergence as well as the iteration complexity results $o(1/k)$ for ihADMM were given. Instead of solving the primal problem, we introduce a duality-based approach. By taking advantage of the structure of dual problem, and combining the inexact majorized ABCD (imABCD) method and the recent advances in the inexact symmetric Gauss-Seidel (sGS) technique, we introduce the sGS-imABCD method to solve the dual problem.

References

- Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**, 183–202 (2009)
- Bergounioux, M., Ito, K., Kunisch, K.: Primal-dual strategy for constrained optimal control problems, *SIAM J. Control Optim.* **37**, 1176–1194 (1999)
- Blumensath, T., Davies, M.E.: Iterative Thresholding for Sparse Approximations. *J. Fourier Anal. Appl.* **14**, 629–654 (2008)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends® Mach. Learn.* **3**, 1–122 (2011)

- Carstensen, C.: Quasi-interpolation and a posteriori error analysis in finite element methods. *ESAIM: Math. Model. Numer. Anal.* **33**, 1187–1202 (1999)
- Casas, E.: Using piecewise linear functions in the numerical approximation of semilinear elliptic control problems. *Adv. Comput. Math.* **26**, 137–153 (2007)
- Casas, E., Tröltzsch, F.: Error estimates for linear-quadratic elliptic control problems. *Analysis and optimization of differential systems*, pp. 89–100. Springer (2003)
- Casas, E., Clason, C., Kunisch, K.: Approximation of elliptic control problems in measure spaces with sparse solutions. *SIAM J. Control Optim.* **50**, 1735–1752 (2012)
- Casas, E., Herzog, R., Wachsmuth, G.: Approximation of sparse controls in semilinear equations by piecewise linear functions. *Numer. Math.* **122**, 645–669 (2012a)
- Casas, E., Herzog, R., Wachsmuth, G.: Optimality conditions and error analysis of semilinear elliptic control problems with L^1 cost functional. *SIAM J. Optim.* **22**, 795–820 (2012b)
- Chambolle, A., Dossa, C.: A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions (2015). <https://hal.archives-ouvertes.fr/hal-01099182>
- Chen, L. Sun, D.F., Toh, K.C.: An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Math. Program.* **161**(1), 237–270 (2017)
- Ciarlet, P.G.: The finite element method for elliptic problems. *Math. Comput.* **36**, xxviii+530 (1978)
- Clason, C., Kunisch, K.: A duality-based approach to elliptic control problems in non-reflexive Banach spaces. *ESAIM Control Optim. Calc. Var.* **17**, 243–266 (2011)
- Collis, S.S., Heinkenschloss, M.: Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. *CAAM TR02-01* (2002)
- Cui, Y.: Large scale composite optimization problems with coupled objective functions: theory, algorithms and applications. PhD thesis, National University of Singapore (2016)
- de Los Reyes, J.C., Meyer, C., Vexler, B.: Finite element error analysis for state-constrained optimal control of the Stokes equations. *Control. Cybern.* **37**, 251–284 (2008)
- Elvetun, O.L., Nielsen, B.F.: The split bregman algorithm applied to PDE-constrained optimization problems with total variation regularization. *Comput. Optim. Appl.* **64**, 1–26 (2014)
- Falk, R.S.: Approximation of a class of optimal control problems with order of convergence estimates. *J. Math. Anal. Appl.* **44**, 28–47 (1973)
- Fazel, M., Pong, T.K., Sun, D.F., Tseng, P.: Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. Appl.* **34**, 946–977 (2013)
- Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**, 17–40 (1976)
- Geveci, T.: On the approximation of the solution of an optimal control problem governed by an elliptic equation. *RAIRO-Analyse numérique.* **13**, 313–328 (1979)
- Glowinski, R., Marroco, A.: Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires, *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique* **9**, 41–76 (1975)
- Hintermüller, M., Ulbrich, M.: A mesh-independence result for semismooth Newton methods. *Math. Program.* **101**, 151–184 (2004)
- Hiriart-Urruty, J.-B., Strodjot, J.-J., Nguyen, V.H.: Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Appl. Math. Optim.* **11**, 43–56 (1984)
- Herzog, R., Ekkehard S.: Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM J. Matrix Anal. Appl.* **31**, 2291–2317 (2010)
- Hinze, M.: A variational discretization concept in control constrained optimization: the linear-quadratic case. *Comput. Optim. Appl.* **30**, 45–61 (2005)
- Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S.: *Optimization with PDE Constraints, Mathematical Modelling: Theory and Applications*, p. 23. Springer, New York (2009)
- Li, X.D., Sun, D.F., Toh, K.C.: QSDPNAL: A two-phase Newton-CG proximal augmented Lagrangian method for convex quadratic semidefinite programming problems (2015). *arXiv:1512.08872*

- Li, X.D., Sun, D.F., Toh, K.C.: A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions. *Math. Program.* **155**, 333–373 (2016)
- Meyer, C., Rösch, A.: Superconvergence properties of optimal control problems. *SIAM J. Control Optim.* **43**, 970–985 (2004)
- Porcelli, M., Simoncini, V., Stoll, M.: Preconditioning PDE-constrained optimization with L^1 -sparsity and control constraints. *Comput. Math. Appl.* **74**, 1059–1075 (2017)
- Rösch, A.: Error estimates for linear-quadratic control problems with control constraints. *Optim. Methods Softw.* **21**, 121–134 (2006)
- Schindele, A., Borzi, A.: Proximal methods for elliptic optimal control problems with sparsity cost functional. *Appl. Math.* **7**, 967–992 (2016)
- Sun, D.F., Toh, K.C., Yang, L.Q.: An Efficient Inexact ABCD Method for Least Squares Semidefinite Programming. *SIAM J. Optim.* **26**, 1072–1100 (2016)
- Stadler, G.: Elliptic optimal control problems with L^1 -control cost and applications for the placement of control devices. *Comp. Optim. Appl.* **44**, 159–181 (2009)
- Ulbrich, M.: Nonsmooth Newton-like methods for variational inequalities and constrained optimization problems in function spaces. Habilitation thesis, Fakultät für Mathematik, Technische Universität München (2002)
- Ulbrich, M.: Semismooth Newton methods for operator equations in function spaces. *SIAM J. Optim.* **13**, 805–842 (2003)
- Wachsmuth, G., Wachsmuth D.: Convergence and regularisation results for optimal control problems with sparsity functional. *ESAIM Control Optim. Calc. Var.* **17**, 858–886 (2011)
- Wathen, A.J.: Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA J. Numer. Anal.* **7**, 449–457 (1987)



Game Theory and Its Applications in Imaging and Vision

17

Anis Theljani, Abderrahmane Habbal, Moez Kallel, and Ke Chen

Contents

Introduction to Game Theory and Paradigm	678
Applications of Game Theory in Image Restoration and Segmentation	681
Applications of Game Theory in Image Registration	683
Introduction to Image Registration	683
Application of Game Theory to a Simple Registration Model	685
Application of Game Theory to Registering Images Requiring Bias Correction	688
Game Models in Deep Learning	696
Generative Adversarial Networks (GANs)	696
GANs for Image Generation: A Two-Player Game	698
GANs for Image Segmentation: A Two-Player Game	699
Conclusion	703
References	703

A. Theljani (✉) · K. Chen (✉)
Department of Mathematical Sciences, University of Liverpool Mathematical Sciences Building,
Liverpool, UK
e-mail: theljani@liverpool.ac.uk; k.chen@liverpool.ac.uk

A. Habbal (✉)
Université Côte d’Azur, Inria, Sophia Antipolis, France
Modeling and Data Science, Mohammed VI Polytechnic University, Benguerir, Morocco
e-mail: Abderrahmane.Habbal@univ-cotedazur.fr

M. Kallel (✉)
Laboratory for Mathematical and Numerical Modeling in Engineering Science (LAM SIN),
University of Tunis El Manar, National Engineering School of Tunis, Tunis-Belvédère, Tunisia
e-mail: moez.kallel@enit.utm.tn

Abstract

It is very common to see many terms in a variational model from Imaging and Vision, each aiming to optimize some desirable measure. This is naturally so because we desire several objectives in an objective functional. Among these is data fidelity which in itself is not unique and often one hopes to have both L_1 and L_2 norms to be small for instance, or even two differing fidelities: one for geometric fitting and the other for statistical closeness. Regularity is another demanding quantity to be settled on. Apart from combination models where one wants both minimizations to be achieved (e.g., total generalized variation or infimal convolution) in some balanced way through an internal parameter, quite often, we demand both gradient and curvature based terms to be minimized; such demand can be conflicted. A conflict is resolved by a suitable choice of parameters which can be a daunting task. Overall, it is fair to state that many variational models for Imaging and Vision try to make multiple decisions through one complicated functional.

Game theory deals with situations involving multiple decision makers, each making its optimal strategies. When assigning a decision (objective) by a variational model to a player by associating it with a game framework, many complicated functionals from Imaging and Vision modeling may be simplified and studied by game theory. The decoupling effect resulting from game theory reformulation is often evident when dealing with the choice of competing parameters. However, the existence of solutions and equivalence to the original formulations are emerging issues to be tackled.

This chapter first presents a brief review of how game theory works and then focuses on a few typical Imaging and Vision problems, where game theory has been found useful for solving joint problems effectively.

Keywords

Noncooperative game theory · Nash equilibria · Joint restoration and segmentation · Image registration · Deep learning

Introduction to Game Theory and Paradigm

Game theory deals with situations involving multiple decision makers. Each decision maker owns the control on some variable known as his action. All actions are collected in an overall variable known as a strategy. Each of the decision makers owns a specific cost function, to be minimized, which depends on the overall strategy variable. Decision makers are also termed by players or agents, and cost functions could also be replaced by payoffs, to be maximized instead. For readers who are familiar with, let us rephrase the classical optimization problems as follows: optimization deals with situations where a single decision maker owns control over one single overall strategy (all optimization variables), and optimizes a single cost/payoff function, possibly subject to constraints.

To start with some comprehensive and easy-reading reference, the book (Gibbons 1992) introduces, most if not all, the must-have material, including the earliest models of Cournot and Bertrand, those of Stackelberg and actually illustrates with many examples how the game theory first emerged from the need to model economic behavior.

We focus in this introduction on noncooperative games, which means that the players do not share the same cost function, or they do not aggregate their costs into a single one (e.g., a weighted sum). We do not consider as well finite or discrete games, where the set of strategies is either finite (e.g., prisoner's dilemma) or discrete (e.g., games on graphs).

Noncooperative games may be static or dynamic. Roughly speaking, in a dynamic game, players sequentially observe actions of other players and then choose their optimal responses. In a static game, players choose their best responses to the others without exchange (or communication) of information. Remark that the notion of time involved in games is not necessarily the physical time involved in, for example, state equations. As well, a static game could be played by players whose cost functions are constrained by, for example, unsteady fluid mechanics. Games may also be with complete information, meaning that all players know each other's strategy spaces and cost functionals (including their own ones). The failure of this assumption is termed as a game with incomplete information, see Gibbons (1992) for details.

Noncooperative games may also be differential and/or stochastic.

Differential games involve state equations governed by system of differential equations. They model a huge variety of competitive interactions, in social behavior, economics, biology among many others, predator-prey, pursuit-evasion games, and so on (Isaacs 1999). Stochastic games theory, starting from the seminal paper by Shapley (1953), occupies nowadays most of the game theory publications, and a vast literature is dedicated to stochastic differential games (Friedman 1972), robust games (Nishimura et al. 2009), games on random graphs, or agents learning games (Hu and Wellman 2003), among many other branches, and it is definitely out of the scope of the introductory section to review all aspects of the field. See also the introductory book (Neyman and Sorin 2003) to the basic concepts of the stochastic games theory.

Solutions to noncooperative games are called equilibria. Contrarily to classical optimization, the definition of an equilibrium depends on the game setting (game rules). Within the static with complete information setting, a relevant one is the so-called Nash equilibrium (NE).

We consider primarily the standard static, under complete information, Nash equilibrium problem (NEP) (Gibbons 1992).

Definition 1. An NEP consists of $p \geq 2$ decision makers (i.e., players), where each player $i \in \{1, \dots, p\}$ tries to solve his optimization problem:

$$(\mathcal{P}_i) \quad \min_{\mathbf{x}_i \in \mathbb{X}_i} y_i(\mathbf{x}), \quad (1)$$

where $\mathbf{y}(\mathbf{x}) = [y_1(\mathbf{x}), \dots, y_p(\mathbf{x})] : \mathbb{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^p$ (with $n \geq p$) denotes a vector of cost functions (a.k.a. pay-off or utility functions), y_i denotes the specific cost function of player i , and the strategy variable \mathbf{x} consists of block components $\mathbf{x}_1, \dots, \mathbf{x}_p$ ($\mathbf{x} = (\mathbf{x}_j)_{1 \leq j \leq p}$).

Each block \mathbf{x}_i denotes the action variable of player i and \mathbb{X}_i its corresponding action space and $\mathbb{X} = \prod_i \mathbb{X}_i$. We shall use the convention $y_i(\mathbf{x}) = y_i(\mathbf{x}_i, \mathbf{x}_{-i})$ when we need to emphasize the role of \mathbf{x}_i .

Definition 2. A Nash equilibrium (NE) $\mathbf{x}^* \in \mathbb{X}$ is a strategy such that:

$$\text{(NE)} \quad \forall i, 1 \leq i \leq p, \quad \mathbf{x}_i^* = \arg \min_{\mathbf{x}_i \in \mathbb{X}_i} y_i(\mathbf{x}_i, \mathbf{x}_{-i}^*). \quad (2)$$

In other words, when all players have chosen to play an (NE), then no single player has incentive to move from his \mathbf{x}_i^* . Let us however mention by now that, generically, Nash equilibria are not efficient, that is, do not belong to the underlying set of best compromise solutions, called Pareto front, of the objective vector $(y_i(\mathbf{x}))_{\mathbf{x} \in \mathbb{X}}$.

An important class of games are the so-called potential games. As introduced in the survey paper (David and Hernández-Lerma Onésimo 2016), in the static case, a noncooperative game is said to be a potential game if there is a real-valued function, called a potential function, such that a strategy profile that optimizes the potential function is a Nash equilibrium for the game. This is precisely one of the key properties of potential games; namely, in a potential game one can find Nash equilibria by optimizing a single function rather than using a fixed-point argument as is typically done for noncooperative games.

From application side, few papers are dedicated to engineering applications involving partial differential state equations where distributed parameters are seen as Nash strategies. In Habbal et al. (2004), a Nash game is set up between two physical processes, heat transfer and structural mechanics, using cooling and structural material densities (like as in topology optimization) as Nash strategies. Nash games could also be used to model biological processes, as introduced in Habbal (2005), where tumoral angiogenesis is modeled as a Nash game between pro- and anti-angiogenic factors and involves porous media and elasticity state equations. In Roy et al. (2017), Nash strategies are used to model the cognitive process of pedestrian avoidance, with Fokker-Planck state equations.

Engineering applications involving multidisciplinary optimization may also benefit from reframing within a Nash game framework, see Desideri et al. (2014) for an overview and Benki et al. (2015) for an original application in nonlinear mechanics. Finally, and in close connection to image processing, ill-posed inverse problems may find a strikingly efficient benefit in being reformulated as Nash games. See Habbal and Kallel (2013) for a novel approach in solving data recovery problems,

and Habbal et al. (2019); Chamekh et al. (2019) in devising new algorithms to solve the coupled data recovery and parameter or shape identification problems.

Applications of Game Theory in Image Restoration and Segmentation

There are two classical problems associated with image processing: the image denoising (restoration) and contour identification (segmentation). To address these issues, there are various approaches, such as the stochastic modeling, the wavelet approach and the variational approach leading to the partial differential equations. Image restoration is an inverse problem which consists of finding the original image from another observed, often linked by the equation, $I_0 = \mathcal{T}I + v$, where \mathcal{T} is a linear operator modeling the blur, I a (mathematical) image defined by the intensity (or gray level), and v represents the noise (Gaussian for example). Image segmentation is the process of extracting objects from an image, and can be formulated as finding a finite collection $\{\Omega_i\}_{i=1}^K$ of disjoint open subsets of Ω , where Ω is an open and bounded subset of \mathbb{R}^2 and represents the image domain. The restoration and segmentation of the image can be performed simultaneously. In this case, one has to solve a minimization problem of a sum of two energies (see, e.g., Mumford-Shah functional (Mumford 1989)). One favors image regularization and the other detects and enhances the contours presented in the image. If the regularization term of the energy is favored over the segmentation term, then the contours are smoothed and hence destroyed. On the other hand, if the segmentation contribution to the energy is made stronger than the regularization contribution, then we might obtain an oversegmented image.

A game-theoretic approach was proposed in Kallel et al. (2014) to simultaneously restore and segment noisy images. The method is based on iterative negotiation between the two antagonistic processes, segmentation and restoration, where acceptable solutions arise then as stationary (noncooperative) decisions. In this work, the game theory concepts are used and define two players: one is interested in the regularization of the image and the other is concerned with its segmentation. Each of two players will try to increase his profit by making an adequate decision until a “Nash equilibrium” is reached. More specifically, the restoration player’s goal is to minimize the functional

$$\mathcal{J}_1(I, C) = \int_{\Omega} (I - I_0)^2 dx + \mu \int_{\Omega \setminus C} |\nabla I|^2 dx, \quad (3)$$

and the segmentation player’s objective is to minimize the functional

$$\mathcal{J}_2(I, C) = \sum_{i=1}^K \int_{\Omega_i} (I_0 - I_i)^2 dx + v|C|, \quad \text{where } I_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} I(x) dx. \quad (4)$$

The functional (4) is inspired from the Mumford-Shah one and it is obtained by replacing the restriction of I in each connected component Ω_i of Ω with its mean over Ω_i . To summarize this approach, the authors consider a two-player static of complete information game where the first player is restoration, and the second is segmentation. Restoration minimizes the cost $\mathcal{J}_1(I, C)$ with action on the intensity field I , while segmentation minimizes the cost $\mathcal{J}_2(I, C)$ with action on the discontinuity set C . In this case, solving the game amounts to finding a Nash equilibrium (NE), defined as a pair of strategies (I^*, C^*) , such that

$$\begin{cases} I^* = \operatorname{argmin}_I \mathcal{J}_1(I, C^*), \\ C^* = \operatorname{argmin}_C \mathcal{J}_2(I^*, C). \end{cases} \tag{5}$$

The minimizer I^* is sought in the Sobolev space $H^1(\Omega \setminus C^*)$ and C^* is sought in the set of the union of curves made of a finite set of $C^{1,1}$ -arcs.

To compute this equilibrium, they use the classical iterative method with relaxation (Uryas'ev 1994) as described in Algorithm 1. The main advantage of using this algorithm is that $\bar{I}^{(k)}$ and $\bar{C}^{(k)}$ can be numerically computed, separately and parallelly, using descent algorithms.

Algorithm 1 Nash equilibrium algorithm

- 1: Initial guess: $S^{(0)} = (I^{(0)}, C^{(0)})$. Set $k = 0$.
 - 2: **repeat**
 - 3: $\bar{I}^{(k)} = \operatorname{argmin}_I \mathcal{J}_1(I, C^{(k)})$
 - 4: $\bar{C}^{(k)} = \operatorname{argmin}_C \mathcal{J}_2(I^{(k)}, C)$
 - 5: $S^{(k+1)} = (I^{(k+1)}, C^{(k+1)}) = \tau S^{(k)} + (1 - \tau)(\bar{I}^{(k)}, \bar{C}^{(k)})$ {for τ fixed, $0 < \tau < 1$ }
 - 6: $k = k + 1$
 - 7: **until** $S^{(k)}$ converges
-

Finally, the authors use a level-set approach to get rid of the tricky control dependence of functional spaces. After, a numerical study is carried on some real images in order to evaluate the effectiveness of the proposed algorithm. In particular, they show that by decoupling the Mumford-Shah functional using the game algorithm, the dependence on the regularization parameters μ and ν is uncorrelated and the choice of their values becomes more flexible and natural. On the other hand, the dependence of the functional \mathcal{J}_2 only on the mean of I in each connected component has a significant effect on the speed of convergence. In Fig. 1, a numerical result using only one level-set function is represented. The top row displays the evolution of curves over the corresponding images $I^{(k)}$, $k \in \{0, 10, 50\}$. The bottom row displays the final segmentation result (second image) and denoised image (third image) with $PSNR = 31.98$. For this case, the algorithm converges after 135 iterations.

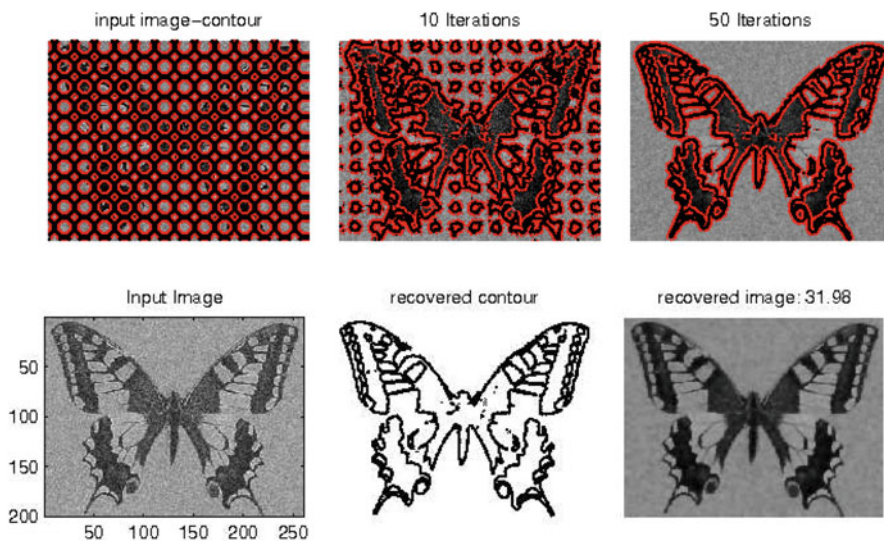


Fig. 1 Top row: noisy image with Gaussian noise (variance = 0.2) and initial contour, evolution by iterations. Bottom row: segmentation and restoration of image by the proposed algorithm with ($\nu = 0.2$, $\mu = 0.01$), for $k = 135$. CPU time = 117 sec

Applications of Game Theory in Image Registration

There exist many image registration models: each is designed for one class of problems. It is challenging to find an universally robust model that can deal with all registration problems, due to the inherent difficulties of image registration. The previous section discussed how game theory can be used to enhance a model for image restoration and segmentation. Here we shall see that game theory is also a natural tool to reformulate an image registration model in achieving better performance and robustness.

In this section, we review recent works on using game theory to design and reformulate the traditional variational models for deformable image registration. The advantages gained will be in reduction of the burden of tuning many parameters; hence a more robust model is obtained. The ideas are generally applicable to almost other variational models.

Introduction to Image Registration

Image registration (Chen et al. 2019) aims to align two given images through mapping one (the template image T) to the other (the reference image R) so that

the aligned (or registered) image $T(\phi)$ may be used to give us complementary information from T to R , or highlight the differences between T and R . Here $\phi(x) = x + \mathbf{u}(x)$ where $\mathbf{u}(x) = (u_1(x), u_2, \dots, u_d(x))$ is the unknown map concerned if $x \in \Omega \subset \mathbb{R}^d$. Practically $d = 2, 3$ are more common.

To find ϕ , a typical variational model takes the form ; Chen et al. (2019)

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{u}) = F(\mathbf{u}) + \alpha S(\mathbf{u}) + \beta C(\mathbf{u}) \quad (6)$$

where $F(\mathbf{u})$, $S(\mathbf{u})$, $C(\mathbf{u})$ are, respectively, the fitting terms to align T, R , the regularization term to overcome the ill-posedness of minimizing the fitting term alone and the control term to ensure the underlying map ϕ does not have folding (e.g., by making ϕ diffeomorphic).

Flexibilities exist for specifying each of the three terms in (6) differently, though none of these flexibilities is sufficient to construct a robust model for a wider class of problems than with a fixed choice of terms.

First, since the fitting term F is supposed to measure the dissimilarity of T, R , it has many possible choices especially for multi-modality pairs of T, R (e.g., T is from MRI and R is from ultrasound).

For single modal images (e.g., when both T, R are CT images), a popular choice for F is the SSD (sum of squared differences)

$$F(\mathbf{u}) = \int_{\Omega} |T(x + \mathbf{u}) - R(x)|^2 dx.$$

For multimodal image pairs, one may take the popular choice of mutual information (Maes et al. 1997). This statistical measure has also been improved a few times since 1997. One alternative is the normalized gradient differences (NGD)

$$F(\mathbf{u}) = \int_{\Omega} |\nabla_n T(x + \mathbf{u}) - \nabla_n R(x)|^2 dx$$

where $\nabla_n T = \nabla T / |\nabla T|$; however, we remark that this fitting term is not very robust, a better variant is proposed in Theljani and Chen (2019a).

Second, as for designing the regularizer S , one way is to regularize all deformation directions individually:

$$S(\mathbf{u}) = S(u_1) + \dots + S(u_d) \quad (7)$$

but one may introduce some coupling between these individual terms.

Finally, the control term C is designed to ensure $\det(\nabla\phi) > 0$. If it makes sense to achieve volume or area preservation in features of T, R , that is, $\det(\nabla\phi) = 1$, a simple method is to define

$$C(\mathbf{u}) = \int_{\Omega} (\det(\nabla\phi) - 1)^2 dx.$$

However, if this is not appropriate for other applications, a robust method seems to define

$$C(\mathbf{u}) = \int_{\Omega} \Phi(\mu(\phi))^2 dx,$$

where Φ is some smooth function (Zhang and Chen 2018) and μ is the Beltrami coefficient for the same mapping ϕ projected to a complex plane with $\phi = \phi_1(x) + \mathbf{i}\phi_2(x)$ with $d = 2$. The central idea is the equivalence relationship $|\mu| < 1 \Leftrightarrow \det(\nabla\phi) > 0$, which facilitates the design of an unconstrained optimization problem (Lam and Lui 2014).

Of course, it is entirely appropriate to propose a minimization problem like (6) without its third term, and to add the constraint $\det(\nabla\phi) > 0$ as done in Zhang et al. (2016) and Thompson and Chen (2019). However, nonlinear constraints are not easy to deal with in numerical implementations.

One drawback of the Beltrami coefficient is that such a quantity μ does not exist when $d \geq 3$, though there are some recent attempts to generalize it to high dimensions. The recent work by Zhang and Chen (2020) designed a 3D Beltrami coefficient-like quantity that possesses the same property as 2D, and hence extended the classical work.

Another method to replace the third term in (6) is the so-called inverse consistent formulation where the folding is avoided by simultaneously registering T to R by ϕ and also R to T by ψ . The central idea is $\phi(\psi) = I$ or $\psi(\phi) = I$ so that the map is inversely consistent and does not fold. See Christensen et al. (2007), Thompson and Chen (2019), Theljani and Chen (2019c) and Chen and Ye (2010).

Application of Game Theory to a Simple Registration Model

To illustrate the idea of using the game theory, let us first consider the diffusion registration model for simple modal images before we elaborate on more robust models in later subsections.

Let us start with the simple diffusion model (Fischer and Modersitzki 2002) which takes the following form:

$$\min_{\mathbf{u} \in W^{1,2}(\Omega)} \mathcal{J}(\mathbf{u}) = \int_{\Omega} |\nabla \mathbf{u}|^2 dx + \alpha M(\mathbf{u}) \quad (8)$$

where $M(\cdot)$ is a similarity measure. One application using game theory for this model is to consider two different similarity measures. For the simple case of monomodal images, using the sum of squared differences is used because of the grey value constancy assumption. However, in some scenarios, the SSD has a big drawback: it is quite susceptible to slight changes in brightness, which often appear in natural scenes. Therefore, it is useful to allow some small variations in the grey value and help to determine the displacement vector by a criterion that is

invariant under grey value changes. Thus, to have a model which is less sensitive to illumination variations, it is interesting to combine SSD with another measure, which can capture more information, such as gradients, and fulfill the gradient constancy assumption.

Coupled Measures: Nongame Approach

The combination of the two measures can be done as a classical variational formulation where one has to optimize one single energy which couples both measures. In case where the SSD is combined with the NGD for monomodal image registration, the natural vibrational approach consists of solving

$$\begin{aligned} \min_{\mathbf{u} \in W^{1,2}(\Omega)} \mathcal{J}(\mathbf{u}) &= \int_{\Omega} |\nabla \mathbf{u}|^2 dx + \lambda_1 \int_{\Omega} |T(x + \mathbf{u}) - R(x)|^2 dx + \lambda_2 \\ &\times \int_{\Omega} |\nabla_n T(x + \mathbf{u}) - \nabla_n R(x)|^2 dx \end{aligned} \quad (9)$$

This approach may lead to a solution which is sensible to choice of the weighting parameters λ_1 and λ_2 between the two measures. In fact, if more weights are put on the SSD term, it seems that the model does not work because the SSD will not handle well the regions in the images that distorted by varying illumination. Only few regions are well registered where there is no big difference in the intensity variation between the two images. Reversely, if the NGD contribution to the model is too much strong by taking large value of λ_2 , then the solution seems to be well registered in regions of varying intensity whereas the registration quality is poorer than the SSD model in clean regions, that is, nor varying intensities.

Coupled Measures: Game Approach

In game formulation, the combination of the two measures can be done differently from the classical approach. We can design a game where the two measures are incorporated in different models that have some communications through a coupling term. As an example, we could consider the following game model: Find a Nash equilibrium (NE) $(\mathbf{u}^*, \mathbf{v}^*)$ such that

$$\begin{cases} \mathbf{u}^* = \operatorname{argmin}_{\mathbf{u} \in W^{1,2}(\Omega)} \mathcal{J}_1(\mathbf{u}, \mathbf{v}^*), \\ \mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in W^{1,2}(\Omega)} \mathcal{J}_2(\mathbf{u}^*, \mathbf{v}), \end{cases} \quad (10)$$

where

$$\mathcal{J}_1(\mathbf{u}, \mathbf{v}) = \int_{\Omega} |\nabla \mathbf{u}|^2 dx + \int_{\Omega} |T(x + \mathbf{u}) - R(x)|^2 dx + \lambda \int_{\Omega} (\mathbf{u} - \mathbf{v})^2 dx, \quad (11)$$

$$\mathcal{J}_2(\mathbf{u}, \mathbf{v}) = \int_{\Omega} |\nabla \mathbf{v}|^2 dx + \int_{\Omega} |\nabla_n T(x + \mathbf{v}) - \nabla_n R(x)|^2 dx + \lambda \int_{\Omega} (\mathbf{u} - \mathbf{v})^2 dx \quad (12)$$

The first energy uses the sum of squared difference as similarity measure, whereas the second energy uses the normalized gradient difference term (NGD). The third part is a coupling term which serves for the communication between the two players u and v . The first player tries to minimize his one cost $\mathcal{J}_1(\cdot)$ taking into account the information about the gradient consistency coming from the second player v through the coupling term, and vice versa.

Examples

Figure 2 shows an example of using game model for a pair of MRI images registration. We assess the registration quality by measuring the normalized cross correlation coefficient (NCC) between the registered image $T(\mathbf{u})$ and R (closer NCC to 1 means better registration). Mainly, the example illustrates how two players in a game model can cooperate to achieve better registration quality. However, by considering two separate models, that is, no communication for $\lambda = 0$, the first model in (11) is unable to achieve an acceptable result.

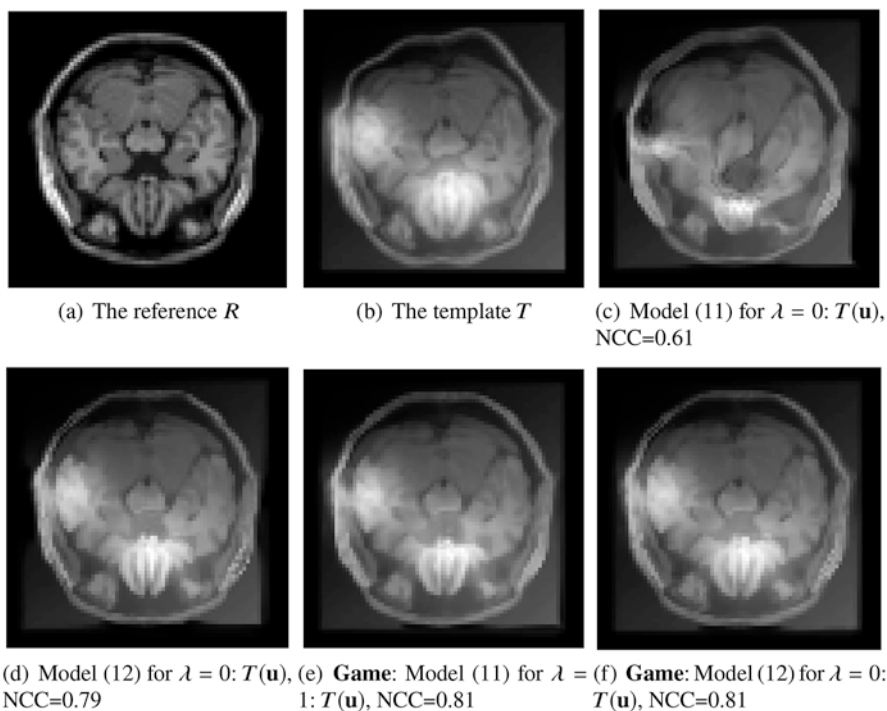


Fig. 2 Example 1: the game approach for registering a pair of MRI images. The template image T contains some undesirable artifact. Clearly, the game approach is able to cope with this case because of the use of two different measures. (a) The reference R (b) The template T (c) Model (11) for $\lambda = 0$: $T(\mathbf{u})$, NCC=0.61 (d) Model (12) for $\lambda = 0$: $T(\mathbf{u})$, NCC=0.79 (e) **Game**: Model (11) for $\lambda = 1$: $T(\mathbf{u})$, NCC=0.81 (f) **Game**: Model (12) for $\lambda = 0$: $T(\mathbf{u})$, NCC=0.81

Application of Game Theory to Registering Images Requiring Bias Correction

In many real-life applications, even a pair of monomodality images acquired from the same source can differ from each other, leading to inaccurate registration results. The difference is often presented as an undesirable artifact either caused by the device itself (spatially homogeneous signal response, bias field, and shading in MRI images) or caused by the imaging modality itself such as perfusion CT which creates some high contrasted regions in the image. In order to obtain accurate registration results and to cope with these problems, many models have been developed for intensity correction (Aghajani et al. 2016; Ebrahimi and Martel 2009; Ghaffari and Fatemizadeh 2018; Li et al. 2009; Rak et al. 2017; Kim and Tagare 2014). It is important to note that, without intensity correction, both monomodality and multimodality models may fail to register the images correctly because bias introduces incorrect intensity values or false edges.

The artifacts can be of either additive or multiplicative type (Modersitzki and Wirtz 2006; Chumchob and Chen 2012; Ghaffari and Fatemizadeh 2018). It has been generally accepted that the image T with bias field, generally presented as a mixed type, relates to the “true” unbiased image T^* via the following affine like intensity relationship: $T = mT^* + s$, where $m(\mathbf{x})$ and $s(\mathbf{x})$ are responsible for the intensity-correction. Rigorously speaking, the word “affine” is misleading because both m , s are never constants so the model is highly nontrivial. Once m , s are found or estimated, the registration task is to find the deformation field \mathbf{u} such that $T^*(\mathbf{u}) \approx R$. Denote by $T_c(\mathbf{u}) = T^*(\mathbf{u})$ the corrected and registered image of T . Hence the equivalent statement to the model $T = mT^* + s$ is

$$R_1 = mR + s, \quad T(\mathbf{u}) \approx R_1, \quad \text{with } T_c(\mathbf{u}) = \frac{T(\mathbf{u}) - s}{m} \approx R, \quad (13)$$

where $T(\mathbf{u})$ is the uncorrected and registered image, carrying the bias field features from T and aligned with R , that is, one may minimize one of these fidelity terms for m , s , \mathbf{u} in some norm:

$$\|mR + s - T(\mathbf{u})\|, \quad \left\| \frac{T(\mathbf{u}) - s}{m} - R \right\|.$$

Any model building on minimization of the above quantities may be much simplified if one of the unknowns is dropped (i.e., $m \equiv 1$ or $s \equiv 0$); however, a full model including both m and s always gives better results in solution quality. In fact, in many cases, intensity correction by either multiplicative or additive model is not always enough (Wang and Pan 2014; Vovk et al. 2007; Park et al. 2019) thus a combined model is necessary.

Non-game Approach

A classical variational approach for joint full bias correction and image registration consists in solving the multivariate optimization problem

$$\mathbf{JM} \quad \mathcal{J}(\mathbf{u}, m, s) := \lambda \int_{\Omega} |mR + s - T(\mathbf{u})|^2 d\mathbf{x} + \mathcal{R}(\mathbf{u}, s, m), \quad (14)$$

where $\mathcal{R}(\mathbf{u}, s, m)$ will be chosen to be the same as comparable models shortly. Since m is not a constant function, the first term in (14) is not convenient for numerical implementation for solving the subproblems. The authors in Theljani and Chen (2019b) proposed a variant to this term. They transformed the multiplicative term into an additive one since the latter is more convenient (a simple filtering problem). This transformation was obtained by applying a splitting method to the bias model (13). The splitting leads the additive problems

$$K_l = m_l + R_l, \quad T(\mathbf{u}) = e^{K_l} + s, \quad (15)$$

which is easier to handle, assuming $m, R > 0$. Here $R_l = \ln(R)$ is known since R is given, $m_l = \log(m)$, and K_l is the intermediate quantity as a spitting variable. The application of a logarithmic transform in the context of intensity transformations increases the contrast between certain intensity values (Duan et al. 2015; Chang et al. 2017; Bansal et al. 2004; Van Leemput et al. 1999). After applying the penalty method to incorporate the constraints (15), the new variational model takes the following form

$$\mathbf{CV} \quad \min_{\mathbf{u}, s, m_l, K_l} \{ \mathcal{J}(\mathbf{u}, s, m_l, K_l) = \mathcal{R}(\mathbf{u}, s, m_l, K_l) + \lambda_1 \int_{\Omega} |T(\mathbf{u}) - e^{K_l} - s|^2 d\mathbf{x} + \lambda_2 \int_{\Omega} |m_l + R_l - K_l|^2 d\mathbf{x} \} \quad (16)$$

where \mathbf{u} is the main deformation field variable, $\mathcal{R}(\cdot)$ contains regularization terms associated to all four unknowns (to be specified), and the rest of the energy are two fidelity terms. Clearly there are no multiplicative terms in (16) as designed. One would normally specify $\mathcal{R}(\cdot)$ and try to solve the joint optimization problem by some techniques, for example, the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) or Augmented Lagrangian (Bonnans et al. 2006; Boyd et al. 2011). The problem (16) is split into 4 subproblems for each of the main variables: \mathbf{u}, s, m_l, K_l . There are two challenges: i) choosing the 5 parameters (assuming there are 3 new parameters from $\mathcal{R}(\cdot)$) suitably is a highly nontrivial task; ii) one cannot avoid coupling all 4 variables in any subproblem. This challenge can be solved using a game theory formulation as described in the sequel.

Game Model

It was shown in Theljani and Chen (2019b) that it is more convenient to reformulate (16) to another form using the Nash game idea where both of these two

challenges are overcome: first, each subproblem will have one parameter which can be tuned for that subproblem in an easier way; second, it is possible to modify the above subproblems to reduce couplings and hence improve convergence. The authors demonstrated that the game model offers a better solution for two main aspects: choice of underlying parameters and proof of solution existence. In fact, the K_I subproblem in model (16) has three terms and involves two penalty parameters λ_1 and λ_2 , which are pretended to be large enough. The solution will be sensitive to these two parameters and the optimal choice is nontrivial. We shall reformulate this problem to yield only one parameter (instead of two) by considering a game approach that has a separable structure and makes the model less sensitive to these parameters. The joint model (16) was reformulated as a game where the solution is a **Nash equilibrium** defined by $(A_1, A_2, A_3, A_4) = (\mathbf{u}, s, m_I, K_I)$ in the space $\mathcal{X} = \mathcal{W} \times W^{1,2}(\Omega) \times W^{1,2}(\Omega) \times W^{1,2}(\Omega)$ where $\mathcal{W} = W^{2,2}(\Omega, \mathbb{R}^2) \cap W_0^{1,2}(\Omega, \mathbb{R}^2)$. The space \mathcal{X} is endowed with the following norm:

$$\|\mathbf{z}\|_{\mathcal{X}} = \left(\|\mathbf{u}\|_{\mathcal{W}}^2 + \|\nabla s\|_{W^{1,2}(\Omega)}^2 + \|\nabla m_I\|_{W^{1,2}(\Omega)}^2 + \|\nabla K_I\|_{W^{1,2}(\Omega)}^2 \right)^{1/2},$$

where $\|\mathbf{u}\|_{\mathcal{W}} = \left(\|\nabla \mathbf{u}\|_2^2 + \|\nabla^2 \mathbf{u}\|_2^2 \right)^{1/2}$. The game formulation allows many choices of energies $\mathcal{R}_i(\cdot)$ and $\mathcal{G}_i(\cdot)$ whose terms may not be part of each other. The choice of the different energies leads to either potential or non-potential games (Monderer and Shapley 1996).

The Potential Game

The potential game structure is very important because it makes easy to prove the existence of Nash equilibrium (**NE**) (Nash 1950, 1951). One example is to make the particular choice of the following energies $\mathcal{J}_i(\cdot) = \mathcal{R}_i(\cdot) + \mathcal{G}_i(\cdot)$ with

$$\left\{ \begin{array}{ll} \mathcal{R}_1(\mathbf{u}) = \|\mathbf{u}\|_{\mathcal{W}}^2, & \mathcal{G}_1(\mathbf{u}, s, m_I, K_I) = \lambda_1 \int_{\Omega} |T(\mathbf{u}) - e^{K_I} - s|^2 d\mathbf{x}, \\ \mathcal{R}_2(s) = \int_{\Omega} |\nabla s|^2 d\mathbf{x}, & \mathcal{G}_2(\mathbf{u}, s, m_I, K_I) = \lambda_2 \int_{\Omega} |T(\mathbf{u}) - e^{K_I} - s|^2 d\mathbf{x}, \\ \mathcal{R}_3(m_I) = \int_{\Omega} |\nabla m_I|^2 d\mathbf{x}, & \mathcal{G}_3(\mathbf{u}, s, m_I, K_I) = \lambda_3 \int_{\Omega} |m_I + R_I - K_I|^2 d\mathbf{x}, \\ \mathcal{R}_4(K_I) = \int_{\Omega} |\nabla K_I|^2 d\mathbf{x}, & \mathcal{G}_4(\mathbf{u}, s, m_I, K_I) = \lambda_4 \int_{\Omega} |m_I + R_I - K_I|^2 d\mathbf{x} \\ & + \lambda_5 \int_{\Omega} |T(\mathbf{u}) - e^{K_I} - s|^2 d\mathbf{x}, \end{array} \right. \tag{17}$$

where $\mathcal{R}_i(\cdot)$ is the regularization term in energy i . There are many possible choices of regularization leading to different solution spaces. For the deformation \mathbf{u} , the authors in Theljani and Chen (2019b) used regularizers based on combined first and second-order derivatives. Using only the first-order derivatives, that is, H^1 semi-norm, is sensitive to affine preregistration. We avoid this problem by combining it with the second-order derivative term which are not sensitive to (affine) preregistration as it has the affine transformations in its kernel. Moreover, this choice

penalizes oscillations and also allows smooth transformations in order to get visually pleasing registration results. The variables K_l , m_l , and s are chosen in the space $W^{1,2}(\Omega)$ and we could consider different spaces such as $W^{2,2}(\Omega)$ or the space of bounded variation functions $BV(\Omega)$. The formulation in (17) is special cases of game formulation known as a potential game (**PG**) (Monderer and Shapley 1996) which amounts to find a minimizer of an energy $\mathcal{L}(\cdot) = \sum_i^4 \mathcal{J}_i(\mathbf{u}, s, m_l, K_l)$ in (16) – then the game model reduces to an ADMM algorithm if alternating iterations are used or a Nash equilibrium of (16) is a minimizer of $\sum_i^4 \mathcal{J}_i(\mathbf{u}, s, m_l, K_l)$. We refer the reader to Monderer and Shapley (1996), Attouch and Soueicat (2008) and Attouch et al. (2008) for more details about potential game in PDEs.

The Non-potential Game

Instead of (17), it is possible to modify $\mathcal{J}_3, \mathcal{J}_4$ to get new subproblems which lead to a better model than (17); the new energies to be minimized are still denoted by $\mathcal{J}_i = \mathcal{R}_i + \mathcal{G}_i$, for $i = 1, 2, 3, 4$, with all terms defined in (17) except these three new terms, that is,

$$\begin{cases} \mathcal{R}_1(\mathbf{u}) = \|\mathbf{u}\|_{\mathcal{W}}^2, & \mathcal{G}_1(\mathbf{u}, s, m_l, K_l) = \lambda_1 \int_{\Omega} |T(\mathbf{u}) - e^{K_l} - s|^2 d\mathbf{x}, \\ \mathcal{R}_2(s) = \int_{\Omega} |\nabla s|^2 d\mathbf{x}, & \mathcal{G}_2(\mathbf{u}, s, m_l, K_l) = \lambda_2 \int_{\Omega} |T(\mathbf{u}) - e^{K_l} - s|^2 d\mathbf{x}, \\ \mathcal{R}_3(m_l) = \int_{\Omega} |\nabla m_l|^2 d\mathbf{x}, & \mathcal{G}_3(\mathbf{u}, s, m_l, K_l) = \lambda_3 \int_{\Omega} |m_l + R_l - \ln(T(\mathbf{u}) - s)|^2 d\mathbf{x}, \\ \mathcal{R}_4(K_l) = \int_{\Omega} |\nabla K_l|^2 d\mathbf{x} + \iota_{\Lambda}(K_l), & \mathcal{G}_4(\mathbf{u}, s, m_l, K_l) = \lambda_4 \int_{\Omega} |m_l + R_l - K_l|^2 d\mathbf{x}, \end{cases} \tag{18}$$

where $\Lambda = \{K_l \in L^2(\Omega); K_{\min} \leq K_l \leq K_{\max}\}$ is a closed and convex set; and $\iota_{\Lambda}(\cdot)$ is a projection into Λ . The variables K_l are bounded for theoretical reasons in order to prove the existence of a Nash equilibrium (**NE**). In this case, an **NE** is not a minimizer of $\sum_i^4 \mathcal{J}_i(\mathbf{u}, s, m_l, K_l)$, which makes the proof of the existence difficult. Formally this Nash game problem is called a non-potential game (denoted by **NPG**). Clearly the essential simplification is in \mathcal{G}_4 and there are other possible alternative formulations, for example, using L_1 semi-norm. These changes simplify the K_l -problem in (17), equivalently in (16), where the K_l -energy has three terms and which necessitates two regularization parameters λ_4 and λ_5 . Whereas, in the game approach (18), the same problem consists only of regularization and one fidelity term, that is, has only one parameter λ_4 . Moreover, to discuss any theory for (18), the non-convexity should be addressed, e.g. the energy $\mathcal{G}_1(\cdot)$ is non-convex w.r.t \mathbf{u} . Non-convexity means that we cannot apply the Nash theorem (Nash 1951) to show the existence of an **NE**.

Iterative Algorithm

To compute the (NE), the authors in Theljani and Chen (2019b) used alternating Forward-Backward algorithm (ADMM-like), by means of the following iterative process:

Algorithm 2 Forward-Backward algorithm for computing a Nash equilibrium

- Set $k = 0$ and choose an initial guess $\mathbf{z}^{(0)} = (\mathbf{u}^{(0)}, s^{(0)}, m_l^{(0)}, K_l^{(0)})$.
- Step 1: Compute (in parallel) $(\mathbf{u}^{(k+1)}, s^{(k+1)}, m_l^{(k+1)}, K_l^{(k+1)})$ solution of

$$\bar{\mathbf{u}}^{(k)} = \mathbf{u}^k - \gamma \nabla \mathcal{G}_{\mathbf{u}}(\mathbf{u}^k, s^k, m_l^k, K_l^k), \quad \mathbf{u}^{(k+1)} = \mathbf{prox}_{\gamma \mathcal{R}_1}(\bar{\mathbf{u}}^{(k)}) \quad (19)$$

$$\bar{s}^{(k)} = s^k - \gamma \nabla \mathcal{G}_s(\mathbf{u}^k, s^k, m_l^k, K_l^k), \quad s^{(k+1)} = \mathbf{prox}_{\gamma \mathcal{R}_2}(\bar{s}^{(k)}) \quad (20)$$

$$\bar{m}_l^{(k)} = m_l^k - \gamma \nabla \mathcal{G}_{m_l}(\mathbf{u}^k, s^k, m_l^k, K_l^k), \quad m_l^{(k+1)} = \mathbf{prox}_{\gamma \mathcal{R}_3}(\bar{m}_l^{(k)}) \quad (21)$$

$$\bar{K}_l^{(k)} = K_l^k - \gamma \nabla \mathcal{G}_{K_l}(\mathbf{u}^k, s^k, m_l^k, K_l^k), \quad K_l^{(k+1)} = \mathbf{prox}_{\gamma \mathcal{R}_4}(\bar{K}_l^{(k)}) \quad (22)$$

- If $\frac{\|\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\|_2}{\|\mathbf{z}^{(k)}\|_2} \leq \epsilon$, stop. Otherwise $k = k + 1$, go to Step 1.

Examples

The experiments show that the game approach can have significant robustness in presence of bias noise and varying illumination. In all examples, the weighting parameters were fixed as $\lambda_1 = 200$ for the \mathbf{u} -subproblem, $\lambda_2 = 20$ for the s -subproblem, $\lambda_3 = 1$ for the m_l -subproblem, and $\lambda_4 = 5$ for the K_l -subproblem. A multi-resolution technique was used to initialize the displacement \mathbf{u} in order to avoid local minima and to speed up registration. The game model, denoted by “**Game**,” is compared with joint models (14) “**JM**” and the classical variational model (16) denoted by “**CV**.” The last models are the more natural choices for the class of joint problems. The authors also compared with the Mutual Information based multi-modality model where they minimize an energy which uses the same regularizer $\mathcal{R}_1(\cdot)$ and the Mutual Information as similarity measure (denoted by “**MI**” below). Numerical experiments on “**MI**” are performed using the publicly available image registration toolbox – Flexible algorithms for image registration (FAIR) (<http://www.siam.org/books/fa06/>), where the implementation is based on the Gauss-Newton method.

In the examples, they show the registered images $T(\mathbf{u})$ and the corrected images $T_c(\mathbf{u})$. The latter are defined by the formula $T_c = (T(\mathbf{u}) - s)/e^{m_l}$ for ‘**Game**’ and ‘**CV**’, and $T_c = (T(\mathbf{u}) - s)/m$ for **JM**. In contrast, the final registered image for **MI** is just $T(\mathbf{u})$. The normalized correlation coefficient (NCC) between T_c and R and between $T(\mathbf{u})$ and R was used as evaluation metric to quantify the performance of the models and the comparison (the closer the NCC is to 1, the better is the alignment).

Example 1: MRI Images

Figure 3 shows an example of registering two MRI images using the complete models. The moving image T (synthetically enhanced) contains some bias field and varying illumination. The results of all models are displayed and they show that except ‘**MI**’, all models perform well in most parts of the image. However,

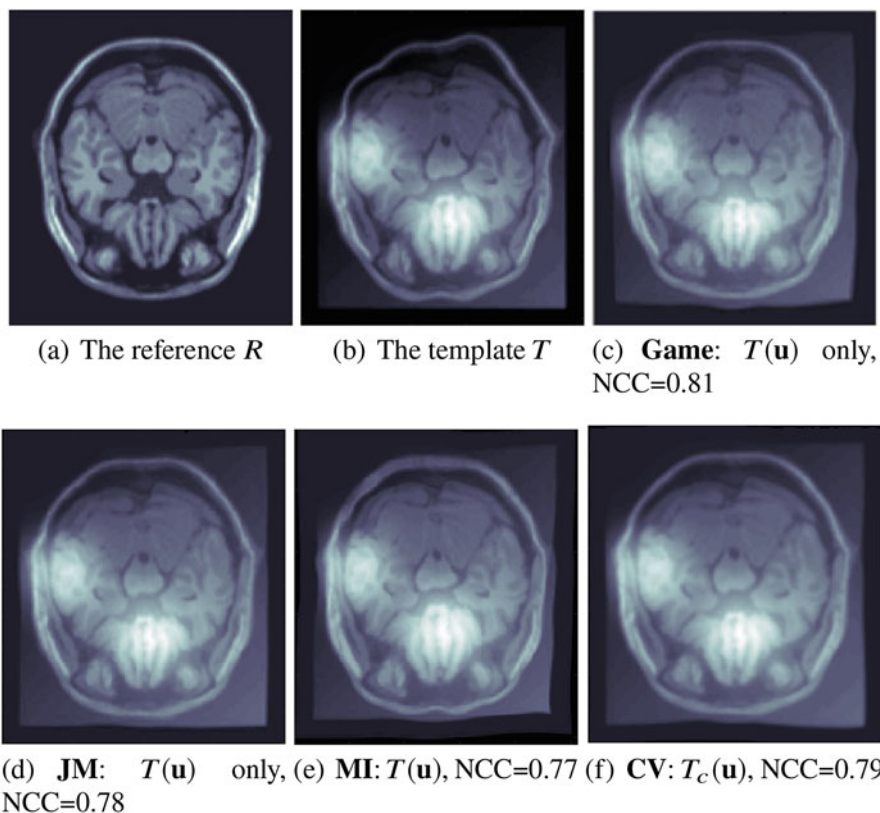


Fig. 3 Example 2: Comparison of 3 different models to register MRI T-1 and T-2 images. From this figure and Fig. 4, we see that **Game** model gives the best registration result. (a) The reference R (b) The template T (c) **Game**: $T(\mathbf{u})$ only, NCC=0.81 (d) **JM**: $T(\mathbf{u})$ only, NCC=0.78 (e) **MI**: $T(\mathbf{u})$, NCC=0.77 (f) **CV**: $T_c(\mathbf{u})$, NCC=0.79

in the middle of the images, the game model is the most advantageous and this can be observed in the zoomed details in Fig. 4. For the parameters tuning, the authors tested different values and they are tabulated in Table 1 which indicates the registration results for different parameters λ_i ($i = 1, \dots, 4$). The table shows that the game approach is stable.

Example 2: Application to Perfusion CT Registration

In Fig. 5, pair of CT and Perfusion CT lung images are registered. In the middle of the images T and R , there is a big difference because of the high contrast in T which makes inefficient the use of classical monomodal measures. The registered images using “**Game**”, “**CV**”, “**JM**”, and “**MI**” models are shown. We easily see that **Game** model gives a satisfactory result and the corrected part of the moving image is very similar to the middle part of the reference whereas the registration is not good.

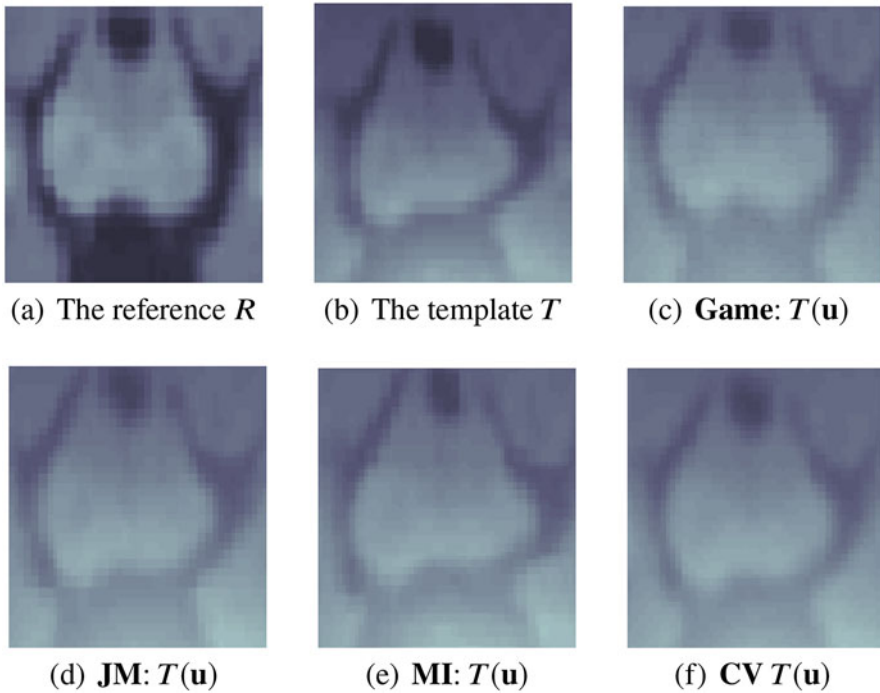


Fig. 4 Example 2: Compared zoom regions of 5 different models to register MRI T-1 and T-2 images. Again **Game** model is the best in solving the registration and the intensity correction jointly, whereas **JM** model cannot solve both problem jointly, only the image correction task is successful. (a) The reference R (b) The template T (c) **Game**: $T(\mathbf{u})$ (d) **JM**: $T(\mathbf{u})$ (e) **MI**: $T(\mathbf{u})$ (f) **CV** $T(\mathbf{u})$

Table 1 Parameters tuning for the pair of MRI images in Fig. 3 using **Game**. In the first column, we fix the parameters λ_3 and λ_4 and we vary the parameters λ_1 and λ_2 . In the third column, we vary λ_1 and λ_3 where λ_2 and λ_4 are fixed, whereas, in the last column, we vary λ_1 and λ_4 for fixed λ_2 and λ_3 . The NCC errors for the different values of parameters are comparable

Parameters			
λ_1	λ_2 NCC	λ_3 NCC	λ_4 NCC
100	05 NCC=0.77	0.5 NCC=0.78	01 NCC=0.78
150	15 NCC=0.79	01 NCC=0.80	05 NCC=0.80
200	20 NCC=0.80	05 NCC=0.80	20 NCC=0.79
250	40 NCC=0.79	10 NCC=0.77	50 NCC=0.78
	$\lambda_3 = 1$ and $\lambda_4 = 5$	$\lambda_2 = 20$ and $\lambda_4 = 5$	$\lambda_2 = 20$ and $\lambda_3 = 1$

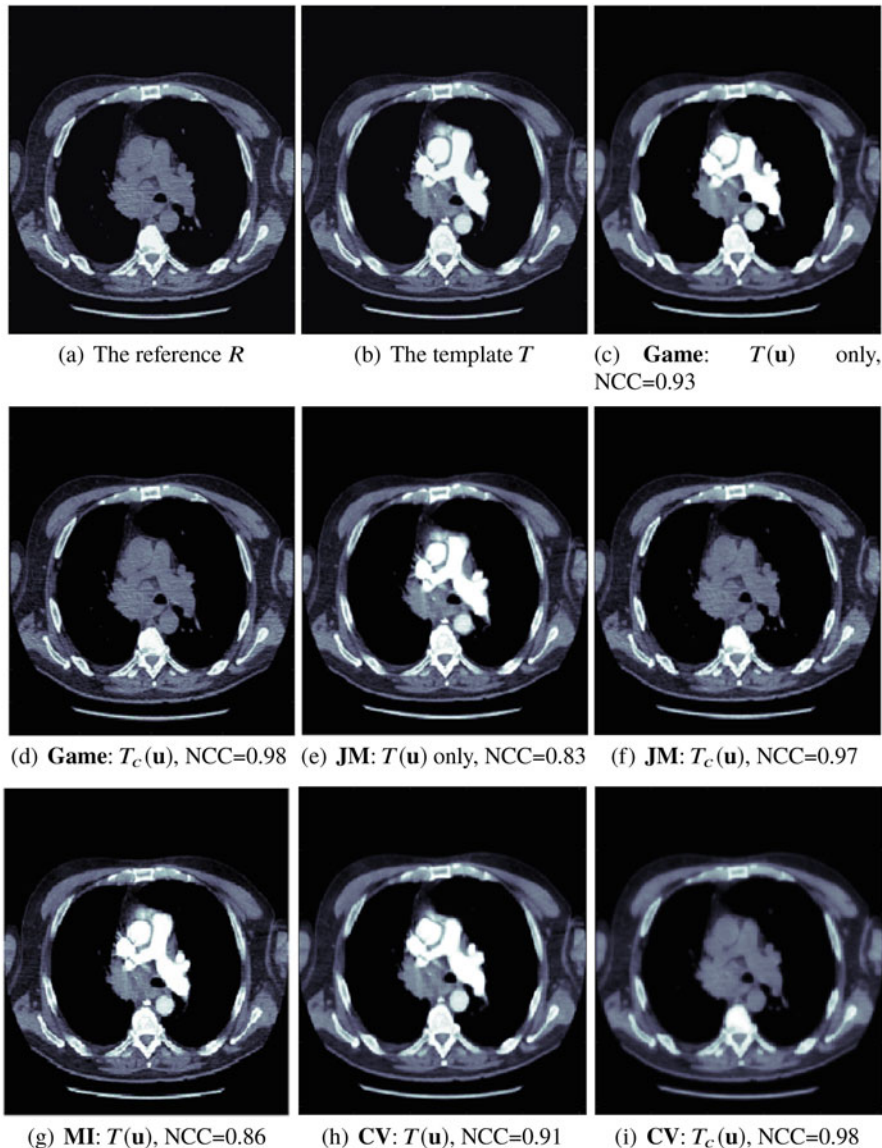


Fig. 5 Example 4: Comparison of 5 different models in registering CT and perfusion CT images. **Game** model performs the best. (a) The reference R (b) The template T (c) **Game**: $T(\mathbf{u})$ only, NCC=0.93 (d) **Game**: $T_c(\mathbf{u})$, NCC=0.98 (e) **JM**: $T(\mathbf{u})$ only, NCC=0.83 (f) **JM**: $T_c(\mathbf{u})$, NCC=0.97 (g) **MI**: $T(\mathbf{u})$, NCC=0.86 (h) **CV**: $T(\mathbf{u})$, NCC=0.91 (i) **CV**: $T_c(\mathbf{u})$, NCC=0.98

The result of both registration and correction is satisfactory and this underlines the performance of this model in solving both problems jointly and efficiently which is not the case for **CV**, **JM**, and **MI** as they only handle the correction task correctly and fail in registration. For this particular example, $T(\mathbf{u})$ is very useful as clinicians like to where the contrasts from perfusion CT (“artifacts”) would be located on the CT.

Game Models in Deep Learning

Game theory is a crucial element in building artificial intelligence (**AI**) models today for solving a multitasking models. In fact, a model which is designed to have multitasking property is the natural setting for Nash game formulation where the problem can effectively solved by considering different networks and different losses, one for each task. Good model would involve interaction between game theory and deep learning, that is, deep learning games. It is a very recent and interesting technique in artificial intelligence which uses neural networks and game strategies. Game environments and models are increasingly becoming popular training mechanisms for machine learning such as generative adversarial networks (Goodfellow et al. 2014), which have become one of the most successful frameworks for unsupervised generative modeling. Game theory is also recently used in reinforcing learning (Sutton et al. 2018) where various agents in the model compete against each other to improve the overall behavior. These both approaches represent the most recent powerful game models in artificial intelligence and have been used in different challenges and applications. In the sequel, we discuss the generative adversarial networks approach and its application in some image processing problems.

Generative Adversarial Networks (GANs)

Generative Adversarial modeling is a particular case of deep learning models which is based on the competition between two networks, pitting one against the other (thus the “adversarial”). Originally, it was developed for the image generation task from random samples (Goodfellow et al. 2014). It has progressed remarkably with the advent of convolution neural networks (CNNs) and is widely used for various imaging problems, mainly in the unsupervised learning context.

Generative vs Discriminative Algorithms

The generative adversarial models are based on the competition of two neural blocks, the discriminator and the generator. The generator is a convolutional neural network designed to create new instances of an object. The discriminator, on the other hand, is a “deconvolutional” neural network that determines the authenticity of the object and whether or not it is part of a set of true data. In terms of optimization, a backpropagation is used to make sure that the parameters in both networks are optimized by minimizing and/or maximizing a specific losses between

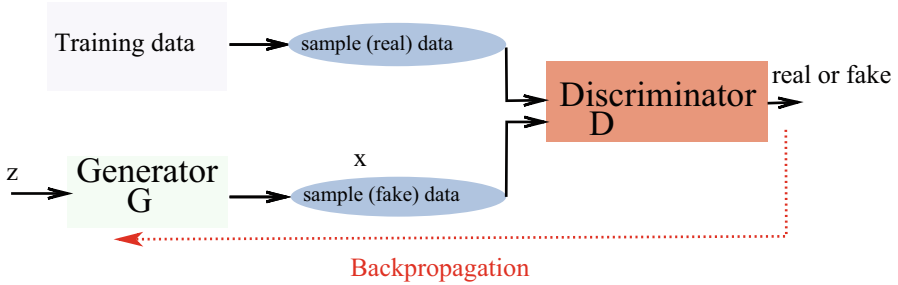


Fig. 6 The architecture of GAN models

true and generated data. They are trained in an adversarial and iterative manner until convergence is achieved when both are satisfied, that is, equilibrium situation. The illustration in Fig. 6 gives a rough idea on the work-flow of the generator and discriminator in the Generative Adversarial Networks.

Theory and Numerics

GANs models are an infinite zero-sum minmax game where Nash equilibrium (NE) is considered as a saddle point and the existence result is not straightforward. The existence of saddle points, equivalently a Nash equilibria, in infinite action games requires some “strong” properties like convexity and concavity of the loss functions, which is not always true as these losses are mostly nonconvex w.r.t networks are the weights of the network.

In various studies, existence was considered only for local Nash equilibrium or for Mixed Nash equilibrium (MNE), that is, with respect to probability distributions.

In practice, the training of GANs is considered as a tricky matter. In fact, reaching a Nash equilibrium for GANs through an optimization algorithm can be difficult to prove theoretically. Empirically, it has been observed that common algorithms, such as Stochastic Gradient Descent (SGD), lead to unstable training. Some studies on the convergence behaviors of gradient based training have been evolving throughout the years. The local convergence behavior has been studied in Nagarajan and Kolter (2017) and Heusel et al. (2017). The gradient-based optimization is proved to converge assuming that the discriminator and the generator is convex over the network parameters (Nowozin et al. 2016). However, even though research has been focused on understanding the training dynamics of GANs (Balduzzi et al. 2018; Gemp and Mahadevan 2018; Gidel et al. 2018a,b), a provably convergent algorithm for general GANs, even under reasonably strong assumptions, is still lacking.

GANs have been used for various image processing tasks with satisfactory results: images generation, image deblurring (Kupyn et al. 2018), image registration (Mahapatra et al. 2018), image classification, etc. In the sequel, we describe the GANs framework for the image generation problem which is a particular case of two-player game. We also give an example of using GANs for solving the image segmentation problem.

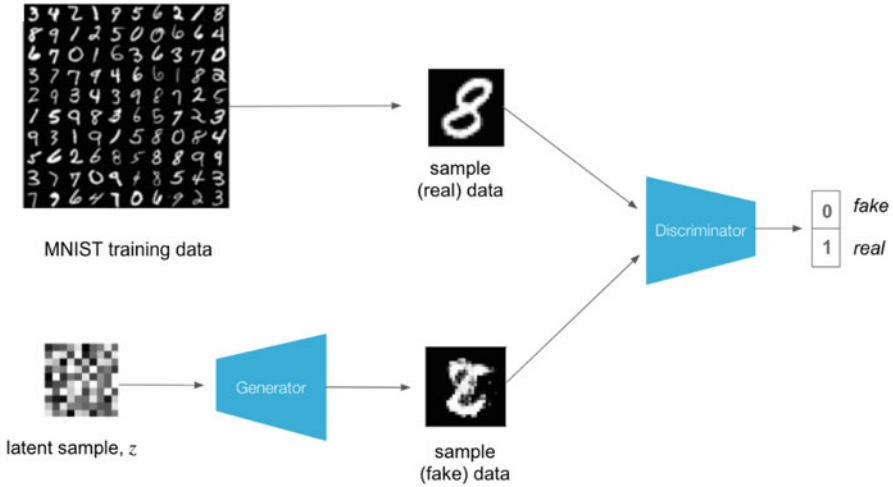


Fig. 7 The model architecture of GANs model for the image generation problem

GANs for Image Generation: A Two-Player Game

We consider the example of handwritten digits generation using generative adversarial network (Goodfellow et al. 2014) trained on the MNIST dataset (<http://yann.lecun.com/exdb/mnist/>). The aim is to be able to generate new digits from a random vector x of size 784. As mentioned, the GANs model is composed by the two networks, Generator \mathbf{G} and the discriminator \mathbf{D} . The generator takes the input random vector z (noise) and tries to generate a 28×28 image which is intended to be very close to the original images of MNIST dataset. Whereas the discriminator \mathbf{D} takes generated images by \mathbf{G} and tries to discriminate between them and real data. It is a binary classification network which turns the probability that the generated image by \mathbf{G} belongs to real dataset, that is, a class 1 means that it is real and 0 means fake (Fig. 7). Theoretically, GANs is a game model which is designed to compete the two networks \mathbf{G} and \mathbf{D} by solving the following min-max problem

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{J}(\mathbf{D}, \mathbf{G}) = \mathbf{E}_{x \sim p_{\text{data}}(x)} \log[\mathbf{D}(x)] + \mathbf{E}_{z \sim p_{\text{data}}(z)} \log[(1 - \mathbf{D}(\mathbf{G}(z)))] \quad (23)$$

where $\mathbf{E}_{x \sim p_{\text{data}}(x)}$ is the expected value over all real data instances. It is easy to prove the existence of Nash equilibrium for this model as it is two-player zero-sum minimax game. However, the main challenge in GANs is the training as finding a Nash equilibrium is not straightforward. The model is trained in alternating way; the \mathbf{D} -problem consists of solving the maximization problem

$$\min_{\mathbf{G}} \mathcal{J}(\mathbf{D}, \mathbf{G}) = \mathbf{E}_{x \sim p_{\text{data}}(x)} \log[\mathbf{D}(x)] + \mathbf{E}_{z \sim p_{\text{data}}(z)} \log[(1 - \mathbf{D}(\mathbf{G}(z)))] \quad (24)$$

where the first allows to recognize real images, whereas the second helps to recognize fake ones. The **G**-problem consists in solving the minimization problem

$$\min_{\mathbf{G}} \mathcal{J}(\mathbf{D}, \mathbf{G}) = \mathbf{E}_{x \sim p_{\text{data}}(x)} \log[(1 - \mathbf{D}(\mathbf{G}(x)))] \quad (25)$$

The GANs training algorithm involves training both the discriminator and the generator nets in parallel. The algorithm used in the original 2014 paper by Goodfellow (Goodfellow et al. 2014) is summarized in the figure below:

Algorithm 3 Mini batch stochastic gradient descent training of generative adversarial nets

for number of training iterations **do**

for k steps **do**

- Sample mini batch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample mini batch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log \mathbf{D}(x^{(i)}) + \log \left(1 - \mathbf{D}(\mathbf{G}(z^{(i)})) \right) \right].$$

end for

- Sample mini batch of m noise samples $\{x^{(1)}, \dots, x^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log \left(1 - \mathbf{D}(\mathbf{G}(z^{(i)})) \right).$$

endfor

With iteration, Generator **G** gets stronger and stronger at generating the real images and the discriminator **D** also gets stronger and stronger at identifying which one is real, which one is fake.

Examples

Few examples of images created by GANs for MNIST dataset are given in Fig. 8.

GANs for Image Segmentation: A Two-Player Game

Several approaches for the image segmentation problem based on the GANs framework were proposed in Luc et al. (2016), Mahapatra et al. (2018), and Tanner et al. (2018). We describe here the proposed GANs model in Luc et al. (2016) for the particular case of semantic segmentation.

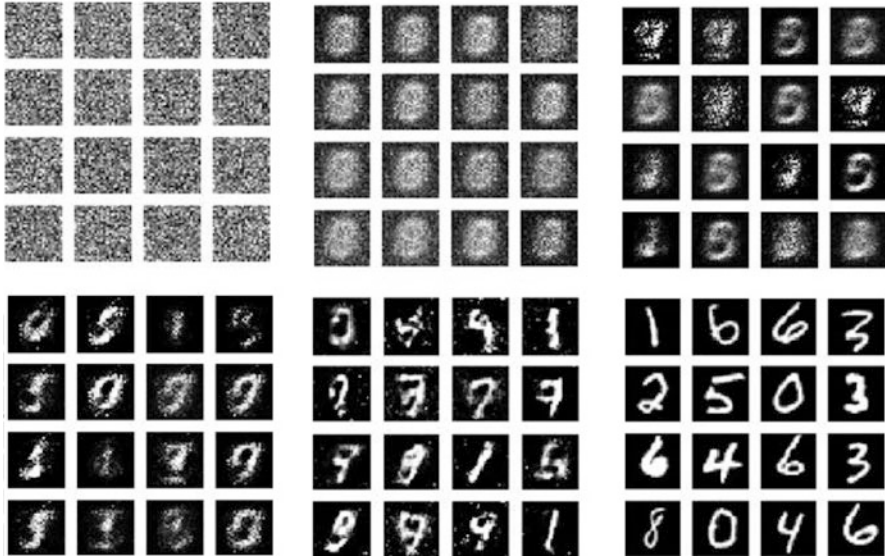


Fig. 8 Starting from random noise images, the generator gradually learns with iterations to emulate the features of the training dataset; it produces like-handwritten digits

The idea consists of using a generative adversarial networks (GANs) for RGB images segmentation where the trained network takes an RGB image x of size $H \times W \times 3$ as inputs and outputs the segmented image which is represented as a class label at each pixel location independently.

Generator and Discriminator

The generator is a segmentation CNN model which predicts a segmentation class from the input x by minimizing a segmentation loss. Its goal is to produce segmentation maps that are hard to distinguish from ground-truth ones for the adversarial model. The discriminator \mathbf{D} uses the generated maps by \mathbf{G} and compares it to the ground truth data in order to discriminate segmentation maps coming either from the ground truth or from the segmentation network. The model is summarized in Fig. 9.

Model Loss

The generator and the discriminator are trained together to optimize global loss function which is a weighted sum two terms. Given a data set of N training color images x_n of size $H \times W \times C$ and a corresponding label maps y_n , the authors defined a global loss as

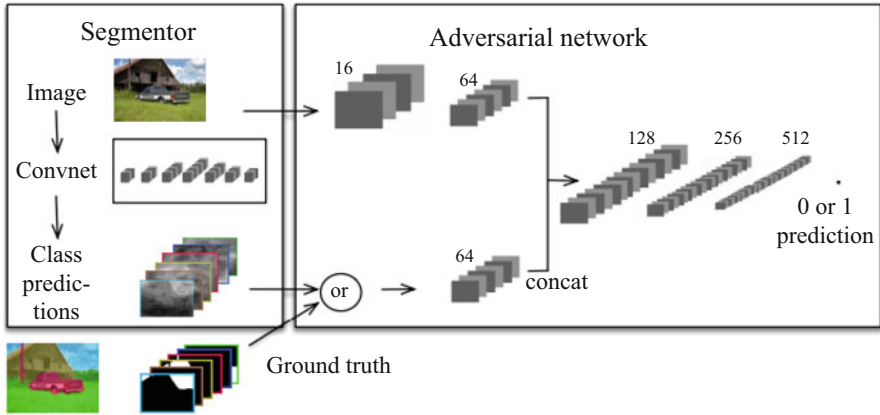


Fig. 9 Figure taken from Luc et al. (2016). Overview of the proposed approach. Left: segmentation net takes RGB image as input, and produces per-pixel class predictions. Right: Adversarial net takes label map as input and produces class label (1=ground truth, or 0=synthetic). Adversarial optionally also takes RGB image as input

$$\mathcal{J}(\mathbf{G}, \mathbf{D}) = \sum_{n=1}^N \ell_{mce}(\mathbf{G}(x_n), y_n) - \lambda [\ell_{bce}(\mathbf{D}(x_n, y_n), 1) + \ell_{bce}(\mathbf{D}(x_n, \mathbf{G}(x_n)), 0)] \tag{26}$$

where $\lambda = 10$ controls the contribution of the two terms, that is, the multi-class cross-entropy loss

$$\ell_{mce}(\mathbf{G}(x_n), y_n) = - \sum_{i=1}^{H \times W} \sum_{c=1}^C y_{ni} \log(\mathbf{G}(x_n)_c),$$

and the binary cross-entropy loss

$$\ell_{bce}(z_1, z_2) = - [z_2 \log z_1 + (1 - z_2) \log(1 - z_1)]$$

The term $\ell_{mce}(\mathbf{G}(x_n), y_n)$ denotes the multi-class cross-entropy loss for predictions $\mathbf{G}(x_n)$ and is a standard loss for semantic segmentation models. It encourages the segmentation model to predict the right class label at each pixel location independently. The Discriminator output $\mathbf{D}(x_n, y_n) \in [0, 1]$ represents the scalar probability of y_n being the ground truth label map of x_n , or being a fake map produced by Generator \mathbf{G} . The second part of the loss is for adversarial convolutional network and is large if the adversarial network can discriminate the generated segmentation map by Generator \mathbf{G} from ground-truth label maps.

Similar to all GANs models, this is a min-max game model where the full loss is minimized with respect to Generator \mathbf{G} of the segmentation, and maximized with

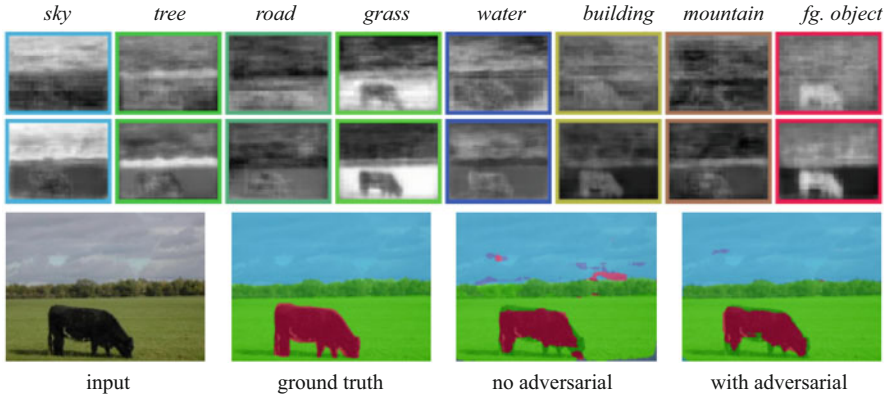


Fig. 10 Figure taken from Luc et al. (2016). Segmentations on Stanford Background. Class probabilities without (first row) and with (second row) adversarial training. In the last row the class labels are superimposed on the image

respect the adversarial model \mathbf{D} model.

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{J}(\mathbf{G}, \mathbf{D}). \tag{27}$$

Training

The model is trained in alternating way; the \mathbf{D} -problem consists in solving the minimization problem

$$\min_{\mathbf{D}} \ell_{bce}(\mathbf{D}(x_n, y_n), 1) + \ell_{bce}(\mathbf{D}(x_n, \mathbf{G}(x_n)), 0), \tag{28}$$

where the first allows to recognize real labels, whereas the second helps to recognize fake ones. The \mathbf{G} -problem consists in solving the minimization problem

$$\min_{\mathbf{G}} \sum \ell_{mce}(\mathbf{G}(x_n), y_n) - \lambda \ell_{bce}(\mathbf{D}(x_n, \mathbf{G}(x_n)), 0) \tag{29}$$

The GANs training algorithm involves training both the discriminator and the generator nets in parallel.

Example

The numerical example in Fig. 10 illustrate a comparison between the segmentation results using adversarial (GANs) and non-adversarial approaches. The results state that GANs approach clearly enhances the segmentation better than a classical deep learning approach, that is, non-adversarial.

Conclusion

Mathematical modeling of Vision and Imaging problems do naturally lead to formulations where antagonistic optimal decisions are aimed at. To this end, the recourse to a non-cooperative game paradigm seems to be very promising. The present chapter has addressed major imaging problematics, namely restoration versus segmentation and registration. Game theory can also be applied in various aspects of artificial intelligence, in particular for Adversarial Machine learning. Generative adversarial deep learning models have taken advantages from the game theory to reinvent generative models for different image processing problems. The authors have provided different illustrations of the strikingly efficient ability of game theory to address difficult concurrent optimization problems arising from these problematics.

References

- Aghajani, K., Manzuri, M.T., Yousefpour, R.: A robust image registration method based on total variation regularization under complex illumination changes. *Comput. Meth. Prog. Biomed.* **134**, 89–107 (2016)
- Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Alternating proximal algorithms for weakly coupled convex minimization problems. applications to dynamical games and pde's. *J. Convex Anal.* **15**(3), 485 (2008)
- Attouch, H., Soueiyat, M.: Augmented lagrangian and proximal alternating direction methods of multipliers in hilbert spaces. applications to games, pde's and control. *Pac. J. Optim.* **5**(1), 17–37 (2008)
- Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., Graepel, T.: The mechanics of n-player differentiable games. arXiv preprint arXiv:1802.05642 (2018)
- Bansal, R., Staib, L.H., Peterson, B.S.: Correcting nonuniformities in MRI intensities using entropy minimization based on an elastic model. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 78–86. Springer (2004)
- Benki, A., Habbal, A., Mathis, G., Beigneux, O.: Multicriteria shape design of an aerosol can. *J. Comput. Design Eng.* **11** (2015). <https://doi.org/10.1016/j.jcde.2015.03.003>. <https://hal.inria.fr/hal-01144269>
- Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: *Numerical Optimization: Theoretical and Practical Aspects*. Springer Science & Business Media (2006)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* **3**(1), 1–122 (2011)
- Chamekh, R., Habbal, A., Kallel, M., Zemzemi, N.: A nash game algorithm for the solution of coupled conductivity identification and data completion in cardiac electrophysiology. *Math. Modell. Nat. Phenom.* **14**(2), 15 (2019). <https://doi.org/10.1051/mmnp/2018059>. <https://hal.archives-ouvertes.fr/hal-01923819>
- Chang, H., Huang, W., Wu, C., Huang, S., Guan, C., Sekar, S., Bhakoo, K.K., Duan, Y.: A new variational method for bias correction and its applications to rodent brain extraction. *IEEE Trans. Med. Imaging* **36**(3), 721–733 (2017)
- Chen, K., Lui, L.M., Modersitzki, J.: Image and surface registration. In: *Handbook of Numerical Analysis – Processing, Analyzing and Learning of Images, Shapes, and Forms*, vol. 20. Elsevier (2019)

- Chen, Y., Ye, X.: Inverse consistent deformable image registration. In: *The Legacy of Alladi Ramakrishnan in the Mathematical Sciences*, pp. 419–440. Springer (2010)
- Christensen, G.E., Song, J.H., Lu, W., ElNaqa, I., Low, D.A.: Tracking lung tissue motion and expansion/compression with inverse consistent image registration and spirometry. *Med. Phys.* **34**, 2155–2163 (2007)
- Chumchob, N., Chen, K.: Improved variational image registration model and a fast algorithm for its numerical approximation. *Numer. Meth. Partial Differen. Equations* **28**(6), 1966–1995 (2012)
- Mumford, D.J.S.: Optimal approximations by piecewise smooth functions and variational problems. *Commun. Pure Appl. Math.* **42**, 577–685 (1989)
- Desideri, J.A., Duvigneau, R., Habbal, A.: Multiobjective design optimization using nash Games. In: M. Vasile, V.M. Becerra (eds.) *Computational Intelligence in the Aerospace Sciences, Progress in Astronautics and Aeronautics*. American Institute of Aeronautics and Astronautics (AIAA) (2014). <https://hal.inria.fr/hal-00923584>
- Duan, Y., Chang, H., Huang, W., Zhou, J., Lu, Z., Wu, C.: The $L_{\{0\}}$ regularized mumford–shah model for bias correction and segmentation of medical images. *IEEE Trans. Image Process.* **24**(11), 3927–3938 (2015)
- Ebrahimi, M., Martel, A.L.: A general pde-framework for registration of contrast enhanced images. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 811–819. Springer (2009)
- Fischer, B., Modersitzki, J.: Fast diffusion registration. *Contemp. Math.* **313**, 117–12 (2002)
- Friedman, A.: Stochastic differential games. *J. Differen. Equ.* **11**(1), 79–108 (1972)
- Gemp, I., Mahadevan, S.: Global convergence to the equilibrium of gans using variational inequalities. *arXiv preprint arXiv:1808.01531* (2018)
- Ghaffari, A., Fatemizadeh, E.: Image registration based on low rank matrix: Rank-regularized ssd. *IEEE Trans. Med. Imaging* **37**(1), 138–150 (2018)
- Gibbons, R.S.: *Game Theory for Applied Economists*. Princeton University Press (1992)
- Gidel, G., Berard, H., Vignoud, G., Vincent, P., Lacoste-Julien, S.: A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551* (2018a)
- Gidel, G., Hemmat, R.A., Pezeshki, M., Lepriol, R., Huang, G., Lacoste-Julien, S., Mitliagkas, I.: Negative momentum for improved game dynamics. *arXiv preprint arXiv:1807.04740* (2018b)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
- Habbal, A.: A topology Nash game for tumoral antiangiogenesis. *Struct. Multidiscip. Optim.* **30**(5), 404–412 (2005)
- Habbal, A., Kallel, M.: Neumann-Dirichlet nash strategies for the solution of elliptic cauchy problems. *SIAM J. Control. Optim.* **51**(5), 4066–4083 (2013). <https://hal.inria.fr/hal-00923574>
- Habbal, A., Kallel, M., Ouni, M.: Nash strategies for the inverse inclusion Cauchy-Stokes problem. *Inverse Prob. Imag.* **13**(4), 36 (2019). <https://doi.org/10.3934/ipi.2019038>. <https://hal.inria.fr/hal-01945094>
- Habbal, A., Petersson, J., Thellner, M.: Multidisciplinary topology optimization solved as a Nash game. *Int. J. Numer. Meth. Engng* **61**, 949–963 (2004)
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: *Advances in Neural Information Processing Systems*, pp. 6626–6637 (2017)
- Hu, J., Wellman, M.P.: Nash q-learning for general-sum stochastic games. *J. Mach. Learn. Res.* **4**(Nov), 1039–1069 (2003)
- Isaacs, R.: *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Courier Corporation (1999)
- Kallel, M., Aboulaich, R., Habbal, A., Moakher, M.: A nash-game approach to joint image restoration and segmentation. *Appl. Math. Model.* **38**(11-12), 3038–3053 (2014)
- Kim, Y., Tagare, H.D.: Intensity nonuniformity correction for brain mr images with known voxel classes. *SIAM J. Imag. Sci.* **7**(1), 528–557 (2014)

- Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)
- Lam, K.C., Lui, L.M.: Landmark- and intensity-based registration with large deformations via quasi-conformal maps. *SIAM J. Imag. Sci.* **7**(4), 2364–2392 (2014)
- Li, C., Gatenby, C., Wang, L., Gore, J.C.: A robust parametric method for bias field estimation and segmentation of mr images. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp. 218–223. IEEE (2009)
- Luc, P., Couprie, C., Chintala, S., Verbeek, J.: Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408 (2016)
- Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. *IEEE Trans. Med. Imaging* **16**(2), 187–198 (1997)
- Mahapatra, D., Antony, B., Sedai, S., Garnavi, R.: Deformable medical image registration using generative adversarial networks. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 1449–1453. IEEE (2018)
- Modersitzki, J.: FAIR: Flexible Algorithms for Image Registration. SIAM publications (2009)
- Modersitzki, J., Wirtz, S.: Combining homogenization and registration. In: International Workshop on Biomedical Image Registration, pp. 257–263. Springer (2006)
- Monderer, D., Shapley, L.S.: Potential games. *Games Econom. Behav.* **14**(1), 124–143 (1996)
- Nagarajan, V., Kolter, J.Z.: Gradient descent gan optimization is locally stable. In: Advances in Neural Information Processing Systems, pp. 5585–5595 (2017)
- Nash, J.: Equilibrium points in n -person games. *Proc. Natl. Acad. Sci. USA* **36**(1), 48–49 (1950)
- Nash, J.: Non-cooperative games. *Ann. Math.* 286–295 (1951)
- Neyman, A., Sorin, S.: Stochastic Games and Applications, vol. 570. Springer Science & Business Media (2003)
- Nishimura, R., Hayashi, S., Fukushima, M.: Robust nash equilibria in n -person non-cooperative games: Uniqueness and reformulation. *Pac. J. Optim.* **5**(2), 237–259 (2009)
- Nowozin, S., Cseke, B., Tomioka, R.: f-gan: Training generative neural samplers using variational divergence minimization. In: Advances in Neural Information Processing Systems, pp. 271–279 (2016)
- Park, C.R., Kim, K., Lee, Y.: Development of a bias field-based uniformity correction in magnetic resonance imaging with various standard pulse sequences. *Optik* **178**, 161–166 (2019)
- Rak, M., König, T., Tönnies, K.D., Walke, M., Ricke, J., Wybranski, C.: Joint deformable liver registration and bias field correction for mr-guided hdr brachytherapy. *Int. J. Comput. Assist. Radiol. Surg.* **12**(12), 2169–2180 (2017)
- Roy, S., Borzi, A., Habbal, A.: Pedestrian motion modeled by FP-constrained Nash games. *R. Soc. Open Sci.* (2017). <https://doi.org/10.1098/rsos.170648>. <https://hal.inria.fr/hal-01586678>
- Uryas'ev, S., Rubinstein, R.Y.: On relaxation algorithms in computation of noncooperative equilibria. *IEEE Trans. Autom. Control* **39**, 1263–1267 (1994)
- David, S., Hernández-Lerma Onésimo, G.: A survey of static and dynamic potential games. *Sci. China Math.* **59**(11), 2075–2102 (2016)
- Shapley, L.S.: Stochastic games. *Proc. Natl. Acad. Sci.* **39**(10), 1095–1100 (1953)
- Sutton, R.S., Barto, A.G., et al.: Introduction to Reinforcement Learning, 2nd edn. MIT Press Cambridge (2018)
- Tanner, C., Ozdemir, F., Profanter, R., Vishnevsky, V., Konukoglu, E., Goksel, O.: Generative adversarial networks for mr-ct deformable image registration. arXiv preprint arXiv:1807.07349 (2018)
- Theljani, A., Chen, K.: An augmented lagrangian method for solving a new variational model based on gradients similarity measures and high order regularization for multimodality registration. *Inv. Prob. Imag.* **13**, 309–335 (2019a)
- Theljani, A., Chen, K.: A nash game based variational model for joint image intensity correction and registration to deal with varying illumination. *Inv. Prob.* **36**, 034002 (2019b)

- Theljani, A., Chen, K.: A variational model for diffeomorphic multi-modal image registration using a new correlation like measure. submitted (2019c)
- Thompson, T., Chen, K.: An effective diffeomorphic model and its fast multigrid algorithm for registration of lung ct images improved optimization methods for image registration problems. *J. Comput. Meth. Appl. Math.* (2019)
- Thompson, T., Chen, K.: A more robust multigrid algorithm for diffusion type registration models. *J. Comput. Appl. Math.* **361**, 502–527 (2019)
- Van Leemput, K., Maes, F., Vandermeulen, D., Suetens, P.: Automated model-based bias field correction of mr images of the brain. *IEEE Trans. Med. Imaging* **18**(10), 885–896 (1999)
- Vovk, U., Pernus, F., Likar, B.: A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Trans. Med. Imaging* **26**(3), 405–421 (2007)
- Wang, L., Pan, C.: Nonrigid medical image registration with locally linear reconstruction. *Neurocomputing* **145**, 303–315 (2014)
- Zhang, D., Chen, K.: A novel diffeomorphic model for image registration and its algorithm. *J. Math. Imaging Vision* **60**, 1261–1283 (2018)
- Zhang, D., Chen, K.: 3D orientation-preserving variational models for accurate image registration. *SIAM J. Imaging Sci.* **13**, 1653–1691 (2020)
- Zhang, J., Chen, K., Yu, B.: A novel high-order functional based image registration model with inequality constraint. *Comput. Math. Appl.* **72**, 2887–2899 (2016)



First-Order Primal–Dual Methods for Nonsmooth Non-convex Optimization

18

Tuomo Valkonen

Contents

Introduction	708
Sample Problems	709
Outline	710
Bregman Divergences	711
Primal–Dual Proximal Splitting	713
Optimality Conditions and Proximal Points	714
Algorithm Formulation	715
Block Adaptation	716
Convergence Theory	718
A Fundamental Estimate	718
Ellipticity of the Bregman Divergences	720
Ellipticity for Block-Adapted Methods	723
Nonsmooth Second-Order Conditions	724
Second-Order Growth Conditions for Block-Adapted Methods	727
Convergence of Iterates	729
Convergence of Gaps in the Convex–Concave Setting	732
Inertial Terms	733
A Generalization of the Fundamental Theorem	733
Inertia (Almost) as Usually Understood	735
Improvements to the Basic Method Without Dual Affinity	737
Further Directions	741
Acceleration	741
Stochastic Methods	741
Alternative Bregman Divergences	741
Alternative Approaches	742
Functions on Manifolds and Hadamard Spaces	743
References	745

T. Valkonen (✉)

Center for Mathematical Modeling, Escuela Politécnica Nacional, Quito, Ecuador
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
e-mail: tuomo.valkonen@iki.fi

Abstract

We provide an overview of primal–dual algorithms for nonsmooth and non-convex-concave saddle-point problems. This flows around a new analysis of such methods, using Bregman divergences to formulate simplified conditions for convergence.

Keywords

Primal-dual · Nonsmooth · Nonconvex · Optimization · Inverse problems

Introduction

Interesting imaging problems can often be written in the general form

$$\min_{x \in X} \max_{y \in Y} F(x) + K(x, y) - G_*(y), \quad (\text{S})$$

where X and Y are Banach spaces, $K \in C^1(X, Y)$, and $F : X \rightarrow \overline{\mathbb{R}}$ and $G_* : Y \rightarrow \overline{\mathbb{R}}$ are convex, proper, lower semicontinuous functions with G_* the [preconjugate](#) of some $G : Y^* \rightarrow \overline{\mathbb{R}}$, meaning $G = (G_*)^*$. The functions F and G_* may be nonsmooth. In this chapter, we provide an overview of proximal-type primal–dual algorithms for this class of problems together with a simplified analysis, based on Bregman divergences.

➤ Notation, Conventions, and Basic Convex Analysis

As is standard in optimization, all vector/Banach/Hilbert spaces in this chapter are over the real field without it being explicitly mentioned. For basic definitions of convex analysis, such as the (pre)conjugate and the subdifferential, see the [glossary](#) at the end of the chapter or textbooks such as Hiriart-Urruty and Lemaréchal (2004), Rockafellar (1972), Clason and Valkonen (2020), and Ekeland and Temam (1999).

A common instance of (S) is when $K(x, y) = \langle Ax | y \rangle$ for a linear operator $A \in \mathbb{L}(X; Y^*)$ with $\langle \cdot | \cdot \rangle : Y^* \times Y \rightarrow \mathbb{R}$ denoting the dual product. Then (S) arises from writing G in terms of its (pre)conjugate G_* in

$$\min_{x \in X} F(x) + G(Ax). \quad (1)$$

We now discuss sample imaging and inverse problems of the types (S) and (1) and then outline our approach to solving them in the rest of the chapter.

Sample Problems

Optimization problems of the type (1) can effectively model linear inverse problems; typically one would attempt to minimize the sum of a data term and a regularizer

$$\min_{x \in X} \Phi(z - Tx) + G(Ax), \quad (2)$$

where

- $T : \mathbb{L}(X; \mathbb{R}^n)$ is a forward operator, mapping our unknown x into a finite number of measurements.
- Φ models noise v in the data $z \in \mathbb{R}^n$; for normal-distributed noise, $\Phi(z) = \frac{1}{2} \|z\|^2$.
- $G \circ A$ is a typically nonsmooth regularization term that models our prior assumptions on what a good solution to the ill-posed problem $z = Tx + v$ should be; in imaging, what “looks good.”

For conventional total variation regularization on a domain $\Omega \subset \mathbb{R}^m$, one would take $G(y^*) = \alpha \|y^*\|_{\mathcal{M}(\Omega; \mathbb{R}^m)}$ the Radon norm of the measure $y^* \in \mathcal{M}(\Omega; \mathbb{R}^m)$ weighted by the regularization parameter $\alpha > 0$ and $A = D \in \mathbb{L}(\text{BV}(\Omega); \mathcal{M}(\Omega; \mathbb{R}^m))$ the [distributional derivative](#) (Ambrosio et al. 2000). Simple examples of a *linear* forward operator T include:

- the identity for denoising (Rudin et al. 1992)
- a convolution operation for deblurring or deconvolution (Vogel and Oman 1998)
- a subsampling operator for inpainting (Shen and Chan 2002)
- the Fourier transform for magnetic resonance imaging (MRI) (Nishimura 1996; Lustig et al. 2007)
- the Radon transform for computational (CT) or positron emission tomography (PET) (Ollinger and Fessler 1997)

The last two examples would frequently be combined with subsampling for reconstruction from limited data.

In many important problems, T is, however, nonlinear:

- a pointwise application of $(r, \phi) \mapsto r e^{-i\phi}$ for phase and amplitude reconstruction for velocity-encoded magnetic resonance imaging (Valkonen 2014)
- a pointwise application of $u \mapsto s_0 - s e^{-(u,b)}$ to model the Stejskal–Tanner equation in diffusion tensor imaging (Valkonen 2014; Kingsley 2006)
- the solution operator of nonlinear partial differential equation (PDE) for several forms of tomography from magnetic and electric to acoustic and optical (Nishimura 1996; Ollinger and Fessler 1997; Arridge et al. 2011; Kuchment and Kunyansky 2011; Hunt 2014; Trucu et al. 2009; Uhlmann 2009; Lipponen et al. 2011)

In the last example, the PDE governs the physics of measurement, typically relating boundary measurements and excitations to interior data. The methods we study in this chapter are applied to electrical impedance tomography in Jauhainen et al. (2020) and Mazurenko et al. (2020).

How to fit a nonlinear forward operator T into the framework (S) that requires both F and G_* to be convex? If the noise model $\Phi : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is convex, proper, and lower semicontinuous, we can write (2) using the Fenchel conjugate Φ^* and $K_{TA}(x, (y_1, y_2)) := \langle z - T(x) | y_1 \rangle + \langle Ax | y_2 \rangle$ as

$$\min_{x \in X} \max_{(y_1, y_2) \in \mathbb{R}^n \times Y} K_{TA}(x, (y_1, y_2)) - \Phi^*(y_1) - G_*(y_2). \quad (3)$$

This is of the form (S) for the functions $\tilde{F} \equiv 0$ and $\tilde{G}_*(y_1, y_2) := \Phi^*(y_1) - G_*(y_2)$. Even for linear T , although (2) is readily of the form (1) and hence (S), this reformulation may allow expressing (2) in the form (S) with both \tilde{F} and \tilde{G}_* “prox-simple.” We will make this concept, important for the effective realization of algorithms, more precise in section “[Primal–Dual Proximal Splitting](#).”

Finally, fully general K in (S) was shown in Clason et al. (2020) to be useful for highly nonsmooth and non-convex problems, such as the Geman and Geman (1984). Indeed, the “0-function”

$$|t|_0 := \begin{cases} 0, & t = 0, \\ 1, & t \neq 0, \end{cases}$$

can be written as

$$|t|_0 = \sup_{s \in \mathbb{R}} \rho(st) \quad \text{for} \quad \rho(t) = 2t - t^2.$$

For the (anisotropic) Potts model, this is applied pixelwise on a discretized image gradient computed for an $n_1 \times n_2$ image by $\nabla_h : \mathbb{R}^{n_1 n_2} \rightarrow \mathbb{R}^{2 \times n_1 n_2}$ (Clason et al. 2020):

$$\min_{x \in \mathbb{R}^{n_1 n_2}} \max_{y \in \mathbb{R}^{2 \times n_1 n_2}} \frac{1}{2} \|b - x\|_2^2 + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \rho(\langle [\nabla_h x]_{ij}, y_{ij} \rangle), \quad (4)$$

where $b \in \mathbb{R}^{n_1 n_2}$ is the image to be segmented.

Outline

We introduce in section “[Primal–Dual Proximal Splitting](#)” methods for (S) inspired by the primal–dual proximal splitting (PDPS) of Chambolle and Pock (2011) and Pock et al. (2009) for bilinear K , commonly known as the Chambolle–Pock

method. We work in Banach spaces, as was done in Hohage and Homann (2014). To be able to define proximal-type methods in Banach spaces, in section “[Bregman Divergences](#),” we introduce and recall the crucial properties of the so-called Bregman divergences.

Our main reason for working with Bregman divergences is, however, not the generality of Banach spaces. Rather, they provide a powerful proof tool to deal with the general K in (S). This approach allows us in section “[Convergence Theory](#)” to significantly simplify and better explain the original convergence proofs and conditions of Chambolle and Pock (2011), Valkonen (2014), Clason et al. (2019), Clason et al. (2020), and Mazurenko et al. (2020). Without additional effort, they also allow us to present block-adapted methods like those in Valkonen and Pock (2017), Valkonen (2019), and Mazurenko et al. (2020).

Our overall approach and the internal organization of section “[Convergence Theory](#)” centers around the following three main ingredients of the convergence proof:

- (i) A **three-point identity**, satisfied by all Bregman divergences (shown in section “[Bregman Divergences](#)” and employed in section “[A Fundamental Estimate](#)”)
- (ii) **(Semi-)ellipticity** of the algorithm-defining Bregman divergences (concept defined in section “[Bregman Divergences](#),” specific Bregman divergence in section “[Primal–Dual Proximal Splitting](#),” and its ellipticity verified in sections “[Ellipticity of the Bregman Divergences](#),” and “[Ellipticity for Block-Adapted Methods](#)” through several examples)
- (iii) A **nonsmooth second-order growth** condition around a solution of (S) (treated in sections “[Nonsmooth Second-Order Conditions](#)” and “[Second-Order Growth Conditions for Block-Adapted Methods](#)”)

With these basic ingredients, we then prove convergence in sections “[Convergence of Iterates](#)” and “[Convergence of Gaps in the Convex-Concave Setting](#).” In the present overview, with focus on key concepts and aiming to avoid technical complications, we only cover, weak, strong, and linear convergence of iterates, and the convergence of gap functionals when K is convex-concave.

In section “[Inertial Terms](#)” we improve the basic method by adding dependencies to earlier iterates, a form of inertia. This is needed to develop an effective algorithm for K not affine in y , including the aforementioned formulation of the Potts segmentation model. We finish in section “[Further Directions](#)” with pointers to alternative methods and further extensions.

Bregman Divergences

The norm and inner product in a (real) Hilbert space X satisfy the three-point identity:

$$\langle x - y, x - z \rangle_X = \frac{1}{2} \|x - y\|_X^2 - \frac{1}{2} \|y - z\|_X^2 + \frac{1}{2} \|x - z\|_X^2 \quad (x, y, z \in X). \quad (5)$$

This is crucial for convergence proofs of optimization methods (Valkonen 2020), so we would like to have something similar in Banach spaces—or other more general spaces. Towards this end, we let $J : X \rightarrow \mathbb{R}$ be a Gâteaux-differentiable function¹. Then one can define the asymmetric Bregman divergence:

$$B_J(z, x) := J(z) - J(x) - \langle DJ(x)|z - x \rangle_X \quad (x, z \in X). \quad (6)$$

This function is non-negative *if and only if*² the generating function J is convex; it is not in general a true distance, as it can happen that $B_J(x, z) = 0$ although $x \neq z$.

Writing D_1 for the Gâteaux derivative with respect to the first parameter, we have

$$D_1 B_J(x, z) = DJ(z) - DJ(x). \quad (7)$$

Moreover, the Bregman divergence satisfies for any $\bar{x} \in X$ the three-point identity

$$\begin{aligned} \langle D_1 B_J(x, z)|x - \bar{x} \rangle_X &= \langle DJ(x) - DJ(z)|x - \bar{x} \rangle_X \\ &= B_J(\bar{x}, x) - B_J(\bar{x}, z) + B_J(x, z). \end{aligned} \quad (8)$$

Indeed, writing the right-hand side out, we have

$$\begin{aligned} B_J(\bar{x}, x) - B_J(\bar{x}, z) + B_J(x, z) &= [J(\bar{x}) - J(x) - \langle DJ(x)|\bar{x} - x \rangle_X] \\ &\quad - [J(\bar{x}) - J(z) - \langle DJ(z)|\bar{x} - z \rangle_X] \\ &\quad + [J(x) - J(z) - \langle DJ(z)|x - z \rangle_X], \end{aligned}$$

which immediately gives the three-point identity.

Example 1. In a Hilbert space X , the standard generating function $J = N_X := \frac{1}{2} \|\cdot\|_X^2$ yields $B_J(z, x) = \frac{1}{2} \|z - x\|_X^2$, so (8) recovers (5).

We will frequently require B_J to be non-negative or semi-elliptic ($\gamma = 0$) or elliptic ($\gamma > 0$) within some $\Omega \subset X$. These notions mean that

$$B_J(z, x) \geq \frac{\gamma}{2} \|z - x\|_X^2 \quad (x, z \in \Omega). \quad (9)$$

¹The differentiability assumption is for notational and presentational simplicity; otherwise we would need to write the Bregman divergence as $B_J^p(z, x) := J(z) - J(x) - \langle p|z - x \rangle_X$ for some subdifferential p of J and define explicit updates of this subdifferential in algorithms.

²For the entirely algebraic proof of the “only if,” see Hiriart-Urruty and Lemaréchal 2004, Theorem 4.1.1.

Equivalently, this defines J to be (γ -strongly) subdifferentiable within Ω . When $\Omega = X$, we simply call B_J (semi-)elliptic and J (γ -strongly) subdifferentiable³.

We will in section “**Inertial Terms**” also need a Cauchy inequality for Bregman divergences. We base this on strong subdifferentiability and the smoothness property (10) in the next lemma. The latter holding with $\Omega = X$ implies that DJ is L -Lipschitz and in Hilbert spaces is equivalent to this property; see Bauschke and Combettes (2017, Theorem 18.15) or Valkonen (2020, Appendix C).

Lemma 1. *Suppose $J : X \rightarrow \mathbb{R}$ is Gâteaux-differentiable and γ -strongly subdifferentiable within Ω and satisfies for some $L > 0$ the subdifferential smoothness*

$$\frac{1}{2L} \|DJ(x) - DJ(y)\|_{X^*}^2 \leq J(x) - J(y) - \langle DJ(y)|x - y \rangle \quad (x, y \in \Omega). \quad (10)$$

Then, for any $\alpha > 0$,

$$|\langle D_1 B_J(x, y)|z - x \rangle| \leq \frac{L}{\alpha} B_J(x, y) + \frac{\alpha}{\gamma} B_J(z, x) \quad (x, y, z \in \Omega).$$

Proof. By Cauchy’s inequality and (7),

$$|\langle D_1 B_J(x, y)|z - x \rangle| \leq \frac{1}{2\alpha} \|DJ(x) - DJ(y)\|_{X^*}^2 + \frac{\alpha}{2} \|z - x\|_X^2.$$

By the strong convexity, $\frac{\gamma}{2} \|z - x\|_X^2 \leq B_J(z, x)$, and by the smoothness property (10), $\frac{1}{2L} \|DJ(x) - DJ(y)\|_{X^*}^2 \leq B_J(x, y)$. Together these estimates yield the claim. \square

Primal–Dual Proximal Splitting

We now formulate a basic version of our primal–dual method. Later in section “**Inertial Terms**” we improve the algorithm to be more effective when K is not affine in y .

➤ Notation

Throughout the manuscript, we combine the primal and dual variables x and y into variables involving the letter u :

$$u = (x, y), \quad u^k = (x^k, y^k), \quad \hat{u} = (\hat{x}, \hat{y}), \quad \text{etc.}$$

³In Banach spaces strong subdifferentiability is implied by strong convexity, as defined without subdifferentials. In Hilbert spaces the two properties are equivalent.

Optimality Conditions and Proximal Points

We define the Lagrangian as

$$\mathcal{L}(x, y) := F(x) + K(x, y) - G_*(y).$$

A saddle point $\hat{u} = (\hat{x}, \hat{y})$ of the problem (S) satisfies, by definition

$$\mathcal{L}(\hat{x}, y) \leq \mathcal{L}(\hat{x}, \hat{y}) \leq \mathcal{L}(x, \hat{y}) \quad \text{for all } u = (x, y) \in X \times Y.$$

Writing $D_x K$ and $D_y K$ for the Gâteaux derivatives of K with respect to the two variables, if K is convex-concave, basic results in convex analysis (Ekeland and Temam 1999; Bauschke and Combettes 2017) show that

$$-D_x K(\hat{x}, \hat{y}) \in \partial F(\hat{x}) \quad \text{and} \quad D_y K(\hat{x}, \hat{y}) \in \partial G_*(\hat{y}) \tag{11}$$

is necessary and sufficient for \hat{u} to be saddle point. If K is C^1 , the theory of generalized subdifferentials of Clarke (1990) still indicates⁴ the necessity of (11).

We can alternatively write (11) as

$$0 \in H(\hat{u}) := \begin{pmatrix} \partial F(\hat{x}) + D_x K(\hat{x}, \hat{y}) \\ \partial G_*(\hat{y}) - D_y K(\hat{x}, \hat{y}) \end{pmatrix}. \tag{12}$$

If X and Y were Hilbert spaces, we could in principle use the classical proximal point method (Minty 1961; Rockafellar 1976) to solve (12): given step length parameters $\tau_k > 0$, iteratively solve u^{k+1} from

$$0 \in H(u^{k+1}) + \tau_k^{-1}(u^{k+1} - u^k). \tag{13}$$

If K were bilinear, H would be a so-called monotone operator and convergence of iterates would follow from Rockafellar (1976). In practice the steps of the method are too expensive to realize as the primal and dual iterates x^{k+1} and y^{k+1} are coupled: generally, one cannot solve one before the other.

Fortunately, the iterates can be decoupled by introducing a preconditioner that switches $D_x K(x^{k+1}, y^{k+1})$ on the first line of $H(u^{k+1})$ to $D_x K(x^k, y^k)$. This gives rise to the primal–dual proximal splitting (PDPS), introduced in Chambolle and Pock (2011) and Pock et al. (2009) for bilinear $K(x, y) = \langle Ax|y \rangle$. That the PDPS is actually a preconditioned proximal point method was first observed in He and Yuan (2012). In the following, we describe its extension from Valkonen (2014) and Clason et al. (2019, 2020) to general K and the general problem (S). To simplify the proofs and concepts in them, we work with Bregman divergences, at no cost in Banach spaces.

⁴The Fermat rule $0 \in \partial_C[F + K(\cdot, \hat{y})](\hat{x})$ holds. Since F is convex and $K(\cdot, \hat{y})$ is C^1 , \hat{x} is a regular point of both, so also the subdifferential sum rule holds. We argue $G_* + K(\hat{y}, \cdot)$ similarly.

Algorithm Formulation

Given Gâteaux-differentiable functions $J_X : X \rightarrow \overline{\mathbb{R}}$ and $J_Y : Y \rightarrow \overline{\mathbb{R}}$ with the corresponding Bregman divergences $B_X := B_{J_X}$ and $B_Y := B_{J_Y}$, we define

$$J^0(x, y) := J_X(x) + J_Y(y) - K(x, y). \quad (14)$$

Introducing the short-hand notation $B^0 := B_{J^0}$, we propose to solve (12) through the iterative solution of

$$0 \in H(u^{k+1}) + D_1 B^0(u^{k+1}, u^k) \quad (15)$$

for u^{k+1} . Inserting (12) and (7) for $J = J^0$ as defined in (14), we expand and rearrange this implicitly defined method as:

Primal–dual Bregman-proximal splitting (PDBS)

Iteratively over $k \in \mathbb{N}$, solve for x^{k+1} and y^{k+1} :

$$\begin{aligned} DJ_X(x^k) - D_x K(x^k, y^k) &\in DJ_X(x^{k+1}) + \partial F(x^{k+1}) \quad \text{and} \\ DJ_Y(y^k) - D_y K(x^k, y^k) &\in DJ_Y(y^{k+1}) + \partial G_*(y^{k+1}) - 2D_y K(x^{k+1}, y^{k+1}). \end{aligned} \quad (16)$$

We readily obtain x^{k+1} if the inverse of $DJ_X + \tau \partial F$ has an analytical closed-form expression. In this case we say that F is prox-simple with respect to J_X . For y^{k+1} , the same is true if K is affine in y and G_* is prox-simple with respect to J_Y . If, however, K is not affine in y , it is practically unlikely that $\partial G_* - 2D_y K(x^{k+1}, \cdot)$ would be prox-simple. We will therefore improve the method for general K in section “[Inertial Terms](#),” after first studying fundamental ideas behind convergence proofs in the following section “[Convergence Theory](#).”

If X and Y are Hilbert spaces with $J_X = \tau^{-1}N_X$ and $J_Y = \sigma^{-1}N_Y$, the standard generating functions divided by some step length parameters $\tau, \sigma > 0$, (16) becomes

Primal–dual proximal splitting (PDPS)

Iterate over $k \in \mathbb{N}$:

$$\begin{aligned} x^{k+1} &:= \text{prox}_{\tau F}(x^k - \tau \nabla_x K(x^k, y^k)), \\ y^{k+1} &:= \text{prox}_{\sigma[G_* - 2K(x^{k+1}, \cdot)]}(y^k - \sigma \nabla_y K(x^k, y^k)). \end{aligned} \quad (17)$$

The proximal map is defined as

$$\text{prox}_{\tau F}(x) := (I + \tau \partial F)^{-1}(x) = \arg \min_{\tilde{x} \in X} \left(\tau F(\tilde{x}) + \frac{1}{2} \|\tilde{x} - x\|_X^2 \right).$$

When this map has an analytical closed-form expression, we say that F is prox-simple (without reference to J_X). In finite dimensions, several worked out proximal maps may be found online (Chierchia et al. 2019) or in the book (Beck 2017). Some extend directly to Hilbert spaces or by superposition to L^2 .

Remark 1. For K affine in y , i.e., $K(x, y) = \langle A(x)|y \rangle$ for some differentiable $A : X \rightarrow Y^*$, the dual update of (17) reduces to

$$\begin{aligned} y^{k+1} &= \text{prox}_{\sigma G_*}(y^k + \sigma[2\nabla_y K(x^{k+1}, y^k) - \nabla_y K(x^k, y^k)]) \\ &= \text{prox}_{\sigma G_*}(y^k + \sigma[2\nabla A(x^{k+1}) - \nabla A(x^k)]). \end{aligned}$$

This corresponds to the “linearized” variant of the NL-PDPS of Valkonen (2014). The “exact” variant, studied in further detail in Clason et al. (2019), updates

$$y^{k+1} := \text{prox}_{\sigma G_*}(y^k + \sigma \nabla_y K(2x^{k+1} - x^k, y^k)).$$

If K is bilinear, the two variants are the exactly same PDPS of Chambolle and Pock (2011). For K not affine in y , the method is neither the generalized PDPS of Clason et al. (2020) nor the version for convex-concave K from Hamedani and Aybat (2018).

Block Adaptation

We now derive a version of the PDBS (16) adapted to the structure of

$$F(x) = \sum_{j=1}^m F_j(x_j) \quad \text{and} \quad G_*(y) = \sum_{\ell=1}^n G_{\ell_*}(y_\ell),$$

where $x = (x_1, \dots, x_m)$ and $y = (y_1, \dots, y_n)$ in the (for simplicity) Hilbert spaces $X = \prod_{j=1}^m X_j$ and $Y = \prod_{\ell=1}^n Y_\ell$, and $F_j : X_j \rightarrow \mathbb{R}$ and $G_{\ell_*} : Y_\ell \rightarrow \mathbb{R}$ are convex, proper, and lower semicontinuous.

For some “blockwise” step length parameters $\tau_j, \sigma_\ell > 0$, we take

$$J_X(x) = \sum_{j=1}^m \tau_j^{-1} N_{X_j}(x_j) \quad \text{and} \quad J_Y(y) = \sum_{\ell=1}^n \sigma_\ell^{-1} N_{Y_\ell}(y_\ell)$$

If K is now affine in y , observing Remark 1, (16) readily transforms into:

Block-adapted PDPS for K affine in y

Iteratively over $k \in \mathbb{N}$, for all $j = 1, \dots, m$ and $\ell = 1, \dots, n$, update:

$$\begin{aligned} x_j^{k+1} &:= \text{prox}_{\tau_j F_j}(x_j^k - \tau_j \nabla_{x_j} K(x^k, y^k)), \\ y_\ell^{k+1} &:= \text{prox}_{\sigma_\ell G_{\ell*}}(y_\ell^k + \sigma_\ell [2\nabla_{y_\ell} K(x^{k+1}, y^k) - \nabla_{y_\ell} K(x^k, y^k)]). \end{aligned} \quad (18)$$

The idea is that the blockwise step length parameters adapt the algorithm to the structure of the problem. We will return their choices in the examples of section “[Ellipticity for Block-Adapted Methods](#).”

➤ Performance gains

Correct adaptation of the blockwise step length parameters to the specific problem structure can yield significant performance gains compared to not exploiting the block structure (Pock and Chambolle 2011; Jauhainen et al. 2020; Mazurenko et al. 2020).

Remark 2. For bilinear K , (18) is the “diagonally preconditioned” method of Pock and Chambolle (2011), or an unaccelerated non-stochastic variant of the methods in Valkonen (2019). For K affine in y , (18) differs from the methods in Mazurenko et al. (2020) by placing the over-relaxation in the dual step outside K , compare Remark 1.

Recall the saddle-point formulation (3) for inverse problems with nonlinear forward operators. We can now adapt step lengths to the constituent dual blocks.

Example 2. Let $A_1 \in C^1(X; Y_1^*)$ and $A_2 \in \mathbb{L}(X; Y_2^*)$, and suppose the convex functions $G_1 : Y_1^* \rightarrow \overline{\mathbb{R}}$ and $G_2 : Y_2^* \rightarrow \overline{\mathbb{R}}$ have the preconjuguates G_{1*} and G_{2*} . Then we can write the problem

$$\min_{x \in X} G_1(A_1(x)) + G_2(A_2x) + F(x).$$

in the form (S) with $G_*(y_1, y_2) = G_{1*}(y_1) + G_{2*}(y_2)$ and $K(x, y) = \langle A_1(x) | y_1 \rangle + \langle A_2x | y_2 \rangle$. The algorithm (18) specializes as

$$\begin{aligned} x^{k+1} &:= \text{prox}_{\tau F}(x^k - \tau[\nabla A_1(x^k)^* y_1 + A_2^* y_2]), \\ y_1^{k+1} &:= \text{prox}_{\sigma_1 G_{1*}}(y_1^k + \sigma_1 [2A_1(x^{k+1}) - A_1(x^k)]), \\ y_2^{k+1} &:= \text{prox}_{\sigma_2 G_{2*}}(y_2^k + \sigma_2 [A_2(2x^{k+1} - x^k)]) \end{aligned}$$

for some step length parameters $\tau, \sigma_1, \sigma_2 > 0$. We return to their choices and the local neighborhood of convergence in Examples 8 and 17 after developing the necessary convergence theory.

Convergence Theory

We now seek to understand when the basic version (15) of the PDBS convergences. The organization of this section centers around the [three main ingredients](#) of the convergence proof, as discussed in the Introduction:

- (i) the three-point identity (8) employed in the general-purpose estimate of section [“A Fundamental Estimate”](#)
- (ii) the (semi-)ellipticity of the algorithm-generating Bregman divergences B_{J_0} for J^0 as in (14), verified for several examples in sections [“Ellipticity of the Bregman Divergences”](#) and [“Ellipticity for Block-Adapted Methods”](#)
- (iii) a second-order growth condition on (S), verified for several examples in sections [“Nonsmooth Second-Order Conditions”](#) and [“Second-Order Growth Conditions for Block-Adapted Methods”](#)

With these basic ingredients, we then prove various convergence results in sections [“Convergence of Iterates”](#) and [“Convergence of Gaps in the Convex-Concave Setting”](#). The usefulness of both (ii) and (iii) will become apparent from the fundamental estimates and examples of the next section [“A Fundamental Estimate.”](#)

A Fundamental Estimate

We start with a simple estimate applicable to general methods of the form

$$0 \in H(u^{k+1}) + D_1 B(u^{k+1}, u^k) \tag{BP}$$

for some set-valued $H : U \rightrightarrows U^*$ and a Bregman divergence $B := B_J$ generated by some Gâteaux-differentiable $J : U \rightarrow \mathbb{R}$. We analyze (BP) following the “testing” ideas introduced in Valkonen (2020), extending them to the Bregman–Banach space setting, however in a simplified constant-metric setting that cannot model accelerated methods. The generic gap functional $\mathcal{G}(u^{k+1}, \bar{u})$ in the next result models any function value differences available from H . Its non-negativity will provide the basis for the aforementioned second-order growth conditions of sections [“Nonsmooth Second-Order Conditions”](#) and [“Second-Order Growth Conditions for Block-Adapted Methods.”](#) We provide an example and interpretation after the theorem.

Theorem 1. *On a Banach space U , let $H : U \rightrightarrows U^*$, and let $B := B_J$ be generated by a Gâteaux-differentiable $J : U \rightarrow \mathbb{R}$. Suppose (BP) is solvable for $\{u^{k+1}\}_{k \in \mathbb{N}}$ given an initial iterate $u^0 \in U$. Let $N \geq 1$. If for all $k = 0, \dots, N - 1$, for some $\bar{u} \in U$ and $\mathcal{G}(u^{k+1}, \bar{u}) \in \mathbb{R}$, the fundamental condition*

$$\langle h^{k+1} | u^{k+1} - \bar{u} \rangle \geq \mathcal{G}(u^{k+1}, \bar{u}) \quad (h^{k+1} \in H(u^{k+1})) \quad (\text{C})$$

holds, then so do the quantitative Δ -Féjer monotonicity

$$B(\bar{u}, u^{k+1}) + B(u^{k+1}, u^k) + \mathcal{G}(u^{k+1}, \bar{u}) \leq B(\bar{u}, u^k) \quad (\text{F})$$

and the descent inequality

$$B(\bar{u}, u^N) + \sum_{k=0}^{N-1} B(u^{k+1}, u^k) + \sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \bar{u}) \leq B(\bar{u}, u^0). \quad (\text{D})$$

Proof. We can write (BP) as

$$0 = h^{k+1} + D_1 B(u^{k+1}, u^k) \quad \text{for some } h^{k+1} \in H(u^{k+1}). \quad (\text{19})$$

Testing (19) by applying $\langle \cdot | u^{k+1} - \bar{u} \rangle$, we obtain

$$0 = \langle h^{k+1} + D_1 B(u^{k+1}, u^k) | u^{k+1} - \bar{u} \rangle.$$

We use the three-point identity (8) to transform this into

$$B(\bar{u}, u^k) = \langle h^{k+1} | u^{k+1} - \bar{u} \rangle + B(\bar{u}, u^{k+1}) + B(u^{k+1}, u^k).$$

Inserting (C), we obtain (F). Summing the latter over $k = 0, \dots, N - 1$ yields (D). \square

Example 3. If $H = \partial F$ for a convex function F , then by the definition of the convex subdifferential, (C) holds with the gap functional

$$\mathcal{G}(u, \bar{u}) = F(u) - F(\bar{u}).$$

If we take \bar{u} is a minimizer of F , then the gap functional is non-negative and indeed positive if u is also not minimizer. This is why it is called a gap functional.

Consider then for some step length parameter $\tau > 0$ the proximal point method (13) in a Hilbert space X , that is, taking $B = \tau^{-1}N_X$

$$u^{k+1} := \text{prox}_{\tau F}(x^k), \quad \text{equivalently } 0 \in \partial F(u^{k+1}) + \tau(u^{k+1} - u^k).$$

Then (D) reads

$$\frac{1}{2\tau} \|u^N - \bar{u}\|_X^2 + \sum_{k=0}^{N-1} \frac{1}{2} \|u^{k+1} - u^k\|_X^2 + \sum_{k=0}^{N-1} \tau (F(u^{k+1}) - F(\bar{u})) \leq \frac{1}{2} \|\bar{u} - u^0\|_X^2. \tag{20}$$

With \bar{u} a minimizer, this clearly forces $F(u^N) \searrow F(\bar{u})$ as $N \nearrow \infty$, suggesting why we call (D) the “descent inequality.”

If our problem is non-convex, then we can try to locally ensure second-order growth by imposing $\mathcal{G}(u^{k+1}, \bar{u}) \geq 0$. Verifying this for the PDBS will be the topic of sections “Nonsmooth Second-Order Conditions” and “Second-Order Growth Conditions for Block-Adapted Methods.” If B is not given by the standard generating function N_X on a Hilbert spaces X , then to get from (D) an estimate like (20) on norms, we can assume the ellipticity or at least semi-ellipticity of the overall Bregman divergence B . Verifying this for $B = B_{J^0}$ with J^0 given in (14) is our next topic.

Ellipticity of the Bregman Divergences

As just discussed, for Theorem 1 to provide estimates that we can use to prove the convergence of the PDBS, we need at least the semi-ellipticity of B^0 generated by J^0 given in (14). Deriving simple conditions that ensure such semi-ellipticity or ellipticity is the topic of the present subsection. To do this, we need the “basic” Bregman divergences B_X and B_Y on both spaces X and Y to be elliptic:

➤ **Standing assumption**

In this subsection, we assume that B_X is τ^{-1} -elliptic and B_Y is σ^{-1} -elliptic for some $\tau, \sigma > 0$. This is true for the Hilbert-space PDPS (17) where τ and σ are the primal and dual step length parameters.

The examples that follow the next general lemma will provide improved estimates.

Lemma 2. *Suppose $K \in C^1(X \times Y)$ is Lipschitz-continuously differentiable with the factor L_{DK} in a convex subdomain $\Omega \subset X \times Y$. Then for $u, u' \in \Omega$*

$$B_K(u', u) \leq \frac{L_{DK}}{2} \|u' - u\|_{X \times Y}^2. \tag{21}$$

Consequently, if B_X is τ^{-1} -elliptic and B_Y is σ^{-1} -elliptic and $1 \geq \max\{\tau, \sigma\}L_{DK}$, then B^0 is semi-elliptic (elliptic if the inequality is strict) within Ω .

Proof. By definition, $B_K(u', u) = K(u') - K(u) - \langle DK(u)|u' - u \rangle$. Using the mean value equality in \mathbb{R} with the chain rule and the Cauchy–Schwarz inequality, we get

$$B_K(u', u) = \int_0^1 \langle DK(u+t(u'-u)) - DK(u) | u' - u \rangle dt \leq \int_0^1 t L_{DK} \|u' - u\|_{X \times Y}^2 dt.$$

Calculating the last integral yields (21).

For the (semi-)ellipticity, we need $B^0(u, u') \geq \frac{\epsilon}{2} \|u - u'\|_{X \times Y}^2$ for some $\epsilon > 0$ ($\epsilon = 0$) and all $u, u' \in \Omega$. Since B_X and B_Y are τ^{-1} - and σ^{-1} -elliptic, we have

$$\begin{aligned} B^0(u', u) &= B_X(x', x) + B_Y(y', y) - B_K(u', u) \\ &\geq \frac{1}{2\tau} \|x' - x\|_X^2 + \frac{1}{2\sigma} \|y' - y\|_Y^2 - B_K(u', u). \end{aligned} \tag{22}$$

Using (21), therefore $B^0(u', u) \geq \frac{\tau^{-1} - L_{DK}}{2} \|x' - x\|_X^2 + \frac{\sigma^{-1} - L_{DK}}{2} \|y' - y\|_Y^2$. Thus B^0 is ϵ -elliptic when $\tau^{-1}, \sigma^{-1} \geq L_{DK} + \epsilon$. This gives the claim. \square

We now provide several examples of ellipticity. In practice, to guarantee ellipticity, we would choose $\tau, \sigma > 0$ to satisfy the stated conditions.

Example 4. Suppose $K(x, y) = E(x)$ with DE L_{DE} -Lipschitz in $\Omega = X \times Y$. Then $L_{DK} = L_{DE}$, so we recover the standard-for-gradient-descent step length bound $1 \geq \tau L_{DE}$ for B^0 to be semi-elliptic in Ω (elliptic if the inequality is strict).

Example 5. If $K(x, y) = \langle Ax | y \rangle$ for $A \in \mathbb{L}(X; Y^*)$, then B^0 is elliptic under the standard-for-PDPS (Chambolle and Pock 2011) step length condition

$$1 > \tau \sigma \|A\|^2.$$

Indeed

$$\langle DK(u + t(u' - u)) - DK(u) | u' - u \rangle = 2t \langle A(x - x') | y - y' \rangle.$$

Therefore, taking any $w > 1$, we easily improve (21) to

$$\begin{aligned} B_K(u', u) &\leq \|A\| \|x' - x\|_X \|y' - y\|_Y \\ &\leq \frac{w\|A\|}{2} \|x' - x\|_X^2 + \frac{w^{-1}\|A\|}{2} \|y' - y\|_Y^2 \quad (u, u' \in X \times Y). \end{aligned} \tag{23}$$

By (22), B^0 is therefore ϵ -elliptic if $\tau^{-1} \geq w\|A\| + \epsilon$ and $\sigma^{-1} \geq w^{-1}\|A\| + \epsilon$. Taking $w = \sigma\|A\|/(1 - \sigma\epsilon)$, this holds if $1 \geq \tau\sigma\|A\|^2/(1 - \sigma\epsilon) + \tau\epsilon$. Since $\epsilon > 0$ was arbitrary, the claimed step length condition follows.

Example 6. Suppose $K(x, y) = \langle A(x)|y \rangle$ with A and DA Lipschitz with the respective factors $L_A, L_{DA} \geq 0$. Then B^0 is elliptic within $\Omega = X \times B(0, \rho_y)$ if

$$1 > \tau \sigma L_A^2 + \tau \frac{L_{DA} \rho_y}{2}.$$

Indeed, for any $w > 1$, using the mean value equality as in the proof of Lemma 2, we deduce

$$\begin{aligned} B_K(u', u) &= \langle A(x') - A(x)|y' \rangle - \langle DA(x)(x' - x)|y \rangle \\ &= \langle A(x') - A(x)|y' - y \rangle + \langle A(x') - A(x) - DA(x)(x' - x)|y \rangle \\ &\leq L_A \|x' - x\|_X \|y' - y\|_Y + \frac{L_{DA} \|y'\|}{2} \|x' - x\|_X^2 \\ &\leq \frac{w L_A + L_{DA} \|y\|}{2} \|x' - x\|_X^2 + \frac{w^{-1} L_A}{2} \|y' - y\|_Y^2. \end{aligned} \tag{24}$$

If $\rho_y > 0$ is such that $\|y\| \leq \rho_y$, taking $w = \sigma L_A / (1 - \sigma \epsilon)$, similarly to Example 5, we deduce the claimed bound.

We can combine the examples above:

Example 7. As in Example 2, take $K(x, (y_1, y_2)) = \langle A_1(x)|y_1 \rangle + \langle A_2(x)|y_2 \rangle$ with $A_1 \in C^1(X; Y_1^*)$ and $A_2 \in \mathbb{L}(X; Y_2^*)$. Then B^0 is elliptic within $\Omega = X \times B(0, \rho_y)$ if

$$1 > \tau \sigma (L_{A_1}^2 + \|A_2\|^2) + \tau \frac{L_{DA_1} \rho_{y_1}}{2}.$$

Indeed, we bound B_K by summing (23) for A_1 and (24) for A_2 . This yields for any $w_1, w_2 > 0$ the estimate

$$\begin{aligned} B_K(u', u) &\leq \frac{w_1 L_{A_1} + L_{DA_1} \|y_1\|}{2} \|x - x'\|_X^2 + \frac{w_1^{-1} L_{A_1}}{2} \|y'_1 - y_1\|_{Y_1}^2 \\ &\quad + \frac{w_2 \|A_2\|}{2} \|x' - x\|_X^2 + \frac{w_2^{-1} \|A_2\|}{2} \|y'_2 - y_2\|_{Y_2}^2. \end{aligned} \tag{25}$$

Taking $w_1 = \sigma L_{A_1} / (1 - \sigma \epsilon)$ and $w_2 = \sigma \|A_2\| / (1 - \sigma \epsilon)$ and using (22), we deduce the claimed ellipticity for small enough $\epsilon > 0$.

Remark 3. In Examples 6 and 7, we needed a bound on the dual variable y . In the latter, as an improvement, this was only needed on the subspace Y_1 of non-bilinearity. An ad hoc solution is to introduce the bound into the problem. In the

Hilbert case, Clason et al. (2019, 2020) secure such bounds by taking the primal step length τ small enough and arguing as in Theorem 1 individually on the primal and dual iterates.

Ellipticity for Block-Adapted Methods

We now study ellipticity for block-adapted methods. The goal is to obtain faster convergence by adapting the blockwise step length parameters to the problem structure (connections between blocks) and the local (blockwise) properties of the problem.

➤ **Standing assumption**

In this subsection, we assume $F, G_*, J_X,$ and J_Y to have the form of section “Block Adaptation.” In particular, X and Y are (products of) Hilbert spaces, and

$$B^0(u', u) = \sum_{j=1}^m \frac{1}{2\tau_j} \|x'_j - x_j\|_{X_j}^2 + \sum_{\ell=1}^n \frac{1}{2\sigma_\ell} \|y'_\ell - y_\ell\|_{Y_\ell}^2 - B_K(u', u). \quad (26)$$

We start by refining the two-block Example 7 to be adapted to the blocks:

Example 8. Let $K(x, (y_1, y_2)) = \langle A_1(x) | y_1 \rangle + \langle A_2 x | y_2 \rangle$ with $A_1 \in C^1(X; Y_1^*)$ and $A_2 \in \mathbb{L}(X; Y_2^*)$ as in Examples 2 and 7. Write $\tau = \tau_1$. Using (25) in (26) for $m = 1$ and $n = 2$ with (25), we see B^0 to be ϵ -elliptic within $\Omega = X \times B(0, \rho_{y_1}) \times Y_2$ if $\tau^{-1} \geq w_1 L_{A_1} + L_{DA_1} \rho_{y_1} + w_2 \|A_2\| + \epsilon$ and $\sigma_1^{-1} \geq w_1^{-1} L_{A_1}$ as well as $\sigma_2^{-1} \geq w_2^{-1} \|A_2\| + \epsilon$. Taking $w_1 = \sigma_1 L_{A_1} / (1 - \sigma_1 \epsilon)$ and $w_2 = \sigma_2 \|A_2\| / (1 - \sigma_2 \epsilon)$, B^0 is therefore elliptic (some $\epsilon > 0$) within Ω if $1 > \tau(\sigma_1 L_{A_1}^2 + \sigma_2 \|A_2\|^2) + \tau \frac{L_{DA_1} \rho_{y_1}}{2}$.

Example 9. In Example 8, if both $A_1 \in \mathbb{L}(X; Y_1^*)$ and $A_2 \in \mathbb{L}(X; Y_2^*)$, then B^0 is elliptic within $\Omega = X \times Y_1 \times Y_2$ if $1 > \tau(\sigma_1 \|A_1\|^2 + \sigma_2 \|A_2\|^2)$.

Example 10. Suppose we can write $K(x, y) = \sum_{j=1}^m \sum_{\ell=1}^n K_{j\ell}(x_j, y_\ell)$ with each $K_{j\ell}$ Lipschitz-continuously differentiable with the factor $L_{j\ell}$. Following Lemma 2

$$B_K(u', u) \leq \sum_{j=1}^m \sum_{\ell=1}^n \frac{L_{j\ell}}{2} (\|x'_j - x_j\|^2 + \|y'_\ell + y_\ell\|^2). \quad (27)$$

Consequently, using (26), we see that B^0 is ϵ -elliptic if $1 \geq \tau_j (\sum_{\ell=1}^n L_{j\ell} + \epsilon)$ and $1 \geq \sigma_\ell (\sum_{j=1}^m L_{j\ell} + \epsilon)$ for all $j = 1, \dots, m$ and $\ell = 1, \dots, n$.

Example 11. If $K(x, y) = \sum_{j=1}^m \sum_{\ell=1}^m \langle A_{j\ell} x_j | y_\ell \rangle$ for some $A_{j\ell} \in \mathbb{L}(X_j; Y_\ell^*)$, then following Example 5, for arbitrary $w_{j\ell} > 0$

$$\begin{aligned} B_K(u', u) &\leq \sum_{j=1}^m \sum_{\ell=1}^m \|A_{j\ell}\| \|x'_j - x_j\| \|y'_j - y_j\| \\ &\leq \sum_{j=1}^m \sum_{\ell=1}^n \left(\frac{w_{j\ell} \|A_{j\ell}\|}{2} \|x'_j - x_j\|^2 + \frac{w_{j\ell}^{-1} \|A_{j\ell}\|}{2} \|y'_j - y_j\|^2 \right). \end{aligned}$$

Using (26), B^0 is thus ϵ -elliptic if $1 \geq \tau_j(\epsilon + \sum_{\ell=1}^n w_{j\ell} \|A_{j\ell}\|)$ and $1 \geq \sigma_\ell(\epsilon + \sum_{j=1}^m w_{j\ell}^{-1} \|A_{j\ell}\|)$ for all $j = 1, \dots, m$ and $\ell = 1, \dots, n$. We can use the factors $w_{j\ell}$ to adapt the algorithm to the different blocks for potentially better convergence.

Nonsmooth Second-Order Conditions

We now study conditions for (C) to hold with $\mathcal{G}(\cdot, \bar{u}) \geq 0$. We start by writing out the condition for the PDBS.

Lemma 3. *Let $\bar{u} = (\bar{x}, \bar{y}) \in X \times Y$, and suppose for some $\mathcal{G}(u, \bar{u}) \in \mathbb{R}$ and a neighborhood $\Omega_{\bar{u}} \subset X \times Y$ that for all $u = (x, y) \in \Omega_{\bar{u}}$, $x^* \in \partial F(x)$ and $y^* \in \partial G_*(y)$*

$$\langle x^* + D_x K(x, y) | x - \bar{x} \rangle + \langle y^* - D_y K(x, y) | y - \bar{y} \rangle \geq \mathcal{G}(u, \bar{u}). \tag{C^2}$$

Let $\{u^{k+1}\}_{k \in \mathbb{N}}$ be generated by the PDBS (16) for some $u^0 \in X \times Y$, and suppose $\{u^k\}_{k \in \mathbb{N}} \subset \Omega_{\bar{u}}$. Then with $B = B^0$ the fundamental condition (C) and the quantitative Δ -Féjer monotonicity (F) hold for all $k \in \mathbb{N}$, and the descent inequality (D) holds for all $N \geq 1$.

Proof. Theorem 1 proves (F) and (D) if we show (C²). For H in (12), we have

$$h^{k+1} = \begin{pmatrix} x_{k+1}^* + D_x K(x^{k+1}, y^{k+1}) \\ y_{k+1}^* - D_y K(x^{k+1}, y^{k+1}) \end{pmatrix} \in H(u^{k+1}) \quad \text{with} \quad \begin{cases} x_{k+1}^* \in \partial F(x^{k+1}), \\ y_{k+1}^* \in \partial G_*(y^{k+1}). \end{cases}$$

Thus (C) expands as (C²) for $u = u^{k+1}$ and $(x^*, y^*) = (x_{k+1}^*, y_{k+1}^*)$. □

In section “Convergence of Gaps in the Convex-Concave Setting” on the convergence of gap functionals, we will consider general \bar{u} in (C²). For the moment, we however fix a root $\bar{u} = \hat{u} \in H^{-1}(0)$. Then

$$0 = \begin{pmatrix} \hat{x}^* + D_x K(\hat{x}, \hat{y}) \\ \hat{y}^* - D_y K(\hat{x}, \hat{y}) \end{pmatrix} \in H(\hat{u}) \quad \text{with} \quad \begin{cases} \hat{x}^* \in \partial F(\hat{x}), \\ \hat{y}^* \in \partial G_*(\hat{y}). \end{cases} \quad (28)$$

Since we assume F and G_* to be convex; their subdifferentials are monotone. When K is not convex-concave and to obtain strong convergence of iterates even when it is, we will need some strong monotonicity of the subdifferentials, but only *at* a solution. Specifically, for $\gamma > 0$, we say that $T : X \rightrightarrows X^*$ is γ -strongly monotone at \hat{x} for $\hat{x}^* \in T(\hat{x})$ if

$$\langle x^* - \hat{x}^* | x - \hat{x} \rangle \geq \gamma \|x - \hat{x}\|_X^2 \quad (x \in X, x^* \in T(x)). \quad (29)$$

If $\gamma = 0$, we drop the word “strong.” For $T = \partial F$, (29) follows from the γ -strong subdifferentiability of F .

> Standing assumption

Throughout the rest of this subsection, we assume (28) to hold and that ∂F is (γ_F -strongly) monotone at \hat{x} for \hat{x}^* and ∂G_* is (γ_{G_*} -strongly) monotone at \hat{y} for \hat{y}^* .

Lemma 4. *The nonsmooth second-order growth condition (C²) holds provided*

$$\gamma_F \|x - \hat{x}\|^2 + \gamma_{G_*} \|y - \hat{y}\|^2 \geq B_K(\hat{u}, u) + B_K(u, \hat{u}) + \mathcal{G}(u, \hat{u}) \quad (u \in \Omega_{\bar{u}}), \quad (30)$$

equivalently

$$\gamma_F \|x - \hat{x}\|^2 + \gamma_{G_*} \|y - \hat{y}\|^2 \geq a_K(\hat{u}, u) + a_K(u, \hat{u}) + \mathcal{G}(u, \hat{u}) \quad (u \in \Omega_{\bar{u}}) \quad (30')$$

for

$$a_K(u, \bar{u}) := K(x, y) - K(\bar{x}, \bar{y}) + \langle D_x K(x, y) | \bar{x} - x \rangle + \langle D_y K(\bar{x}, \bar{y}) | \bar{y} - y \rangle. \quad (31)$$

Note that (30) involves the symmetrized Bregman divergence $B_K^S(u, u') := B_K(u, u') + B_K(u', u)$ generated by K .

Proof. Inserting the zero of (28) in (C²), we rewrite the latter as

$$\begin{aligned} \langle x^* - \hat{x}^* | x - \hat{x} \rangle + \langle y^* - \hat{y}^* | y - \hat{y} \rangle &\geq \langle D_x K(x, y) - D_x K(\hat{x}, \hat{y}) | \hat{x} - x \rangle \\ &\quad + \langle D_y K(x, y) - D_y K(\hat{x}, \hat{y}) | y - \hat{y} \rangle + \mathcal{G}(u^{k+1}, \hat{u}). \end{aligned}$$

Using the assumed strong monotonicities, and the definitions of B_K and a_K , this is immediately seen to hold when (30) or (30') does. □

Example 12. If K is convex-concave, the next Lemma 5 and Lemma 4 prove (C²) for

$$\mathcal{G}(u, \hat{u}) = \gamma_F \|x - \hat{x}\|^2 + \gamma_{G_*} \|y - \hat{y}\|^2 \geq 0 \quad \text{within} \quad \Omega_{\hat{u}} = X \times Y.$$

This is in particular true for $K(x, y) = \langle Ax|y \rangle + E(x)$ with $A \in \mathbb{L}(X; Y^*)$ and $E \in C^1(X)$ convex.

Lemma 5. *Suppose $K : X \times Y \rightarrow \mathbb{R}$ is Gâteaux-differentiable and convex-concave. Then $a_K(u, \bar{u}) \leq 0$ and $B_K^S(u, \bar{u}) \leq 0$ for all $u, \bar{u} \in X \times Y$.*

Proof. The convexity of $K(\cdot, y)$ and the concavity of $K(\bar{x}, \cdot)$ show

$$\begin{aligned} K(x, y) - K(\bar{x}, y) + \langle D_x K(x, y)|\bar{x} - x \rangle &\leq 0 \quad \text{and} \\ K(\bar{x}, y) - K(\bar{x}, \bar{y}) + \langle D_y K(\bar{x}, \bar{y})|\bar{y} - y \rangle &\leq 0. \end{aligned}$$

Summing these two estimates proves $a_K(u, \bar{u}) \leq 0$, consequently $B_K^S(u, \bar{u}) = a_K(u, \bar{u}) + a_K(\bar{u}, u) \leq 0$. \square

Example 13. Suppose K has L_{DK} -Lipschitz derivative within $\Omega \subset X \times Y$. If $\hat{u} \in \Omega$, then by Lemma 2, $B_K(u, \hat{u}), B_K(\hat{u}, u) \leq \frac{L_{DK}}{2} \|u - \hat{u}\|_{X \times Y}^2$ for $u \in \Omega$. Thus (C²) holds by Lemma 4 with $\Omega_{\hat{u}} = \Omega$ and

$$\mathcal{G}(u, \hat{u}) = (\gamma_F - L_{DK}) \|x - \hat{x}\|^2 + (\gamma_{G_*} - L_{DK}) \|y - \hat{y}\|^2.$$

This is non-negative if $\gamma_F, \gamma_{G_*} \geq L_{DK}$.

Example 14. Let $K(x, y) = \langle A(x)|y \rangle$ for some $A \in \mathbb{L}(X; Y^*)$ such that DA is Lipschitz with the factor $L_{DA} \geq 0$. For some $\tilde{\gamma}_F, \tilde{\gamma}_{G_*} \geq 0$ and $\rho_y, \hat{\rho}_x, \alpha > 0$, let either

- (a) $\tilde{\gamma}_F \geq \frac{L_{DA}}{2} (\rho_y + \|\hat{y}\|_Y)$, $\tilde{\gamma}_{G_*} \geq 0$, and $\Omega_{\hat{u}} = X \times B(0, \rho_y)$; or
- (b) $\tilde{\gamma}_F > L_{DA} \left(\|\hat{y}\|_Y + \frac{\alpha}{2} \right)$, $\tilde{\gamma}_{G_*} \geq \frac{L_{DA}}{2\alpha} \hat{\rho}_x^2$, and $\Omega_{\hat{u}} = B(\hat{x}, \hat{\rho}_x) \times Y$.

Then Lemma 4 proves (C²) with

$$\mathcal{G}(u, \hat{u}) = (\gamma_F - \tilde{\gamma}_F) \|x - \hat{x}\|^2 + (\gamma_{G_*} - \tilde{\gamma}_{G_*}) \|y - \hat{y}\|^2.$$

To see this, we need to prove (30'). Now

$$a_K(u, \hat{u}) := \langle A(x) - A(\hat{x}) + DA(x)(\hat{x} - x)|y \rangle \quad (u, \hat{u} \in X \times Y). \quad (32)$$

Arguing with the mean value equality and the Lipschitz assumption as in Lemma 2, we get $a_K(\hat{u}, u) + a_K(u, \hat{u}) \leq \frac{L_{DA}}{2}(\|y\|_Y + \|\hat{y}\|_Y)\|x - \hat{x}\|^2$. Thus (a) implies (30'). By (32), the mean-value equality, and the Lipschitz assumption, also

$$\begin{aligned} a_K(u, \hat{u}) + a_K(\hat{u}, u) &= \langle [DA(x) - DA(\hat{x})](\hat{x} - x) | \hat{y} \rangle \\ &\quad + \langle A(x) - A(\hat{x}) + DA(x)(\hat{x} - x) | y - \hat{y} \rangle \\ &\leq L_{DA}\|x - \hat{x}\|_X^2 (\|\hat{y}\|_Y + \frac{1}{2}\|y - \hat{y}\|_Y). \end{aligned}$$

Using Cauchy's inequality and (b) we deduce (30').

Remark 4. In the last two examples, we need to bound some of the iterates and to initialize close enough to a solution. Showing that the iterates stay in a local neighborhood is a large part of the work in Clason et al. (2019, 2020), as discussed in Remark 3.

Second-Order Growth Conditions for Block-Adapted Methods

We now study second-order growth for problems with a block structure as in section “Block Adaptation”:

➤ Standing assumption

In this subsection, F and G_* are as in section “Block Adaptation,” each component subdifferential ∂F_j now (γ_{F_j} -strongly) monotone at \hat{x}_j for \hat{x}_j^* and each ∂G_{ℓ^*} ($\gamma_{G_{\ell^*}}$ -strongly) monotone at \hat{y}_ℓ for \hat{y}_ℓ^* . Here \hat{x}_j , \hat{x}_j^* , \hat{y}_ℓ , and \hat{y}_ℓ^* are the components of \hat{x} , \hat{x}^* , \hat{y} , and \hat{y}^* in the corresponding subspace, assumed to satisfy the critical point condition (28).

As only some of the component functions may have $\gamma_{F_j}, \gamma_{G_{\ell^*}} > 0$, through detailed analysis of the block structure, we hope to obtain (strong) convergence on some subspaces even if the entire primal or dual variables might not converge.

Similarly to Lemma 4 we prove:

Lemma 6. *Suppose for some neighborhood $\Omega_{\hat{u}} \subset X \times Y$ that*

$$\Delta_{k+1} := \sum_{j=1}^m \tilde{\gamma}_{F_j} \|x_j - \hat{x}_j\|_{X_j}^2 + \sum_{\ell=1}^n \tilde{\gamma}_{G_{\ell^*}} \|y_\ell - \hat{y}_\ell\|_{Y_\ell}^2 \geq a_K(\hat{u}, u) + a_K(u, \hat{u})$$

for some $\tilde{\gamma}_{F_j}, \gamma_{G_{\ell^*}} \geq 0$ for all $u \in \Omega_{\hat{u}}$. Then (C²) holds with

$$\mathcal{G}(u, \hat{u}) = \sum_{j=1}^m (\gamma_{F_j} - \tilde{\gamma}_{F_j}) \|x_j - \hat{x}_j\|_{X_j}^2 + \sum_{\ell=1}^n (\gamma_{G_{\ell^*}} - \tilde{\gamma}_{G_{\ell^*}}) \|y_\ell - \hat{y}_\ell\|_{Y_\ell}^2. \tag{33}$$

In the convex-concave case, we can transfer all strong monotonicity into \mathcal{G} :

Example 15. If K is convex-concave, then by Lemmas 5 and 6, (C^2) holds with $\Omega_{\hat{u}} = X \times Y$ and \mathcal{G} as in (33) for $\tilde{\gamma}_{F_j} = 0$ and $\tilde{\gamma}_{G_{\ell^*}} = 0$. We have $\mathcal{G}(\cdot, \hat{u}) \geq 0$ always.

Example 16. As in Example 10, suppose we can write $K(x, y) = \sum_{j=1}^m \sum_{\ell=1}^n K_{j\ell}(x_j, y_\ell)$ with each $K_{j\ell}$ Lipschitz-continuously differentiable with the factor $L_{j\ell}$ in Ω . Then using (27) and Lemma 6, we see (C^2) to hold with $\Omega_{\hat{u}} = \Omega$ and \mathcal{G} as in (33) with

$$\tilde{\gamma}_{F_j} = \sum_{\ell=1}^n L_{j\ell} \quad (j = 1, \dots, m) \quad \text{and} \quad \tilde{\gamma}_{G_{\ell^*}} = \sum_{j=1}^m L_{j\ell} \quad (\ell = 1, \dots, n).$$

Thus $\mathcal{G}(\cdot, \hat{u}) \geq 0$ if $\gamma_{F_j} \geq \sum_{\ell=1}^n L_{j\ell}$ and $\gamma_{G_{\ell^*}} \geq \sum_{j=1}^m L_{j\ell}$ for all ℓ and j .

The special case of Example 10 with each $K_{j\ell}$ bilinear, corresponding to Example 11 for ellipticity, is covered by Example 15.

We consider in detail the two dual block setup of Examples 2 and 8:

Example 17. As in Example 2, let $K(x, y) = \langle A_1(x)|y_1 \rangle + \langle A_2x|y_2 \rangle$ for $A_1 \in C^1(X; Y_1^*)$ and $A_2 \in \mathbb{L}(X; Y_2^*)$. Then, as in (32),

$$a_K(u, \bar{u}) = \langle A_1(x) - A_1(\bar{x}) + DA_1(x)(\bar{x} - x)|y_1 \rangle,$$

which does not depend on A_2 . For any $\alpha, \rho_y, \hat{\rho}_x > 0$, let either

- (a) $\tilde{\gamma}_F \geq \frac{L_{DA_1}}{2}(\rho_{y_1} + \|\hat{y}_1\|_{Y_1})$, $\tilde{\gamma}_{G_{1^*}} \geq 0$, and $\Omega_{\hat{u}} = X \times B(0, \rho_{y_1})$; or
- (b) $\tilde{\gamma}_F > L_{DA_1} \left(\|\hat{y}_1\|_{Y_1} + \frac{\alpha}{2} \right)$, $\tilde{\gamma}_{G_{1^*}} \geq \frac{L_{DA_1}}{2\alpha} \hat{\rho}_x^2$, and $\Omega_{\hat{u}} = B(\hat{x}, \hat{\rho}_x) \times Y$.

Arguing as in Example 14 and using Lemma 6, we then see (C^2) to hold with \mathcal{G} as in (33) and $\tilde{\gamma}_{G_{2^*}} = 0$. In this case $\mathcal{G}(\cdot, \hat{u})$ is non-negative if $\gamma_F \geq \tilde{\gamma}_F$ and $\gamma_{G_{1^*}} \geq \tilde{\gamma}_{G_{1^*}}$.

Convergence of Iterates

We are now ready to prove the convergence of the iterates. We start with weak convergence and proceed to strong and linear convergence. For weak convergence in infinite dimensions, we need some further technical assumptions. We recall that a set-valued map $T : X \rightrightarrows X^*$ is weak-to-strong (weak-*to-strong) outer semicontinuous if $x_k^* \in T(x^k)$ and $x^k \rightharpoonup x$ ($x^k \overset{*}{\rightharpoonup} x$) and $x_k^* \rightarrow x^*$ imply $x^* \in T(x)$. The nonreflexive case of the next assumption covers spaces of functions of bounded variation (Ambrosio et al. 2000, Remark 3.12), important for total variation based imaging.

Assumption 1. Each of the spaces X and Y is, individually, either a reflexive Banach space or the dual of separable space. The operator $H : X \times Y \rightrightarrows X^* \times Y^*$ is weak(-*)-to-strong outer semicontinuous, where we mean by “weak(-*)” that we take the weak topology if the space is reflexive and weak-* otherwise, individually on X and Y .

Subdifferentials of lower semicontinuous convex functions are weak(-*)-to-strong outer semicontinuous⁵, so the outer semicontinuity of H depends mainly on K .

Example 18. If X and Y are finite-dimensional, Assumption 1 holds if $K \in C^1(X; Y)$.

Example 19. More generally, Assumption 1 holds if $K \in C^1(X \times Y)$ and DK is continuous from the weak(-*) topology to the strong topology.

Example 20. If $K = \langle Ax|y \rangle + E(x)$ for $A \in \mathbb{L}(X; Y^*)$ and $E \in C^1(X)$ convex, then H satisfies Assumption 1. Indeed, it can be shown that H is maximal monotone, hence weak(-*) outer semicontinuous similarly to convex subdifferentials.

> Verification of the conditions

To verify the nonsmooth second-order growth condition (C^2) for each of the following Theorems 2, 3, and 4, we point to sections “Nonsmooth Second-Order Conditions” and “Second-Order Growth Conditions for Block-Adapted Methods.” For the verification of the (semi-)ellipticity of B^0 , we point to sections “Ellipticity of the Bregman Divergences” and “Ellipticity for Block-Adapted Methods.” As special cases of the PDBS (16), the theorems apply to the Hilbert-space PDPS (17) and its block adaptation (18). Then J_X and J_Y are continuously differentiable and convex.

⁵This result seems difficult to find in the literature for Banach spaces but follows easily from the definition of the subdifferential: If $F(x) \geq F(x^k) + \langle x_k^*|x - x^k \rangle$ and $x_k^* \rightarrow \hat{x}^*$ as well as $x^k \rightharpoonup$ (or $\overset{*}{\rightharpoonup}) \hat{x}$, then, using the fact that $\{\|x^k - \hat{x}\|\}_{k \in \mathbb{N}}$ is bounded, in the limit $F(x) \geq F(\hat{x}) + \langle \hat{x}^*|x - \hat{x} \rangle$.

Theorem 2 (Weak convergence). *Let F and G_* be convex, proper, and lower semicontinuous; $K \in C^1(X \times Y)$; and both $J_X \in C^1(X)$ and $J_Y \in C^1(Y)$ convex. Suppose Assumption 1 holds and for some $\hat{u} \in H^{-1}(0)$ that*

- (i) (C^2) holds with $\mathcal{G}(\cdot, \hat{u}) \geq 0$ within $\Omega_{\hat{u}} \subset X \times Y$.
- (ii) B^0 is elliptic within $\Omega \ni \hat{u}$.

Let $\{u^{k+1}\}_{k \in \mathbb{N}}$ be generated by the PDBS (16) for any initial u^0 , and suppose $\{u^k\}_{k \in \mathbb{N}} \subset \Omega \cap \Omega_{\hat{u}}$. Then there exists at least one cluster point of $\{u^k\}_{k \in \mathbb{N}}$, and all weak(-*) cluster points belong to $H^{-1}(0)$.

Proof. Lemma 3 establishes (D) for $B = B^0$ and all $N \geq 1$. With $\epsilon > 0$ the factor of ellipticity of B^0 , it follows

$$\frac{\epsilon}{2} \|u^N - \hat{u}\|_{X \times Y}^2 + \frac{\epsilon}{2} \sum_{k=0}^{N-1} \|u^{k+1} - u^k\|_{X \times Y}^2 \leq B^0(\hat{u}, u^0) \quad (N \geq 1).$$

Clearly $\|u^{k+1} - u^k\| \rightarrow 0$ while $\{\|u^k - \hat{u}\|\}_{k \in \mathbb{N}}$ is bounded. Using the Eberlein–Šmulyan theorem in a reflexive X or Y and the Banach–Alaoglu theorem otherwise (X or Y the dual of a separable space), we may therefore find a subsequence of $\{u^k\}_{k \in \mathbb{N}}$ converging weakly(-*) to some \bar{x} . Since $J^0 \in C^1(X \times Y)$, we deduce $D_1 B^0(u^{k+1}, u^k) \rightarrow 0$. Consequently (15) implies that $0 \in \limsup_{k \rightarrow \infty} H(u^{k+1})$, where we write “lim sup” for the Painlevé–Kuratowski outer limit of a sequence of sets in the strong topology. Since H is weak(-*)-to-strong outer semicontinuous by Assumption 1, it follows that $0 \in H(\hat{u})$. Therefore, there exists at least one cluster point of $\{u^k\}_{k \in \mathbb{N}}$ belonging to $H^{-1}(0)$. Repeating the argument on any weak(-*) convergent subsequence, we deduce that all cluster points belong to $H^{-1}(0)$. \square

Remark 5. For a unique weak limit, we may in Hilbert spaces use the quantitative Féjér monotonicity (F) with Opial’s lemma (Opial 1967; Browder 1967). For bilinear K the result is relatively immediate, as B^0 is a squared matrix-weighted norm; see Valkonen (2020). Otherwise a variable-metric Opial’s lemma (Clason et al. 2019) and additional work based on the Brezis–Crandall–Pazy lemma (Brezis et al. 1970, Corollary 20.59 (iii)) are required; see Clason et al. (2019) for $K(x, y) = \langle A(x)|y \rangle$ and Clason et al. (2020) for general K .

Theorem 3 (Strong convergence). *Let F and G_* be convex, proper, and lower semicontinuous; $K \in C^1(X \times Y)$; and both $J_X \in C(X)$ and $J_Y \in C(Y)$ convex and Gâteaux-differentiable. Suppose for some $\hat{u} \in H^{-1}(0)$ that*

- (i) (C^2) holds with $\mathcal{G}(\cdot, \hat{u}) \geq 0$ within $\Omega_{\hat{u}} \subset X \times Y$.
- (ii) B^0 is semi-elliptic within $\Omega \ni \hat{u}$.

Let $\{u^{k+1}\}_{k \in \mathbb{N}}$ be generated by the PDBS (16) for any initial u^0 , and suppose $\{u^k\}_{k \in \mathbb{N}} \subset \Omega \cap \Omega_{\hat{u}}$. Then $\mathcal{G}(u^{k+1}, \hat{u}) \rightarrow 0$ as $N \rightarrow \infty$.

In particular, if $\mathcal{G}(u, \hat{u}) \geq \|P(u - \hat{u})\|_Z^2$ for some $P \in \mathbb{L}(X; Z)$, then $Px^N \rightarrow P\hat{x}$ strongly in Z and the ergodic sequence $\tilde{x}_P^N := \frac{1}{N} \sum_{k=0}^{N-1} Px^{k+1} \rightarrow P\hat{x}$ at rate $O(1/N)$.

Proof. Lemma 3 establishes (D). By the semi-ellipticity of B^0 , then $\sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \hat{u}) \leq B^0(\hat{u}, u^0)$, ($N \in \mathbb{N}$). Since $\mathcal{G}(u^{k+1}, \hat{u}) \geq 0$, this shows that $\mathcal{G}(u^N, \hat{u}) \rightarrow 0$. The strong convergence of the primal variable for quadratically minorized \mathcal{G} is then immediate whereas following by Jensen’s inequality gives the ergodic convergence claim. \square

Example 21. In section “[Nonsmooth Second-Order Conditions](#),” we can take $Pu = \sqrt{\gamma_F - \tilde{\gamma}_F}x$ if $\gamma_F > \tilde{\gamma}_F$ or $Pu = \sqrt{\gamma_{G_*} - \tilde{\gamma}_{G_*}}y$ if $\gamma_{G_*} > \tilde{\gamma}_{G_*}$. The examples of section “[Second-Order Growth Conditions for Block-Adapted Methods](#)” for $x = (x_1, \dots, x_m)$ and $y = (y_1, \dots, y_n)$ may allow $Pu = \sqrt{\gamma_{F_j} - \tilde{\gamma}_{F_j}}x_j$ or $Pu = \sqrt{\gamma_{G_{\ell_*}} - \tilde{\gamma}_{G_{\ell_*}}}y_{\ell}$.

Remark 6. Under similar conditions as Theorem 3, it is possible to obtain $O(1/N^2)$ convergence rates; see Chambolle and Pock (2011) and Valkonen (2020) for the convex-concave case and Clason et al. (2019, 2020) in general.

Theorem 4 (Linear convergence). *Let F and G_* be convex, proper, and lower semicontinuous; $K \in C^1(X \times Y)$; and both $J_X \in C(X)$ and $J_Y \in C(Y)$ convex and Gâteaux-differentiable. Suppose for some $\gamma > 0$ and $\hat{u} \in H^{-1}(0)$ that*

- (i) (C^2) holds with $\mathcal{G}(u, \hat{u}) \geq \gamma B^0(\hat{u}, u)$ within $\Omega_{\hat{u}} \subset X \times Y$.
- (ii) B^0 is elliptic within $\Omega \supset \hat{u}$.

Let $\{u^{k+1}\}_{k \in \mathbb{N}}$ be generated by the PDBS (16) for any initial u^0 , and suppose $\{u^k\}_{k \in \mathbb{N}} \subset \Omega \cap \Omega_{\hat{u}}$. Then $B^0(\hat{u}, u^N) \rightarrow 0$ and $u^N \rightarrow \hat{u}$ at a linear rate.

In particular, if $\mathcal{G}(u, \hat{u}) \geq \gamma \|u - \hat{u}\|^2$, ($k \in \mathbb{N}$), for some $\gamma > 0$, and J^0 is Lipschitz-continuously differentiable, then $u^N \rightarrow \hat{u}$ at a linear rate.

Proof. Lemma 3 establishes the quantitative Δ -Féjer monotonicity (F). Using (i), this yields $(1 + \gamma)B^0(\hat{u}, u^{k+1}) \leq B^0(\hat{u}, u^k)$. By the semi-ellipticity of B^0 , the claimed linear convergence of $B^0(\hat{u}, u^N) \rightarrow 0$ follows. Since B^0 is assumed elliptic, also $u^N \rightarrow \hat{u}$ linearly. If J^0 is Lipschitz-continuously differentiable, then, similarly to Lemma 2, $B^0(\hat{u}, u^{k+1}) \leq L_{DJ} \|u^{k+1} - \hat{u}\|^2$ for some $L_{DJ} > 0$. Thus $\mathcal{G}(u^{k+1}, \hat{u}) \geq \gamma H_{DJ}^{-1} B^0(\hat{u}, u^{k+1})$, so the main claim establishes the particular claim. \square

Example 22. J^0 is Lipschitz-continuously differentiable if X and Y are Hilbert spaces with $J_X = \tau^{-1}N_X$ and $J_Y = \sigma^{-1}N_Y$, and K is Lipschitz-continuously differentiable.

Convergence of Gaps in the Convex-Concave Setting

We finish this section by studying the convergence of gap functionals in the convex-concave setting.

Lemma 7. *Suppose F and G_* are convex, proper, and lower semicontinuous and $K \in C^1(X \times Y)$ is convex-concave on $\text{dom } F \times \text{dom } G_*$. Then (C²) holds for all $\bar{u} \in X \times Y$ with $\Omega_{\bar{u}} = X \times Y$ and $\mathcal{G} = \mathcal{G}^{\mathcal{L}}$ the Lagrangian gap*

$$\begin{aligned} \mathcal{G}^{\mathcal{L}}(u, \bar{u}) &:= \mathcal{L}(x, \bar{y}) - \mathcal{L}(\bar{x}, y) \\ &= [F(x) + K(x, \bar{y}) - G_*(\bar{y})] - [F(\bar{x}) + K(\bar{x}, y) - G_*(y)]. \end{aligned}$$

This functional is non-negative if $\bar{u} \in H^{-1}(0)$.

Moreover, if $\sum_{k=0}^{N-1} \mathcal{G}^{\mathcal{L}}(u^{k+1}, \bar{u}) \leq M(\bar{u})$ for some $M(\bar{u}) \geq 0$, for all $\bar{u} \in X \times Y$ and all $N \in \mathbb{N}$, and we define the ergodic sequence $\tilde{u}^N := \frac{1}{N} \sum_{k=0}^{N-1} u^{k+1}$, then

- (i) $0 \leq \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{G}^{\mathcal{L}}(u^{k+1}, \hat{u}) \rightarrow 0$ at the rate $O(1/N)$ for $\hat{u} \in H^{-1}(0)$.
- (ii) $0 \leq \mathcal{G}^{\mathcal{L}}(\tilde{u}^N, \hat{u}) \rightarrow 0$ at the rate $O(1/N)$ for $\hat{u} \in H^{-1}(0)$.
- (iii) If $M \in C(X \times Y)$ and $\Omega \subset X \times Y$ is bounded with $\Omega \cap H^{-1}(0) \neq \emptyset$, then $0 \leq \mathcal{G}_{\Omega}(\tilde{u}^N) \rightarrow 0$ at the rate $O(1/N)$ for the partial gap $\mathcal{G}_{\Omega}(u) := \sup_{\bar{u} \in \Omega} \mathcal{G}^{\mathcal{L}}(u, \bar{u})$.

The convergence results in Lemma 7 are ergodic because they apply to sequences of running averages. To understand the partial gap, we recall that with $K(x, y) = \langle Ax|y \rangle$ bilinear Fenchel–Rockafellar’s theorem shows that the duality gap $\mathcal{G}^D(u) := [F(x) + G_*(Ax)] + [F_*(-A^*y) + G^*(y)] \geq 0$ and is zero if and only if $u \in H^{-1}(0)$. The duality gap can be written $\mathcal{G}^D(u) = \mathcal{G}_{X \times Y}(u)$.

Proof. By the convex-concavity of K and the definition of the subdifferential

$$\begin{aligned} &\langle D_x K(x, y)|x - \bar{x} \rangle - \langle D_y K(x, y)|y - \bar{y} \rangle \\ &\geq [K(x, y) - K(\bar{x}, y)] - [K(x, y) - K(x, \bar{y})] = K(x, \bar{y}) - K(\bar{x}, y). \end{aligned}$$

for all $(x, y) \in X \times Y$. Also using $x^* \in \partial F(x^{k+1})$ and $y^* \in \partial G(y^{k+1})$ with the definition of the convex subdifferential, we see that $\mathcal{G} = \mathcal{G}^{\mathcal{L}}$ satisfies (C²). The non-negativity of $\mathcal{G}(\cdot, \hat{u})$ follows by similar reasoning, first using that

$$K(x, \hat{y}) - K(\hat{x}, y) \geq \langle D_x K(\hat{x}, \hat{y})|x - \hat{x} \rangle - \langle D_y K(\hat{x}, \hat{y})|y - \hat{y} \rangle \tag{34}$$

for all $(x, y) \in X \times Y$ and following by the definition of the subdifferential applied to $-D_x K(\hat{x}, \hat{y}) \in \partial F(\hat{x})$ and $D_y K(\hat{x}, \hat{y}) \in \partial G_*(\hat{y})$.

For (i)–(iii), we first observe that the semi-ellipticity of B^0 and (C²) imply $\sum_{k=0}^{N-1} \mathcal{G}^{\mathcal{L}}(u^{k+1}, \bar{u}) \leq M(\bar{u})$. Dividing by N and using that $\mathcal{G}^{\mathcal{L}}(u^{k+1}, \hat{u}) \geq 0$ for $\bar{u} \in H^{-1}(0)$, we obtain (i). Jensen’s inequality then gives $\mathcal{G}^{\mathcal{L}}(\tilde{u}^{k+1}, \bar{u}) \leq M(\bar{u})/N$,

hence (ii) for $\bar{u} \in H^{-1}(0)$. Finally, taking the supremum over $\bar{u} \in \Omega$ gives (iii) because M is bounded on bounded sets. \square

In the following theorem, we may in particular take $K(x, y) = \langle Ax|y \rangle$ bilinear or $K(x, y) = \langle Ax|y \rangle + E(x)$ with E convex. Lemma 2 and Examples 4 and 5 provide step length conditions that ensure the semi-ellipticity required of B^0 in Theorem 5.

Theorem 5 (Gap convergence). *Let $F : X \rightarrow \overline{\mathbb{R}}$ and $G_* : Y \rightarrow \overline{\mathbb{R}}$ be convex, proper, and lower semicontinuous. Also let $K \in C^1(X \times Y)$ be convex-concave within $\text{dom } F \times \text{dom } G_*$. Finally, let $J_X \in C^1(X)$ and $J_Y \in C^1(Y)$ convex. If B^0 is semi-elliptic, then the iterates $\{u^{k+1}\}_{k \in \mathbb{N}}$ generated by the PDBS (16) for any initial $u^0 \in X \times Y$ satisfies Lemma 7 (i)–(iii).*

Proof. By Lemma 7, holds with $\mathcal{G} = \mathcal{G}^{\mathcal{L}}$. Hence by Lemma 3, (D) holds. Since B^0 is semi-elliptic, this implies that that $\sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \bar{u}) \leq M(\bar{u}) := B^0(\bar{u}, u^0)$ for all $N \in \mathbb{N}$. Since J_X, J_Y , and K are continuously differentiable, $M \in C^1(X \times Y)$. The rest follows from the second part of Lemma 7. \square

Inertial Terms

We now generalize (BP), making the involved Bregman divergences dependent on the iteration k and earlier iterates

$$0 \in H(u^{k+1}) + D_1 B_{k+1}(u^{k+1}, u^k) + D_1 B_{k+1}^-(u^k, u^{k-1}), \tag{IPP}$$

for $B_{k+1} := B_{J_{k+1}}$ and $B_{k+1}^- := B_{J_{k+1}^-}$ generated by $J_{k+1}, J_{k+1}^- : U \rightarrow \overline{\mathbb{R}}$. We take $u^{-1} := u^0$ for this to be meaningful for $k = 0$. Our main reason for introducing the dependence on u^{k-1} is to improve (16) and (17) to be explicit in K when K is not affine in y : Otherwise the dual step of those methods is in general not practical to compute unlike the affine case of Remark 1. Along the way we also construct a more conventional inertial method.

A Generalization of the Fundamental Theorem

We realign indices to get a simple fundamental condition to verify on each iteration.

Theorem 6. *On a Banach space U , let $H : U \rightrightarrows U^*$, and let $J_k, J_k^- : U \rightarrow \overline{\mathbb{R}}$ be Gâteaux-differentiable with the corresponding Bregman divergences $B_k := B_{J_k}$ and $B_k^- := B_{J_k^-}$ for all $k = 1, \dots, N$. Suppose (IPP) is solvable for $\{u^{k+1}\}_{k \in \mathbb{N}}$ given an initial iterate $u^0 \in U$. If for all $k = 0, \dots, N - 1$, for some $\bar{u} \in U$ and $\mathcal{G}(u^{k+1}, \bar{u}) \in \mathbb{R}$, for all $h^{k+1} \in H(u^{k+1})$ the modified fundamental condition*

$$\langle h^{k+1} | u^{k+1} - \bar{u} \rangle \geq [(B_{k+2} + B_{k+3}^-) - (B_{k+1} + B_{k+2}^-)](\bar{u}, u^{k+1}) + \mathcal{G}(u^{k+1}, \bar{u}) \tag{IC}$$

holds, and B_{k+1}^- satisfies the general Cauchy inequality

$$\langle D_1 B_{k+1}^-(u^k, u) | u^k - u' \rangle \leq B'_{k+1}(u^k, u) + B''_{k+1}(u', u^k) \quad (u, u' \in X) \tag{35}$$

for some $B'_{k+1}, B''_{k+1} : U \times U \rightarrow \mathbb{R}$, then we have the modified descent inequality

$$\begin{aligned} [B_{N+1} + B_{N+2}^- - B''_{N+1}](\bar{u}, u^N) + \sum_{k=0}^{N-1} [B_{k+1} + B_{k+2}^- - B''_{k+1} - B'_{k+2}](u^{k+1}, u^k) \\ + \sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \bar{u}) \leq [B_1 + B_2^-](\bar{u}, u^0). \end{aligned} \tag{ID}$$

Proof. We can write (IPP) as

$$0 = h^{k+1} + D_1 B_{k+1}(u^{k+1}, u^k) + D_1 B_{k+1}^-(u^k, u^{k-1}) \text{ for some } h^{k+1} \in H(u^{k+1}). \tag{36}$$

Testing (IPP) by applying $\langle \cdot | u^{k+1} - \bar{u} \rangle$, we obtain

$$0 = \langle h^{k+1} + D_1 B_{k+1}(u^{k+1}, u^k) + D_1 B_{k+1}^-(u^k, u^{k-1}) | u^{k+1} - \bar{u} \rangle.$$

Summing over $k = 0, \dots, N-1$ and using $u^{-1} = u^0$ to eliminate $B_1^-(u^0, u^{-1}) = 0$, we rearrange

$$0 = S_N + \sum_{k=0}^{N-1} \langle h^{k+1} + D_1 [B_{k+1} + B_{k+2}^-](u^{k+1}, u^k) | u^{k+1} - \bar{u} \rangle \tag{37}$$

for

$$S_N := \langle D_1 B_{J_{N+1}^-}(u^N, u^{N-1}) | \bar{u} - u^N \rangle + \sum_{k=0}^{N-1} \langle D_1 B_{J_{k+1}^-}(u^k, u^{k-1}) | u^{k+1} - u^k \rangle.$$

Abbreviating $\bar{B}_{k+1} := B_{k+1} + B_{k+2}^-$ and using (IC) and the three-point identity (8) in (37), we obtain

$$0 \geq S_N + \sum_{k=0}^{N-1} \left(\bar{B}_{k+2}(\bar{u}, u^{k+1}) - \bar{B}_{k+1}(\bar{u}, u^k) + \bar{B}_{k+1}(u^{k+1}, u^k) + \mathcal{G}(u^{k+1}, \bar{u}) \right).$$

Using the generalized Cauchy inequality (35) and, again, that $u^{-1} = u^0$, we get

$$\begin{aligned}
S_N &\geq -B'_{N+1}(u^N, u^{N-1}) - B''_{N+1}(\bar{u}, u^N) - \sum_{k=0}^{N-1} (B'_{k+1}(u^k, u^{k-1}) + B''_{k+1}(u^{k+1}, u^k)) \\
&= -B''_{N+1}(\bar{u}, u^N) - \sum_{k=0}^{N-1} [B''_{k+1} + B'_{k+2}](u^{k+1}, u^k).
\end{aligned}$$

These two inequalities yield (ID). \square

Inertia (Almost) as Usually Understood

We take $J_{k+1} = J^0$ and $J_{k+1}^- = -\lambda_k J^0$ for some $\lambda_k \in \mathbb{R}$. We then expand (IPP) as

Inertial PDBS

Iteratively over $k \in \mathbb{N}$, solve for x^{k+1} and y^{k+1} :

$$\begin{aligned}
(1 + \lambda_k)[DJ_X(x^k) - D_x K(x^k, y^k)] - \lambda_k [DJ_X(x^{k-1}) - D_x K(x^{k-1}, y^{k-1})] \\
&\in DJ_X(x^{k+1}) + \partial F(x^{k+1}), \\
(1 + \lambda_k)[DJ_Y(y^k) - D_y K(x^k, y^k)] - \lambda_k [DJ_Y(y^{k-1}) - D_y K(x^{k-1}, y^{k-1})] \\
&\in DJ_Y(y^{k+1}) + \partial G_*(y^{k+1}) - 2D_y K(x^{k+1}, y^{k+1})
\end{aligned} \tag{38}$$

If X and Y are Hilbert spaces with $J_X = \tau^{-1}N_X$ and $J_Y = \sigma^{-1}N_Y$, the standard generating functions divided by some step length parameters $\tau, \sigma > 0$, and $K(x, y) = \langle Ax | y \rangle$ for $A \in \mathbb{L}(X; Y)$, (38) reduces to the inertial method of Chambolle and Pock (2015):

Inertial PDPS for bilinear K

With initial $\tilde{x}^0 = x^0$ and $\tilde{y}^0 = y^0$, iterate over $k \in \mathbb{N}$:

$$\begin{aligned}
x^{k+1} &:= \text{prox}_{\tau F}(\tilde{x}^k - \tau A^* \tilde{y}^k), \\
y^{k+1} &:= \text{prox}_{\sigma G_*}(\tilde{y}^k + \sigma A(2x^{k+1} - \tilde{x}^k)), \\
\tilde{x}^{k+1} &:= (1 + \lambda_{k+1})x^{k+1} - \lambda_{k+1}x^k, \\
\tilde{y}^{k+1} &:= (1 + \lambda_{k+1})y^{k+1} - \lambda_{k+1}y^k.
\end{aligned} \tag{39}$$

More generally, however, (38) does not directly apply inertia to the iterates. It applies inertia to K .

The general Cauchy inequality (35) automatically holds by the three-point identity (8) with $J''_{k+1} = J'_{k+1} = J^-_{k+1}$ if $B^-_{k+1} \geq 0$, which is to say that J^-_{k+1} is convex. This is the case if $\lambda_k \leq 0$. For usual inertia we, however, want $\lambda_k > 0$. We will therefore use Lemma 1, requiring:

Assumption 2. For some $\beta > 0$, in a domain $\Omega \subset X \times Y$

$$|(D_1 B^0(u^k, u)|u^k - u)| \leq B^0(u^k, u) + \beta B^0(u', u^k) \quad (u, u', u^k \in \Omega). \tag{40}$$

Moreover, the parameters $\{\lambda_k\}_{k \in \mathbb{N}}$ are non-increasing and for some $\epsilon > 0$

$$0 \leq \lambda_{k+1} \leq \frac{1 - \epsilon - \lambda_k \beta}{2} \quad (k \in \mathbb{N}). \tag{41}$$

Example 23. Suppose the generating function J^0 is γ -strongly subdifferentiable (i.e., B^0 is γ -elliptic, see sections “[Ellipticity of the Bregman Divergences](#)” and “[Ellipticity for Block-Adapted Methods](#)”) within $\Omega \subset X \times Y$ and satisfies the subdifferential smoothness property (10) with the factor $L > 0$. Then by Lemma 1, (40) holds with $\beta = L\gamma^{-1}$ in some domain $\Omega \subset X \times Y$.

As a particular case, let X and Y be Hilbert spaces with the standard generating functions $J_X = \tau^{-1}N_X$ and $J_Y = \sigma^{-1}N_Y$. Also let DK be L_{DK} -Lipschitz within Ω . Then J^0 is Lipschitz with factor $L = \max\{\sigma^{-1}, \tau^{-1}\} + L_{DK}$. Consequently the required subdifferential smoothness property (10) holds with the same factor L ; see Bauschke and Combettes (2017, Theorem 18.15) or Valkonen (2020, Appendix C).

We computed L_{DK} for some specific K in section “[Ellipticity of the Bregman Divergences.](#)”

Example 24. If $K(x, y) = \langle Ax|y \rangle$ with $A \in \mathbb{L}(X; Y^*)$, and if $J_X = \tau^{-1}N_X$, $J_Y = \sigma^{-1}N_Y$, in Hilbert spaces X and Y , then $B^0(u', u) = \frac{1}{2\tau}\|x - x'\|^2 + \frac{1}{2\sigma}\|y - y'\|^2 + \langle A(x - x')|y - y' \rangle$. By standard Cauchy inequality, (40) holds for $\beta = 1$ in $\Omega = X \times Y$. Consequently the next example recovers the upper bound for λ in Chambolle and Pock (2015):

Example 25. The bound (41) holds for some $\epsilon > 0$ if $\lambda_k \equiv \lambda$ for $0 \leq \lambda < 1/(2+\beta)$.

Lemma 8. *Suppose Assumption 2 holds and that (C²) holds within $\Omega_{\bar{u}}$ for some $\bar{u} \in \Omega$ and $\mathcal{G}(u, \bar{u})$. Given $u^0 \in \Omega$, suppose the iterates generated by the inertial PDBS (38) satisfy $\{u^k\}_{k=0}^N \subset \Omega_{\bar{u}} \cap \Omega$. Then*

$$\epsilon B^0(\bar{u}, u^N) + \epsilon \sum_{k=0}^{N-1} B^0(u^{k+1}, u^k) + \sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \bar{u}) \leq (1 - \lambda_1) B^0(\bar{u}, u^0). \tag{42}$$

Proof. Since $B_{k+1} = B^0$ and $B_{k+1}^- = -\lambda_k B^0$ for all $k \in \mathbb{N}$,

$$(B_{k+2} + B_{k+3}^-) - (B_{k+1} + B_{k+2}^-) = (\lambda_{k+1} - \lambda_{k+2})B^0.$$

Since λ_k is decreasing and B^0 is semi-elliptic within $\Omega \supset \{u^k, \bar{u}\}$, we deduce that $(\lambda_{k+1} - \lambda_{k+2})B^0(\bar{u}, u^k) \geq 0$. Consequently (IC) holds if (C) does. By the proof of Lemma 3, (IC) then holds if (C²) does. Using (40), (35) holds with $B'_{k+1} = \lambda_k B_0$ and $B''_{k+1} = \lambda_k \beta B_0$. Referring to Theorem 6, we now obtain (ID). We expand

$$\begin{aligned} [B_{N+1} + B_{N+2}^- - B''_{N+1}](\bar{u}, u^N) &= (1 - \lambda_{k+1} - \lambda_k \beta)B^0(\bar{u}, u^N) \quad \text{and} \\ [B_{k+1} + B_{k+2}^- - B''_{k+1} - B'_{k+2}](u^{k+1}, u^k) &= (1 - \lambda_{k+1} - \lambda_k \beta - \lambda_{k+1})B^0(u^{k+1}, u^k). \end{aligned}$$

Since $\bar{u}, u^k \in \Omega$ for all $k = 0, \dots, N$, using the ellipticity of B^0 within Ω as well as (41), we now estimate the first from below by $\epsilon B^0(\bar{u}, u^N)$ and the second by $\epsilon B^0(u^{k+1}, u^k)$. Thus (ID) produces (42). \square

We may now proceed as in sections “Convergence of Gaps in the Convex–Concave Setting” and “Convergence of Iterates” to prove convergence. For the verification of Assumption 2, we can use Examples 23, 24, and 25.

Theorem 7 (Convergence, inertial method). *Theorems 2, 3, and 5 apply to the iterates $\{u^{k+1}\}_{k \in \mathbb{N}}$ generated by the inertial PDBS (38) if we replace the assumptions of (semi-)ellipticity of B^0 with Assumption 2.*

Proof. We replace Lemma 3 and (D) by Lemma 8 and (42) in the proofs of Theorems 2, 3, and 5. Observe that Assumption 2 implies that B^0 is (semi-)elliptic. \square

Remark 7. The inertial PDPS is improved in Valkonen (2020) to yield *non-ergodic* convergence of the Lagrangian gap. To do the “inertial unrolling” that leads to such estimates, one, however, needs to correct for the anti-symmetry introduced by K into H .

Remark 8. Since Theorem 6 does not provide the quantitative Δ -Féjer monotonicity used in Theorem 4, we cannot prove linear convergence using our present simplified “testing” approach lacking the “testing parameters” of Valkonen (2020).

Improvements to the Basic Method Without Dual Affinity

We now have the tools to improve the basic PDBS (16) to enjoy prox-simple steps for general K not affine in y . Compared to (14) we amend $J_{k+1} = J^0$ by taking

$$\begin{aligned} J_{k+1}(x, y) &:= J_X(x) + J_Y(y) - K(x, y) + 2K(x^{k+1}, y) \\ &= J^0(x, y) + 2K(x^{k+1}, y). \end{aligned} \quad (43)$$

This would be enough for K to be explicit in the algorithm; however, proofs of convergence would practically require G_* to be strongly convex even in the convex-concave case. To fix this, we introduce the inertial term generated by

$$J_{k+1}^-(u) := [J^0 - J_k](u) = -2K(x^k, y). \quad (44)$$

As always, we write B_{k+1} , B^0 , and B_{k+1}^- for the Bregman divergences generated by J_{k+1} , J^0 , and J_{k+1}^- .

Since

$$D_1[B_{k+1} - B^0](u^k, u^{k-1}) + D_1 B_{k+1}^-(u^k, u^{k-1}) = (0, \tilde{y}_{k+1}^*)$$

for

$$\tilde{y}_{k+1}^* = 2[D_y K(x^{k+1}, y^{k+1}) - D_y K(x^{k+1}, y^k) - D_y K(x^k, y^k) + D_y K(x^k, y^{k-1})],$$

the algorithm (IPP) expands similarly to (16) as the

Modified PDBS

Iteratively over $k \in \mathbb{N}$, solve for x^{k+1} and y^{k+1} :

$$\begin{aligned} DJ_X(x^k) - D_x K(x^k, y^k) &\in DJ_X(x^{k+1}) + \partial F(x^{k+1}) \quad \text{and} \\ DJ_Y(y^k) + [2D_y K(x^{k+1}, y^k) + D_y K(x^k, y^k) - 2D_y K(x^k, y^{k-1})] \\ &\in DJ_Y(y^{k+1}) + \partial G_*(y^{k+1}). \end{aligned} \quad (45)$$

The method reduces to the basic PDBS (16) when K is affine in y . In Hilbert spaces X and Y with $J_X = \tau^{-1}N_X$ and $J_Y = \sigma^{-1}N_Y$, we can rearrange (45) as

Modified PDPS

Iterate over $k \in \mathbb{N}$:

$$\begin{aligned} x^{k+1} &:= \text{prox}_{\tau F}(x^k - \tau \nabla_x K(x^k, y^k)), \\ y^{k+1} &:= \text{prox}_{\sigma G_*}(y^k + \sigma[2\nabla_y K(x^{k+1}, y^k) + \nabla_y K(x^k, y^k) - 2\nabla_y K(x^k, y^{k-1})]). \end{aligned} \quad (46)$$

Remark 9. The modified PDPS (46) is slightly more complicated than the method in Clason et al. (2020), which would update

$$y^{k+1} := \text{prox}_{\sigma G_*}(y^k + \sigma \nabla_y K(2x^{k+1} - x^k, y^k)).$$

Likewise, (45) is different from the algorithm presented in Hamedani and Aybat (2018) for convex-concave K . It would, for the standard generating functions, update⁶

$$y^{k+1} := \text{prox}_{\sigma G_*}(y^k + \sigma[2\nabla_y K(x^{k+1}, y^k) - \nabla_y K(x^k, y^{k-1})]).$$

We could produce this method by taking $J_{k+1}^-(u) = -K(x^k, y)$. However, the convergence proofs would require some additional steps.

The main difference to the overall analysis of section “[Convergence Theory](#)” is in bounding from below the Bregman divergences in (ID). We now have

$$B_{N+1} + B_{N+2}^- - B_{N+1}'' = B^0 - B_{N+1}'' \quad \text{and} \quad (47a)$$

$$B_{k+1} + B_{k+2}^- - B_{k+1}'' - B_{k+2}' = B^0 - B_{k+1}'' - B_{k+2}'. \quad (47b)$$

If $D_y K(x^k, \cdot)$ is $L_{DK,y}$ -Lipschitz

$$\begin{aligned} \langle D_1 B_{k+1}^-(u^k, u) | u^k - u' \rangle &= 2 \langle D_y K(x^k, y^k) - D_y K(x^k, y) | y^k - y' \rangle \\ &\leq \sqrt{L_{DK,y}} \|y - y^k\|^2 + \sqrt{L_{DK,y}} \|y' - y^k\|^2 \\ &=: B_{k+1}'(u^k, u) + B_{k+1}''(u', u^k). \end{aligned} \quad (48)$$

Therefore, for the modified descent inequality (ID) to be meaningful, we require:

Assumption 3. We assume that $\|D_y K(x, y) - D_y K(x, y')\| \leq L_{DK,y} \|y - y'\|$ when $(x, y), (x, y') \in \Omega$ for some domain $\Omega \subset X \times Y$. Moreover, for some $\epsilon \geq 0$, we have

$$B^0(u, u') \geq \frac{\epsilon}{2} \|u - u'\|_{X \times Y}^2 + 2\sqrt{L_{DK,y}} \|y - y'\|_Y^2 \quad (u, u' \in \Omega). \quad (49)$$

We say that the present assumption holds *strongly* if $\epsilon > 0$.

⁶Note that Hamedani and Aybat (2018) uses the historical ordering of the primal and dual updates from Chambolle and Pock (2011), prior to the proof-simplifying discovery of the proximal point formulation in He and Yuan (2012). Hence our y^k is their y^{k+1} .

Example 26. If K is affine in y , $L_{DK,y} = 0$. Therefore, Assumption 3 reduces to the (semi-)ellipticity of B^0 , which can be verified as in sections “[Ellipticity of the Bregman Divergences](#)” and “[Ellipticity for Block-Adapted Methods](#).”

Example 27. Generally, it is easy to see that if one of the results of section “[Ellipticity of the Bregman Divergences](#)” holds with $\tilde{\sigma} = 1/(\sigma^{-1} - 4\sqrt{L_{DK,y}}) > 0$ in place of σ , then (49) holds. In particular, if K has L_{DK} -Lipschitz derivative within Ω , then Lemma 2 gives the condition $1 \geq L_{DK} \max\{\tau, \sigma/(1 - 4\sigma\sqrt{L_{DK,y}})\}$ and $1 > 4\sigma\sqrt{L_{DK,y}}$ for (49) to hold with $\epsilon = 0$. The assumption holds strongly if the first inequality is strict.

Similarly to Lemma 8, we now have the following replacement for Lemma 3:

Lemma 9. *Suppose Assumption 3 holds and (C^2) holds within $\Omega_{\bar{u}}$ for some $\bar{u} \in X \times Y$ and $\mathcal{G}(u, \bar{u})$. Given $u^0 \in X \times Y$, suppose the iterates generated by the modified PDBS (45) satisfy $\{u^k\}_{k=0}^N \subset \Omega_{\bar{u}}$. Then*

$$\epsilon B^0(\bar{u}, u^N) + \epsilon \sum_{k=0}^{N-1} B^0(u^{k+1}, u^k) + \sum_{k=0}^{N-1} \mathcal{G}(u^{k+1}, \bar{u}) \leq [B_1 + B_2^-](\bar{u}, u^0). \quad (50)$$

Proof. Inserting (43) and (44), (IC) reduces to (C), which follows from (C^2) as in Lemma 3. We verify (35) via (48) and Assumption 3. Thus Theorem 6 proves (ID). Inserting (47) and (49) with B'_{k+1} and B''_{k+1} from (48) into (ID) proves (50). \square

We may now proceed as in sections “[Convergence of Gaps in the Convex-Concave Setting](#)” and “[Convergence of Iterates](#)” to prove convergence. For the verification of Assumption 3, we can use Examples 26 and 27.

Theorem 8 (Convergence, modified method). *Theorems 2, 3, and 5 apply to the iterates $\{u^{k+1}\}_{k \in \mathbb{N}}$ generated by the modified PDBS (45) if we replace the assumptions of semi-ellipticity (resp. ellipticity) of B^0 with Assumption 3 holding (strongly).*

Proof. We replace Lemma 3 and (D) by Lemma 9 and (50) in Theorems 2, 3, and 5. Observe that (strong) Assumption 3 implies the (semi-)ellipticity of B^0 . \square

Now we have a locally convergent method (46) with easily implementable steps to tackle problems such as Potts segmentation (4) (Clason et al. 2020).

Further Directions

We close by briefly reviewing some things not covered, other possible extensions, and alternative algorithms.

Acceleration

To avoid technical detail, we did not cover $O(1/N^2)$ acceleration. The fundamental ingredients of proof are, however, exactly the same as we have used: sufficient second-order growth and ellipticity of the Bregman divergences B_k^0 , which are now iteration-dependent. Additionally, a portion of the second-order growth must be used to make the metrics B_k^0 grow as $k \rightarrow \infty$. For bilinear K in Hilbert spaces, such an argument can be found in Valkonen (2020); for $K(x, y) = \langle A(x)|y \rangle$ in Clason et al. (2019); and for general K in Clason et al. (2020). As mentioned in Remarks 1 and 9, the algorithms in the latter two differ slightly from the ones presented here.

Stochastic Methods

It is possible to refine the block-adapted (18) and its accelerated version into stochastic methods. The idea is to take on each step subsets of primal-blocks $S(i) \subset \{1, \dots, m\}$ and dual blocks $V(i+1) \subset \{1, \dots, n\}$ and to only update the corresponding x_j^{k+1} and y_ℓ^{k+1} . Full discussion of such technical algorithms is outside the scope of our present overview. We refer to Valkonen (2019) for an approach covering block-adapted acceleration and both primal and dual randomization in the case of bilinear K , but see also Chambolle et al. (2018) for a more basic version. For more general K affine in y , see Mazurenko et al. (2020).

Alternative Bregman Divergences

We have used Bregman divergences as a proof tool, in the end opting for the standard quadratic generating functions on Hilbert spaces. Nevertheless, our theory works for arbitrary Bregman divergences. The practical question is whether F and G_* remain prox-simple with respect to such a divergence. This can be the case for the “entropic distance” generated on $L^1(\Omega; [0, \infty))$ by

$$J(x) := \begin{cases} \int_{\Omega} x(t) \ln x(t) \, dt, & x \geq 0 \text{ a.e. on } \Omega, \\ \infty, & \text{otherwise} \end{cases}$$

See, for example, Burger et al. (2019) for a Landweber method (gradient descent on regularized least squares) based on such a distance.

Alternative Approaches

The derivative $D_1 B^0$ in (15) can be seen as a preconditioner, replacing $\tau(u - u')$ in the proximal point method (13). Our choice of B^0 is not the only option.

Consider the problem

$$\min_{x \in X} F(x) + E(x). \quad (51)$$

Provided E is differentiable and F prox-simple, i.e., the proximal map of F has a closed-form expression, (1) can be solved by forward-backward splitting methods as first introduced in Lions and Mercier (1979). In a Hilbert space X , this can be written

$$x^{k+1} := \text{prox}_{\tau F}(x^k - \tau \nabla E(x^k)). \quad (52)$$

Variants based on Bregman divergences were introduced in Nemirovski and Yudin (1983) under the name “mirror prox” or “mirror descent”; see also the review Chambolle and Pock (2016). The method and convergence proofs for it can be derived from our primal-dual approach. Indeed, if we take $G_* \equiv \delta_{\{0\}}$ as the indicator function of zero, and $K(x, y) = E(x)$ for some $E \in C^1(X)$, then (S) is equivalent to (51). Now the dual step of (17) is $y^{k+1} := 0$, and the primal step is (52).

Forward-backward splitting is especially popular under the name iterative soft thresholding (ISTA) in the context of sparse reconstruction (i.e., regularization of linear inverse problems with ℓ^1 penalties), see, e.g., Chambolle et al. (1998), Daubechies et al. (2004), and Beck and Teboulle (2009). However, forward-backward splitting has limited applicability in imaging and inverse problems due to the joint prox-simplicity and smoothness requirements. Sometimes these can be circumvented by considering so-called dual problems (Beck and Teboulle 2009).

Let then E be Gâteaux-differentiable and $F = G \circ A$ for a nonsmooth function F and a linear operator A in (51), i.e., consider the problem

$$\min_{x \in X} E(x) + G(Ax),$$

Forward-backward splitting is impractical as $G \circ A$ is in general not prox-simple. Assuming G to have the preconjugate G_* , we can write this problem as an instance of (S) with $F = 0$ and $K(x, y) = E(x) + \langle Ax | y \rangle$. Therefore the methods we have presented are applicable. However, in this instance, also $J^0(u) := \frac{1}{2} \|u\|_{X \times Y}^2 + \frac{1}{2} \|A^* y\|_{X^*}^2$ would produce an algorithm with realizable steps. In analogy to the PDPS, it might be called the primal-dual explicit spitting (PDES). The method was introduced in Loris and Verhoeven (2011) for $E(z) = \frac{1}{2} \|b - z\|^2$

as the “generalized iterative soft thresholding” (GIST), but has also been called the primal–dual fixed point method (PDFP, Chen et al. 2013) and the proximal alternating predictor corrector (PAPC, Drori et al. 2015).

The classical Augmented Lagrangian method solves the saddle point problem

$$\min_x \max_y F(x) + \frac{\tau}{2} \|E(x)\|^2 + \langle E(x)|y \rangle, \quad (53)$$

alternatingly for x and y . The alternating directions method of multipliers (ADMM) of Gabay (1983) and Arrow et al. (1958) takes $E(x) = Ax_1 + Bx_2 - c$ and $F(x) = F_1(x_1) + F_2(x_2)$ for $x = (x_1, x_2)$ and alternates between solving (53) for x_1 , x_2 , and y , using the most recent iterate for the other variables. The method cannot be expressed in our Bregman divergence framework, as the preconditioner $D_1 B_{k+1}(\cdot, x^k)$ would need to be nonsymmetric. The steps of the method are potentially expensive, each itself being an optimization problem. Hence the *preconditioned ADMM* of Zhang et al. (2011), which is equivalent to the PDPS, and the classical Douglas–Rachford splitting (DRS, Douglas and Rachford 1956) are applied to appropriate problems (Chambolle and Pock 2011; Clason and Valkonen 2020). The preconditioned ADMM was extended to nonlinear E in Benning et al. (2016).

Based on derivations avoiding the Lipschitz gradient assumption (cocoercivity) in forward–backward splitting, Malitsky and Tam (2018) moves the over-relaxation step $\bar{x}^{k+1} := 2x^{k+1} - x^k$ of the PDPS outside the proximal operators. This amounts to taking $J_{k+1}^- = \lambda_k K$ in section “Inertia (Almost) as Usually Understood” instead of $J_{k+1}^-(x, y) = \lambda_k J^0 = \lambda_k[\tau^{-1} J_X(x) + \sigma^{-1} J_Y(y) - K(x, y)]$, so is “partial inertia”; compare the “corrected inertia” of Valkonen (2020).

An over-relaxed variant of the same idea may be found in Bredies and Sun (2015). We have not discussed over-relaxation of entire algorithms. To briefly relate it to the basic inertia of (39), the latter “rebases” the algorithm at the inertial iterate \tilde{u}^k constructed from u^k and u^{k-1} , whereas over-relaxation would construct \tilde{u}^k from u^k and \tilde{u}^{k-1} . The derivation in Bredies and Sun (2015) is based on applying Douglas–Rachford splitting on a lifted problem. The basic over-relaxation of the PDPS is known as the Condat–Vũ method (Condat 2013; Vũ 2013).

Functions on Manifolds and Hadamard Spaces

The PDPS has been extended in Begmann et al. (2019) to functions on Riemannian manifolds: the problem $\min_{x \in \mathcal{M}} F(x) + G(Ex)$, where $E : \mathcal{M} \rightarrow \mathcal{N}$ with \mathcal{M} and \mathcal{N} Riemannian manifolds. In general, between manifolds, there are no linear maps, so E is nonlinear. Indeed, besides introducing a theory of conjugacy for functions on manifolds, the algorithm presented in Begmann et al. (2019) is based on the NL-PDPS of Valkonen (2014); Clason et al. (2019).

Convergence could only be proved on Hadamard manifolds, which are special: a type of three-point inequality holds (do Carmo 2013, Lemma 12.3.1). Indeed,

in even more general Hadamard spaces with the metric d , for any three points x^{k+1} , x^k , \bar{x} , we have (Bačák 2014, Corollary 1.2.5)

$$\frac{1}{2}d(x^k, x^{k+1})^2 + \frac{1}{2}d(x^{k+1}, \bar{x})^2 - \frac{1}{2}d(x^k, \bar{x})^2 \leq d(x^k, x^{k+1})d(\bar{x}, x^{k+1}). \quad (54)$$

Therefore, given a function f on such a space, to derive a simple proximal point algorithm, having constructed the iterate x^k , we might try to find x^{k+1} such that

$$f(x^{k+1}) + d(x^k, x^{k+1}) \leq f(x^k).$$

Multiplying this inequality by $d(\bar{x}, x^{k+1})$ and using the three-point inequality (54)

$$\frac{1}{2}d(x^k, x^{k+1})^2 + \frac{1}{2}d(x^{k+1}, \bar{x})^2 + [f(x^{k+1}) - f(x^k)]d(\bar{x}, x^{k+1}) \leq \frac{1}{2}d(x^k, \bar{x})^2.$$

If the space is bounded, $d(\bar{x}, x^{k+1}) \leq C$, so since $f(x^k) \geq f(x^{k+1})$, we may telescope and proceed as before to obtain convergence.

The Hadamard assumption is restrictive: if a Banach space is Hadamard, it is Hilbert, while a Riemannian manifold is Hadamard if it is simply connected with a non-positive sectional curvature (Bačák 2014, Section 1.2).

Glossary

The extended reals We define $\overline{\mathbb{R}} := [-\infty, \infty]$.
 A convex function A function $F : X \rightarrow \overline{\mathbb{R}}$ is convex if for all $x, x' \in X$ and $\lambda \in (0, 1)$, we have

$$F(\lambda x + (1 - \lambda)x') \leq F(\lambda x) + F((1 - \lambda)x').$$

A concave function A function $F : X \rightarrow \overline{\mathbb{R}}$ is concave if $-f$ is convex.
 A convex-concave function A function $K : X \times Y \rightarrow \overline{\mathbb{R}}$ is convex-concave if $K(\cdot, y)$ is convex for all $y \in Y$, and $K(x, \cdot)$ is concave for all $x \in X$.

The dual space We write X^* for the dual space of a topological vector (Banach, Hilbert) space X .

Set-valued map We write $A : X \rightrightarrows Y$ if A is a set-valued map between the spaces X and Y .

Derivative We write $DF : X \rightarrow X^*$ for the derivative of a Gâteaux-differentiable function $F : X \rightarrow \overline{\mathbb{R}}$.

Convex subdifferential This is the map $\partial F : X \rightrightarrows X^*$ for a convex $F : X \rightarrow \overline{\mathbb{R}}$. By definition $x^* \in \partial F(x)$ at $x \in X$ if and only if

$$F(x') - F(x) \geq \langle x^* | x' - x \rangle \quad (x' \in X).$$

Fenchel conjugate This is the function $f^* : X^* \rightarrow \overline{\mathbb{R}}$ defined for $F : X \rightarrow \overline{\mathbb{R}}$ by

$$f^*(x^*) := \sup_{x \in X} \langle x^*, x \rangle - F(x) \quad (x^* \in X^*).$$

Fenchel preconjugate If $X = (X_*)^*$ is the dual space of some space X_* and $F : X \rightarrow \overline{\mathbb{R}}$, then $f_* : X_* \rightarrow \overline{\mathbb{R}}$ is the preconjugate of f if $f = (f_*)^*$.

Proximal map For a function $F : X \rightarrow \overline{\mathbb{R}}$, this can be defined as

$$\text{prox}_F(x) := \arg \min_{\tilde{x} \in X} \left(F(\tilde{x}) + \frac{1}{2} \|\tilde{x} - x\|_X^2 \right).$$

Distributional derivative It arises from integration by parts: If $u : \mathbb{R}^n \supset \Omega \rightarrow \mathbb{R}$ is differentiable and $\phi \in C_c^\infty(\Omega; \mathbb{R}^n)$, then

$$\int_{\Omega} \langle \nabla u, \phi \rangle dx = - \int_{\Omega} u \operatorname{div} \phi dx.$$

If now u is not differentiable, we *define* the distribution $D \in C_c^\infty(\Omega; \mathbb{R}^n)^*$ by

$$Du(\phi) := - \int_{\Omega} u \operatorname{div} \phi dx.$$

If Du is bounded (as a linear operator), it can be presented as a vector Radon measure (Federer 1969), the space denoted $\mathcal{M}(\Omega; \mathbb{R}^n)$.

Indicator function For a set A , we define

$$\delta_A(x) := \begin{cases} 0, & x \in A, \\ \infty, & x \notin A. \end{cases}$$

References

- Ambrosio, L., Fusco, N., Pallara, D.: Functions of Bounded Variation and Free Discontinuity Problems. Oxford University Press (2000)
- Arridge, S.R., Kaipio, J.P., Kolehmainen, V., Tarvainen, T.: Optical imaging. In: Scherzer, O. (ed.) Handbook of Mathematical Methods in Imaging, pp. 735–780. Springer, New York (2011). https://doi.org/10.1007/978-0-387-92920-0_17
- Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in Linear and Non-linear Programming. Stanford University Press (1958)
- Bačák, M.: Convex Analysis and Optimization in Hadamard Spaces, Nonlinear Analysis and Applications. De Gruyter (2014)

- Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics, 2 edition. Springer (2017). <https://doi.org/10.1007/978-3-319-48311-5>
- Beck, A.: *First-Order Methods in Optimization*. SIAM (2017). <https://doi.org/10.1137/1.9781611974997>
- Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **18**, 2419–2434 (2009). <https://doi.org/10.1109/tip.2009.2028250>
- Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**, 183–202 (2009). <https://doi.org/10.1137/080716542>
- Begmann, R., Herzog, R., Tenbrück, D., Vidal-Núñez, J.: Fenchel duality for convex optimization and a primal dual algorithm on Riemannian manifolds (2019). arXiv:1908.02022
- Benning, M., Knoll, F., Schönlieb, C.B., Valkonen, T.: Preconditioned ADMM with nonlinear operator constraint. In: *System Modeling and Optimization: 27th IFIP TC 7 Conference, CSMO 2015, Sophia Antipolis, 29 June–3 July 2015, Revised Selected Papers*, pp. 117–126. Springer (2016). https://doi.org/10.1007/978-3-319-55795-3_10. arXiv:1511.00425
- Bredies, K., Sun, H.: Preconditioned Douglas–Rachford splitting methods for convex-concave saddle-point problems. *SIAM J. Numer. Anal.* **53**, 421–444 (2015). <https://doi.org/10.1137/140965028>
- Brezis, H., Crandall, M.G., Pazy, A.: Perturbations of nonlinear maximal monotone sets in Banach space. *Commun. Pure Appl. Math.* **23**, 123–144 (1970). <https://doi.org/10.1002/cpa.3160230107>
- Browder, F.E.: Convergence theorems for sequences of nonlinear operators in Banach spaces. *Mathematische Zeitschrift* **100**, 201–225 (1967). <https://doi.org/10.1007/bf01109805>
- Burger, M., Resmerita, E., Benning, M.: An entropic Landweber method for linear ill-posed problems (2019) arXiv:1906.10032
- Chambolle, A., DeVore, R.A., Lee, N.Y., Lucier, B.J.: Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process.* **7**, 319–335 (1998). <https://doi.org/10.1109/83.661182>
- Chambolle, A., Ehrhardt, M., Richtárik, P., Schönlieb, C.: Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM J. Optim.* **28**, 2783–2808 (2018). <https://doi.org/10.1137/17m1134834>
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011). <https://doi.org/10.1007/s10851-010-0251-1>
- Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Program.* 1–35 (2015). <https://doi.org/10.1007/s10107-015-0957-3>
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016). <https://doi.org/10.1017/s096249291600009x>
- Chen, P., Huang, J., Zhang, X.: A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Probl.* **29**, 025011 (2013). <https://doi.org/10.1088/0266-5611/29/2/025011>
- Chierchia, G., Chouzenoux, E., Combettes, P.L., Pesquet, J.C.: *The Proximity Operator Repository* (2019). <http://proximity-operator.net>. Online resource
- Clarke, F.: *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics (1990). <https://doi.org/10.1137/1.9781611971309>
- Clason, C., Mazurenko, S., Valkonen, T.: Acceleration and global convergence of a first-order primal-dual method for nonconvex problems. *SIAM J. Optim.* **29**, 933–963 (2019). <https://doi.org/10.1137/18m1170194>. arXiv:1802.03347
- Clason, C., Mazurenko, S., Valkonen, T.: Primal-dual proximal splitting and generalized conjugation in nonsmooth nonconvex optimization. *Appl. Math. Optim.* (2020). <https://doi.org/10.1007/s00245-020-09676-1>. arXiv:1901.02746
- Clason, C., Valkonen, T.: *Introduction to Nonsmooth Analysis and Optimization* (2020). arXiv:2001.00216. Work in progress

- Condat, L.: A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.* **158**, 460–479 (2013). <https://doi.org/10.1007/s10957-012-0245-9>
- Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**, 1413–1457 (2004). <https://doi.org/10.1002/cpa.20042>
- do Carmo, M.P.: *Riemannian Geometry. Mathematics: Theory & Applications.* Birkhäuser (2013)
- Douglas Jim, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. *Trans. Am. Math. Soc.* **82**, 421–439 (1956). <https://doi.org/10.2307/1993056>
- Drori, Y., Sabach, S., Teboulle, M.: A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Oper. Res. Lett.* **43**, 209–214 (2015). <https://doi.org/10.1016/j.orl.2015.02.001>
- Ekeland, I., Temam, R.: *Convex Analysis and Variational Problems.* SIAM (1999)
- Federer, H.: *Geometric Measure Theory.* Springer (1969)
- Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinski, R. (eds.) *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems.* Studies in Mathematics and Its Applications, vol. 15, pp. 299–331. North-Holland (1983)
- Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984). <https://doi.org/10.1109/tpami.1984.4767596>
- Hamedani, E.Y., Aybat, N.S.: A primal-dual algorithm for general convex-concave saddle point problems (2018). arXiv:1803.01401
- He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J. Imaging Sci.* **5**, 119–149 (2012). <https://doi.org/10.1137/100814494>
- Hiriart-Urruty, J.B., Lemaréchal, C.: *Fundamentals of Convex Analysis.* Grundlehren Text Editions. Springer (2004)
- Hohage, T., Homann, C.: A Generalization of the Chambolle-Pock Algorithm to Banach Spaces with Applications to Inverse Problems (2014). arXiv:1412.0126
- Hunt, A.: Weighing without touching: applying electrical capacitance tomography to mass flowrate measurement in multiphase flows. *Meas. Control* **47**, 19–25 (2014). <https://doi.org/10.1177/0020294013517445>
- Jauhainen, J., Kuusela, P., Seppänen, A., Valkonen, T.: Relaxed Gauss–Newton methods with applications to electrical impedance tomography. *SIAM J. Imaging Sci.* **13**, 1415–1445 (2020). <https://doi.org/10.1137/20m1321711>. arXiv:2002.08044
- Kingsley, P.: Introduction to diffusion tensor imaging mathematics: Parts I–III. *Concepts Magn. Reson. Part A* **28**, 101–179 (2006). <https://doi.org/10.1002/cmr.a.20048>
- Kuchment, P., Kunyansky, L.: Mathematics of photoacoustic and thermoacoustic tomography. In: Scherzer, O. (ed.) *Handbook of Mathematical Methods in Imaging*, pp. 817–865. Springer, New York (2011). https://doi.org/10.1007/978-0-387-92920-0_19
- Lions, P., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16**, 964–979 (1979). <https://doi.org/10.1137/0716071>
- Lipponen, A., Seppänen, A., Kaipio, J.P.: Nonstationary approximation error approach to imaging of three-dimensional pipe flow: experimental evaluation. *Meas. Sci. Technol.* **22**, 104013 (2011). <https://doi.org/10.1088/0957-0233/22/10/104013>
- Loris, I., Verhoeven, C.: On a generalization of the iterative soft thresholding algorithm for the case of non-separable penalty. *Inverse Probl.* **27**, 125007 (2011). <https://doi.org/10.1088/0266-5611/27/12/125007>
- Lustig, M., Donoho, D., Pauly, J.M.: Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.* **58**, 1182–1195 (2007). <https://doi.org/10.1002/mrm.21391>
- Malitsky, Y., Tam, M.K.: A forward-backward splitting method for monotone inclusions without cocoercivity (2018). arXiv:1808.04162

- Mazurenko, S., Jauhainen, J., Valkonen, T.: Primal-dual block-proximal splitting for a class of non-convex problems, *Electron. Trans. Numer. Anal.* **52**, 509–552 (2020). https://doi.org/10.1553/etna_vol52s509. arXiv:1911.06284
- Minty, G.J.: On the maximal domain of a “monotone” function. *Mich. Math. J.* **8**, 135–137 (1961)
- Nemirovski, A.S., Yudin, D.: *Problem Complexity and Method Efficiency in Optimization* (Translated from Russian). Wiley Interscience Series in Discrete Mathematics. Wiley (1983)
- Nishimura, D.: *Principles of Magnetic Resonance Imaging*. Stanford University (1996)
- Ollinger, J.M., Fessler, J.A.: Positron-emission tomography. *IEEE Signal Process. Mag.* **14**, 43–55 (1997). <https://doi.org/10.1109/79.560323>
- Opial, Z.: Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull. Am. Math. Soc.* **73**, 591–597 (1967). <https://doi.org/10.1090/s0002-9904-1967-11761-0>
- Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1762–1769. IEEE (2011). <https://doi.org/10.1109/iccv.2011.6126441>
- Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: 12th IEEE Conference on Computer Vision, pp. 1133–1140. IEEE (2009). <https://doi.org/10.1109/iccv.2009.5459348>
- Rockafellar, R.T.: *Convex Analysis*. Princeton University Press (1972)
- Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Optim.* **14**, 877–898 (1976). <https://doi.org/10.1137/0314056>
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
- Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**, 1019–1043 (2002). <https://doi.org/10.1137/s0036139900368844>
- Trucu, D., Ingham, D.B., Lesnic, D.: An inverse coefficient identification problem for the bio-heat equation. *Inverse Probl. Sci. Eng.* **17**, 65–83 (2009). <https://doi.org/10.1080/17415970802082880>
- Uhlmann, G.: Electrical impedance tomography and Calderón’s problem. *Inverse Probl.* **25**, 123011 (2009). <https://doi.org/10.1088/0266-5611/25/12/123011>
- Valkonen, T.: A primal-dual hybrid gradient method for non-linear operators with applications to MRI. *Inverse Probl.* **30**, 055012 (2014). <https://doi.org/10.1088/0266-5611/30/5/055012>. arXiv:1309.5032
- Valkonen, T.: Block-proximal methods with spatially adapted acceleration. *Electron. Trans. Numer. Anal.* **51**, 15–49 (2019). https://doi.org/10.1553/etna_vol51s15. arXiv:1609.07373
- Valkonen, T.: Inertial, corrected, primal-dual proximal splitting. *SIAM J. Optim.* **30**, 1391–1420 (2020). <https://doi.org/10.1137/18m1182851>. arXiv:1804.08736
- Valkonen, T.: Testing and non-linear preconditioning of the proximal point method. *Appl. Math. Optim.* **82** (2020). <https://doi.org/10.1007/s00245-018-9541-6>. arXiv:1703.05705
- Valkonen, T., Pock, T.: Acceleration of the PDHGM on partially strongly convex functions. *J. Math. Imaging Vis.* **59**, 394–414 (2017) <https://doi.org/10.1007/s10851-016-0692-2>. arXiv:1511.06566
- Vogel, C.R., Oman, M.E.: Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Process.* **7**, 813–824 (1998). <https://doi.org/10.1109/83.679423>
- Vũ, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.* **38**, 667–681 (2013). <https://doi.org/10.1007/s10444-011-9254-8>
- Zhang, X., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.* **46**, 20–46 (2011). <https://doi.org/10.1007/s10915-010-9408-8>

Part II
**Model- and Data-Driven Variational
Imaging Approaches**



Jonas Adler

Contents

Introduction	752
Deep Learning	754
Architectures	754
Gradient-Based Architectures	755
Proximal-Based Architectures	757
Primal-Dual Networks	758
Other Schemes	759
Training Procedure	760
Engineering Aspects	763
Architectures for Learned Operator	763
Initialization	763
Parameter Sharing	764
Further Memory	764
Preconditioning	765
Learned Step Length	765
Scalable Training	765
Putting It All Together	766
Conclusions	766
References	767

Abstract

Learned iterative reconstruction methods have recently emerged as a powerful tool to solve inverse problems. These deep learning techniques for image reconstruction achieve remarkable speed and accuracy by combining hard knowledge

J. Adler (✉)

Department of Mathematics, KTH – Royal Institute of Technology, Stockholm, Sweden
e-mail: jonasadl@kth.se

Now with DeepMind, London, UK

about the physics of the image formation process, represented by the forward operator, with soft knowledge about how the reconstructions should look like, represented by deep neural networks. A diverse set of such methods have been proposed, and this chapter seeks to give an overview of their similarities and differences, as well as discussing some of the commonly used methods to improve their performance.

Keywords

Inverse Problems · Deep Learning · Iterative reconstruction · Architectures

Introduction

Inference problems are ubiquitous in the sciences, medicine, and engineering. In these problems, we are given some form of data $y \in Y$ and aim to infer a result $x \in X$ from it. Typical examples include image classification where y is an image and x is a label and image segmentation where y is an image and x is a pointwise label. Inverse problems are a specific class of inference problems where we have access to additional structure. In particular, we assume the existence of a known forward operator $\mathcal{T}: X \rightarrow Y$ such that

$$\mathcal{T}x = y + \delta$$

where $\delta \in Y$ is a noise term with known distribution. The inference problem is hence reduced to inverting this relationship, a process we call reconstruction.

Deep learning techniques (LeCun et al. 2015; Goodfellow et al. 2016) using convolutional neural networks (CNN) (LeCun et al. 1989) have recently achieved state-of-the-art results in almost all fields of image processing (Krizhevsky et al. 2012), but until recently their application to image reconstruction has been limited. Several practical reasons for this can be claimed, notably lack of data, but perhaps one of the strongest reasons is that image reconstruction does not fit snugly into the standard problem formulation common to most image processing methods. In these problems, the input is an image, typically two-dimensional, and the output is also an image. The input and output images have a strong spatial relationship: A point in the input image corresponds to the same point in the output image, and if we translate the image, then the result should be translated as well (equivariance). These properties align perfectly with convolutions, whose use has been a major component in the deep learning revolution.

Neither of these properties hold in image reconstruction. Here, a point in the output image often depends globally on data from the input, and there is no trivial spatial relationship to use. In fact, in most interesting inverse problems, the input and output do not even belong to the same space. For example, in computed tomography, the input is a function on some set of lines through space, while the reconstruction

should be a scalar field in space. Since the input and output lives in different spaces, we cannot even perform standard linear operations on them, such as addition, much less hope that a convolution would take us from one to the other.

One way to solve this would be to generalize the concept of convolutions, and a significant effort has actually been spent on how to connect these spaces in mathematically rigorous ways. Notably the field of Fourier integral operators (FIO) (Hörmander 1971) has been developed, and these operators can be seen as generalizations of convolutions. However, the simple point-correspondence of convolutions breaks down, and instead we get a point-to-set correspondence, the canonical relation. Fourier integral operators are also notoriously complicated to work with and often computationally expensive. For this reason, the generalization of convolutional neural networks, perhaps FIO-neural networks (Feliu-Faba et al. 2019; Alizadeh et al. 2019), has so far not been applied to inverse problems.

While some have gone for a fully learned approach, ignoring the inherent symmetries, this does not seem to scale to realistic problem sizes (Zhu et al. 2018). Instead researchers have taken a middle way of incorporating more knowledge about the forward operator in a separate non-learned way into their learned reconstruction techniques. A very successful early approach has been to somehow convert the reconstruction problem into an image processing problem, which is easier than one might expect. Simply start by applying *any* reconstruction operator to the data to obtain a suboptimal initial reconstruction and then train a convolutional neural network to map the initial reconstruction to a more high-quality reconstruction (Jin et al. 2017; Kang et al. 2017).

While such methods incorporate significant components of the physics of the problem, encapsulated in the initial reconstruction, this also gives the methods a strong bias toward the result of the initial reconstruction, and in particular if there is any information lost in the initial reconstruction, it cannot be recovered by the post-processing.

An alternative, *learned iterative reconstruction*, has been developed in recent years. In learned iterative reconstruction, the physics of the problem is not seen as a separate component to be done prior to applying learning, but rather it is seen as an integral component of the learned reconstruction operator of equal footing with other commonly used components in neural networks such as convolutions and pointwise nonlinearities, thus allowing us to learn a reconstruction method acting on measured raw data.

This chapter will survey the development of these learned iterative reconstruction schemes and try to give an overview of architectures, training procedures, and practical and theoretical results. We note that several other high-quality review papers have looked at deep learning for inverse problems (Wang et al. 2018; McCann and Unser 2019; Arridge et al. 2019; Hammernik and Knoll 2020) and invite the reader to look at them for a broader overview of other techniques to use deep learning for image reconstruction.

Deep Learning

A (deep) neural network is a highly overparametrized function composed of several relatively simple parametrized components, often called layers, where both the components and their composition are differentiable. The exact choice of these components and how they are composed are called the architecture of the neural network, and the archetypical example is the standard feed-forward network which is a composition of parametrized affine operators and pointwise nonlinearities.

Training of a neural network $N_\theta : Y \rightarrow X$ is another term for selecting the parameters θ , and for inference problems, it is typically performed using some set of supervised training data $(y_i, x_i) \in Y \times X$ where the parameters are selected in order to minimize the empirical risk function

$$L(\theta) = \sum_i \ell(N_\theta(y_i), x_i)$$

where $\ell : X \times X \rightarrow \mathbb{R}$ is a loss function describing how close the result is to the ground truth. The networks are trained using some form of stochastic gradient descent over the parameters θ , which is made possible by the back-propagation algorithm (LeCun et al. 1989) which exploits the compositional structure of the networks to compute the gradient of the loss using only knowledge about the derivatives of the components. There is however a wide range of variations in how to train neural networks, as we will explore in section “[Training Procedure](#)”.

Architectures

Over the last years, a range of architectures for learned iterative reconstruction have been investigated, and although there have been steps toward it (Leuschner et al. 2019; Zbontar et al. 2018; Ramzi 2019), there is as of yet no consistent comparison of their performance in a benchmark, with each architecture sporting different upsides and downsides. We’ll here give a broad overview of the most common architectures used in the literature.

The core idea of learned iterative reconstruction is to interlace application of knowledge-driven operators, e.g., the forward operator, with learned operators such as convolutional neural networks. There are multiple ways to motivate specific learned iterative reconstruction architectures, but the most popular is to see them as neural network architectures inspired by unrolling of optimization solvers (Hershey et al. 2014). Specifically one notes that an optimization solver stopped after a finite number of iterations almost satisfies our conditions for a neural network (Banert et al. 2018). It is an operator that takes the data as input, processes it with simple components such as computing linear combinations and gradients, and returns a reconstruction. The individual components are also often differentiable, so the only thing missing is parametrizing the scheme so that there is something to learn.

There are many optimization problems to be inspired by and even more solvers. Learned iterative reconstruction methods can be broadly classified according to what type of optimization solver they were inspired by, and by now most commonly used classes of optimization solvers have been converted into reconstruction schemes. The learning on the other hand is introduced by replacing certain components, such as gradients or proximals, with learned counterparts in the form of neural networks.

Here we must stop and stress that the architectures are merely *inspired* by optimization solvers. Learned iterative reconstruction schemes do not actually try to solve any optimization problem as part of computing the reconstruction, not even approximately.

We'll now introduce some of the most common such constructions in a structured manner. We'll then follow up with various engineering tricks that have been found to sometimes vastly improve performance before finally turning to the training.

Gradient-Based Architectures

A set of very well-studied optimization problems are those associated with the maximum a posteriori solution given some prior. These optimization problems have been extensively explored over the years, including in both Tikhonov and total variation (TV) regularization. It can be studied using Bayes' theorem, according to which the posterior distribution $P(x | y)$ can be decomposed into components

$$P(x | y) = \frac{P(y | x)P(x)}{P(y)}.$$

Assuming that the posterior is differentiable, a gradient-based method to find the maximum of the log posterior can thus be written as in Algorithm 1.

Algorithm 1 Gradient Descent

```

1: Select  $x^0 \in X$ 
2: for  $n = 1, \dots$  do
3:    $x^n \leftarrow x^{n-1} + \alpha (\nabla_x \log P(y | x^{n-1}) + \nabla_x \log P(x^{n-1}))$ 
4: end for

```

Here we note that the data likelihood $P(y | x)$ is exactly known in inverse problems, whereas the prior $P(x)$ has to be chosen by the practitioner. The idea of gradient-based learned iterative reconstruction schemes is to unroll a gradient descent scheme to a finite number of iterations and then replace the gradient of the log-prior with a learned component $\Lambda_\theta : X \rightarrow X$ parametrized by parameters θ as in Algorithm 2.

As long as the operators $x \rightarrow \Lambda_\theta(x)$ and $x \rightarrow \log P(y | x)$ are differentiable, this yields a deep neural network which is end-to-end trainable. Interestingly,

Algorithm 2 Learned Gradient Descent

```

1: Select  $x^0 \in X$ 
2: for  $n = 1, \dots, N$  do
3:    $x^n \leftarrow x^{n-1} + \alpha \left( \nabla_x \log P(y | x^{n-1}) + \Lambda_\theta(x^{n-1}) \right)$ 
4: end for
5:  $N_\theta(y) \leftarrow x^N$ 

```

this very basic form of gradient-based learned iterative reconstruction was never published on its own, but a wide range of closely related schemes have been considered (Hauptmann et al. 2019; Chen et al. 2018).

Variational Networks

Variational networks (Hammernik et al. 2018) are a widely used class of gradient-based learned iterative reconstruction methods that more closely follow the inspiration from optimization than other schemes. In particular, the learned operator Λ_θ is required to be the gradient of some function which is learned

$$\Lambda_\theta(x) = \nabla_x h_\theta(x).$$

In the original papers, the functional $h_\theta : X \rightarrow \mathbb{R}$ was chosen to be of the form

$$h_\theta(x) = \sum_{k=1}^K \phi_{\theta_k}(K_{\theta_k}x)$$

where $\phi : X \rightarrow \mathbb{R}$ is a learnable nonlinear function averaged over the domain and K is a convolution kernel. Similar schemes could be obtained by using more expressive forms, e.g., a multilayer perceptron.

The use of an actual gradient gives some further interpretability of the scheme as minimization of a specific functional and the additional inductive bias helps reducing overfitting. However, since the functional h_θ is typically highly non-convex and since we stop after a finite number of steps, it is hard to exploit this to analyze, e.g., stability of the solution.

Variational networks have been applied to MRI reconstruction, both in the simplified setting of Fourier inversion and for the real nonlinear setting of multi-coil data (Knoll et al. 2019; Schlemper et al. 2019). In addition to this, it has been applied to a range of other imaging modalities such as CT (Hammernik et al. 2017; Vishnevskiy et al. 2019; Kobler et al. 2018) and ultrasound imaging (Vishnevskiy et al. 2018).

Proximal-Based Architectures

The proximal gradient algorithm (Parikh et al. 2014) is a method for solving convex optimization problems given by the sum of two functionals where only one of the functional is required to be differentiable; the other needs only to have a proximal operator defined. The method is an excellent fit for inverse problems since the log data likelihood is typically smooth while the prior is not.

Given a specific (log-)prior, the proximal operator can be seen as a backward gradient step and is given by

$$\text{prox}_{x \rightarrow -\alpha \log P(x)}(\hat{x}) = \arg \min_{x \in X} \left(\frac{1}{2} \|x - \hat{x}\|^2 - \alpha \log P(x) \right).$$

Using this, the proximal gradient algorithm, given in the setting of Bayesian inversion, is given in Algorithm 3.

Algorithm 3 Proximal Gradient

- 1: Select $x^0 \in X$
 - 2: **for** $n = 1, \dots$ **do**
 - 3: $x^n \leftarrow \text{prox}_{-\alpha \log P} \left(x^{n-1} + \alpha \nabla_x \log P(y | x^{n-1}) \right)$
 - 4: **end for**
-

As an opportunity for learning, we note that this is very similar to the gradient ascent scheme except that instead of an additive gradient, the proximal of the log-prior acts on the updated point. The corresponding learned iterative reconstruction scheme can be obtained by replacing the proximal operator by a learned component.

Algorithm 4 Learned Proximal Gradient

- 1: Select $x^0 \in X$
 - 2: **for** $n = 1, \dots, N$ **do**
 - 3: $x^n \leftarrow A_\theta \left(x^{n-1} + \alpha \nabla_x \log P(y | x^{n-1}) \right)$
 - 4: **end for**
 - 5: $N_\theta(y) \leftarrow x^N$
-

This type of scheme was first published under the name *recurrent inference machines* with applications to image processing problems (Putzky and Welling 2017). Several other papers extended the methods by adding further components but also by applying the method to CT (Adler and Öktem 2017; Gupta et al. 2018) MRI (Lønning et al. 2018) and photoacoustic tomography (Hauptmann et al. 2018; Yang et al. 2019).

We should however note that there is a different way to use the proximal gradient scheme. In particular, it is sometimes the case that the proximal of the data log-

likelihood $x \rightarrow -\log P(y | x)$ is easy to compute. This is the case in most image processing problems but also in MRI. If the data likelihood is differentiable, one can use the proximal gradient scheme given in Algorithm 5.

Algorithm 5 Projected Gradient

```

1: Select  $x^0 \in X$ 
2: for  $n = 1, \dots$  do
3:    $x^n \leftarrow \text{prox}_{x \rightarrow -\alpha \log P(y|x)} \left( x^{n-1} + \alpha \nabla_x \log P(x^{n-1}) \right)$ 
4: end for

```

This scheme often has very fast convergence since the proximal (depending on the data likelihood) can be seen as a projection onto the feasible set in a single iteration. For this reason, the algorithm is sometimes called projected gradient descent, and we'll adopt that name here for disambiguation purposes. Algorithms such as ADMM can also be seen as variations of the general idea.

One can then introduce learning as usual by replacing the knowledge-driven prior with a data-driven prior as in Algorithm 6.

Algorithm 6 Learned Projected Gradient

```

1: Select  $x^0 \in X$ 
2: for  $n = 1, \dots, N$  do
3:    $x^n \leftarrow \text{prox}_{x \rightarrow -\alpha \log P(y|x)} \left( x^{n-1} + \Lambda_\theta(x^{n-1}) \right)$ 
4: end for
5:  $N_\theta(y) \leftarrow x^N$ 

```

This class of algorithms has become very popular in MRI reconstruction due to their ease of implementation and speed improvements over gradient-based schemes. In this domain, the proximal step is often called a *data consistency* term, since the proximal enforces the result to be (approximately) consistent with the data (Schlemper et al. 2017; Aggarwal et al. 2018; Kofler et al. 2018). One of the first learned iterative reconstruction schemes, ADMM-Net (Sun et al. 2016) used a related approach for MRI reconstruction, and a range of works have followed with some interesting variations (Mardani et al. 2017a,b, 2018), and there has even been some analysis of their convergence (Schwab et al. 2018).

Primal-Dual Networks

Primal-dual proximal splitting optimization methods are another class of optimization schemes applicable to inverse problems. Specifically, if the data likelihood can be written

Algorithm 7 Primal-Dual

```

1: Select  $x^0 \in X, z^0 \in Y$ 
2: for  $n = 1, \dots$  do
3:    $y^n \leftarrow \text{prox}_{-\alpha(\log \mathcal{L})^*} (y^{n-1} + \mathcal{T}x^{n-1})$ 
4:    $x^n \leftarrow \text{prox}_{-\alpha \log P} (x^{n-1} - \mathcal{T}^*y^n)$ 
5: end for

```

Algorithm 8 Learned Primal-Dual

```

1: Select  $x^0 \in X, z^0 \in Y$ 
2: for  $n = 1, \dots$  do
3:    $y^n \leftarrow \Gamma_\theta (y^{n-1} + \mathcal{T}x^{n-1})$ 
4:    $x^n \leftarrow \Lambda_\theta (x^{n-1} - \mathcal{T}^*y^n)$ 
5: end for
6:  $N_\theta(y) \leftarrow x^N$ 

```

$$P(y | x) = \mathcal{L}(y | \mathcal{T}x),$$

then the problem can be solved using a proximal-based scheme with only knowledge about the proximal of the functional $\mathcal{T}x \rightarrow -\log P(y | \mathcal{T}x)$. The most simple of such scheme, the Arrow-Hurwicz algorithm (Arrow et al. 1958), is given in Algorithm 7. Accelerated versions of the scheme using momentum, including the primal-dual hybrid gradient algorithm (Chambolle and Pock 2011), are very popular for optimization in inverse problems due to their speed and versatility.

Following the recipe from before, we can convert the primal-dual algorithm into a learned scheme by replacing the proximals with learned operators. Here one could replace only the proximal related to the prior or both proximals, but most authors prefer to learn both and this gives rise to the learned primal dual scheme, as in Algorithm 8.

Given the versatility of this kind of algorithm, practically only requiring access to the forward operator, it can be applied to almost any inverse problem. So far, applications have been to CT (Adler and Öktem 2018b; Wu et al. 2018, 2019b), possibly with incomplete data (Zhang et al. 2019), and image processing (Vogel and Pock 2017).

Other Schemes

To round off our expose on classical methods that have been converted into learned iterative reconstruction schemes, we note that some authors have found their inspiration in iterative schemes outside of optimization. One such idea is Neumann networks (Gilton et al. 2019), which gain inspiration from the Neumann series for the inverse

$$\mathcal{T}^{-1} = \sum_{n=1}^{\infty} (I - \eta \mathcal{T}^* \mathcal{T})^n \eta \mathcal{T}^*$$

where $\eta < \|\mathcal{T}^* \mathcal{T}\|$ is a step length. The authors view the partial sums as an iteration and add a learning component as a small offset to the update, which leads to Algorithm 9.

Algorithm 9 Neumann Network

```

1:  $x^0 \leftarrow \eta \mathcal{T}^* y$ 
2:  $\hat{x}^0 \leftarrow x^0$ 
3: for  $n = 1, \dots$  do
4:    $x^n \leftarrow x^{n-1} - \eta \mathcal{T}^* \mathcal{T} x^{n-1} - \eta \Lambda_{\theta}(x^{n-1})$ 
5:    $\hat{x}^n \leftarrow \hat{x}^{n-1} + x^n$ 
6: end for
7:  $N_{\theta}(y) \leftarrow \hat{x}^N$ 

```

We note that the algorithm is very similar to a gradient-based scheme, but that the result is given as the sum of all partial iterates, and that the data only enters in the beginning.

Others have taken inspiration from classical iterative reconstruction schemes, e.g., the Landweber algorithm (Aspri et al. 2018), and there is nothing to stop researchers from using other methods such as conjugate gradient in the future.

Training Procedure

Given an architecture, the next step is to select the optimal parameters. The definition of what's meant by "optimal" is however a hot area for both research and debate. By far the most popular definition of "optimal" for neural networks in general and learned iterative reconstruction schemes in particular is to view the problem as an inference problem where the data is seen as a sample from a random variable y , and we seek to infer the unknown signal which is a sample from another random variable, x . Our training data is seen as N samples (y_n, x_n) from the joint random variable (y, x) . Further, as in the introduction, we introduce a loss function $\ell : X \rightarrow X$ which characterizes how good a single reconstruction is. Given all of this, the optimal parameter choice is defined as the parameters which minimize the risk function

$$L(\theta) = \mathbb{E} \ell(N_{\theta}(y), x).$$

Since the risk involves an expectation over the random variables y and x , which we don't have access to since they should represent all possible inputs/outputs, we need to approximate it using our training data. Thankfully, the sample mean is an

unbiased estimator for the expectation, so we can instead chose to minimize the empirical risk function

$$\hat{L}(\theta) = \sum_{n=1}^N \ell(N_{\theta}(y_n), x_n).$$

This paradigm is known as empirical risk minimization. The next problem is to find a minimizer to $\hat{L}(\theta)$ and this is nontrivial, especially given the scale of both training data and networks in current practice. The machine learning community has converged on approximately solving the optimization problem using variations of stochastic gradient descent (LeCun et al. 1989; Kingma and Ba 2014).

A significant problem in implementing backpropagation for learned iterative reconstruction is that while there are well-maintained libraries for computing gradients of standard neural network components using automatic differentiation (Abadi et al. 2016; Paszke et al. 2017), these very rarely implement, e.g., the Radon transform. Many researchers solved this by wrapping other implementations of these operators such as ASTRA (van Aarle et al. 2015) using some glue library, e.g., ODL (Adler et al. 2017a). While this is very versatile and allows easy comparison to classical reconstruction algorithms, there are some performance downsides. Some have therefore implemented tomographic operators with native backpropagation (Syben et al. 2019), and there is considerable interest toward differentiable programming, a paradigm that would allow backpropagation through any operator (Innes et al. 2019; Bradbury et al. 2018).

Since we only have access to a finite amount of training data, empirical risk minimization will lead to *overfitting* to our available data, e.g., the optimal parameter choice for our training data will differ to the optimal choice for all possible inputs, and parameters that minimize the empirical risk will not be optimal for the expected risk. Classical statistical learning theory (and intuition) tells us that the more parameters we have, the more we will overfit to our training data, although this relation has been called into question for deep learning. Learned iterative reconstruction is often seen to have advantageous properties here since the number of parameters is typically much smaller than fully learned methods.

The choice of loss function can also have a significant impact on the learned operator, and authors have proposed a wide range of options. Thankfully, there is actually quite some theory related to the properties of various optimal reconstruction operators under a choice of loss which we can use to guide our choice of loss function. For example, with the squared norm $\ell(N_{\theta}(y), x) = \|N_{\theta}(y) - x\|^2$, the optimal will be the minimum mean squared error estimator, which is simply the conditional expectation $\mathbb{E}[x | y]$. This implies that a neural network trained with this loss should approximate the conditional expectation. Likewise, it is known that the optimal reconstruction given the 1-norm loss is the conditional median. Even more intricate losses have been investigated in the literature, e.g., when training with a Wasserstein loss, the optimal reconstruction is a *spatial* average of the posterior (Adler et al. 2017b).

Some authors have looked further than these relatively simple losses and have looked toward using neural networks to define a loss function. The earliest such attempts were to use perceptual losses (Johnson et al. 2016), which consider an image as good if it looks like the true image according to a neural network. The definition of “looks like the true image” is taken to have similar intermediate activations and the neural network typically taken to be a ImageNet classifier. This approach has been applied to CT and MRI denoising, where it gave more visually appealing results (Yang et al. 2017a,b, 2018).

A related type of loss is adversarial losses (Goodfellow et al. 2014), where one trains a neural network to judge how good a reconstruction is. In the most simple setting, a *discriminator* network is trained to determine if an image is a reconstruction or a true image, and the reconstruction operator is trained to generate true-looking images. In order to make sure that the network returns a reconstruction that is related to the input, one typically combines this with some form of classical loss and sometimes a cycle-consistency (data-fit) condition. The latter case is especially interesting, since it allows training without paired training data (Mardani 2017; Lei et al. 2019).

Another way of using a neural network to define the loss is to ask “how useful is the reconstruction?”, where we define usefulness by how well another network can be trained on the reconstruction to solve some task (Adler et al. 2018). This general and straightforward idea can be applied to practically any downstream task, but initial work has focused on segmentation (Boink et al. 2019), object detection (Wu et al. 2018), and classification (Efland et al. 2018; Diamond et al. 2017).

All of the above methods (possibly excluding adversarial losses) require supervised training data. However, access to this kind of data, especially in large amounts, is often a luxury. Many hence see training using unsupervised data as something of a grand challenge in order to get truly scalable learned iterative reconstruction that is applicable in practice. Some algorithmic advances have been made in this direction, notably the Noise2Noise (Lehtinen et al. 2018) method which uses the fact that when trained with squared norm loss, the result should only depend on the conditional mean of the data. Hence, it is possible to train using noisy ground truth samples, and the learned reconstruction should approximate their mean. Other methods have been developed with the same goal, e.g., the SURE estimator (Raphan and Simoncelli 2007). These methods have just started being used for image reconstruction, but with promising results (Soltanayev and Chun 2018; Cha et al. 2019).

Finally there is great potential in combining learned reconstruction with advances in deep generative models in order to achieve true Bayesian reconstruction methods where one can sample from the posterior distribution instead of computing a single estimator (Adler and Öktem 2018a; Anonymous 2020). Such methods are especially relevant in the low signal/high noise setting, such as ultralow dose CT and dynamic imaging or for highly complicated imaging modalities such as seismic imaging (Herrmann et al. 2019).

To conclude, supervised training with simple losses is still by far the most popular way to train learned iterative reconstruction schemes, but their combination of expressive power, speed, and versatility allows a huge range of other options for training, and we can only expect this field to grow in the future.

Engineering Aspects

While the inspiration from optimization is important to the performance of learned iterative reconstruction, experience (Hessel et al. 2018) tells us that deep learning is highly sensitive to engineering and implementation choices and that including these can significantly improve performance. Learned iterative reconstruction methods have not turned out to be an exception, and considerable effort has been put into finding the best implementations. We'll here try to give a broad overview of these methods.

Architectures for Learned Operator

All learned iterative reconstruction schemes reduce learning the $Y \rightarrow X$ reconstruction operator into learning a $X \rightarrow X$ (and possibly also $Y \rightarrow Y$) operator such as a learned gradient or a learned proximal. This type of operator can be represented by a standard “off-the-shelf” convolutional neural network without any problems. Many authors have found a small, e.g., one to three layer, neural network to be sufficient for the task (Adler and Öktem 2017; Chen et al. 2018; Diamond et al. 2017; Mardani et al. 2017a), and some have decided to use more complicated architectures (Putzky and Welling 2017; Hauptmann et al. 2018), typically converging on some reduced version of the U-Net (Ronneberger et al. 2015). These networks are almost universally combined with the technique of residual learning (He et al. 2016; Jin et al. 2017) where the learned operators are of the form $A_\theta(x) = x + \hat{A}_\theta(x)$ with \hat{A}_θ a feed forward network.

As a general rule of thumb, for “simple” inverse problems such as fully sampled MRI or CT, small networks seem to work very well, while for more complicated inverse problems such as photoacoustic tomography or ultrasound, a larger network might be needed. However, larger networks typically require more training data to avoid overfitting.

Initialization

Just like optimization, all learned iterative reconstruction methods begin with an initial estimate x^0 which is then refined. Since only a finite number of steps are used, it's reasonable to expect this choice to have quite significant impact on the

final result. Authors have converged on two different initialization schemes. These are zero-initialization (Adler and Öktem 2018b), $x^0 = 0$, and pseudo-inverse initialization $x^0 = \mathcal{T}^\dagger y$, where $\mathcal{T}^\dagger : Y \rightarrow X$ is some pseudo-inverse, e.g., zero filled Fourier inversion (Hammernik et al. 2018) or filtered back projection (Adler et al. 2017b). In some cases where the forward operator is approximately unitary, e.g., in photoacoustic tomography, the adjoint has been used in place of a pseudo-inverse (Hauptmann et al. 2018). Some have also tried learning some parameters of the initial reconstruction, e.g., learning the filters in filtered back projection (Hammernik et al. 2017). These more advanced initialization schemes have possible speed and accuracy advantages over zero-initialization since the learned operator only needs to learn a correction from the initial reconstruction, but they run a risk of overfitting to the initial reconstruction, giving worse generalization.

Parameter Sharing

The algorithms as presented here have been shown with a single learned gradient/proximal operator that is used in all iterations. However, it has been found by several authors that a significant improvement can be obtained by relaxing this requirement and instead learning a different operator Λ_{θ^n} for each iteration, where the full parameter vector is $\theta = [\theta^1, \theta^2, \dots, \theta^N]$. For example, Adler and Öktem (2018b) reports a very noticeable 4.5 dB uplift when learning ten different proximals instead of one.

The reason for this uplift has not been thoroughly explained, but the most simple explanation is that it gives the network ten times more learned parameters. However, making a single proximal ten times larger has not been found to give the same uplift, so perhaps the explanation lies in the ability of different parts of the network to focus on different tasks, with early iterations focusing on large-scale structure while the last iterations finalize the finer structures.

Further Memory

Several optimization algorithms contain some concept of memory, e.g., momentum, which helps the algorithms by giving information from more points than the current point. Given the very high representative power of deep learning methods, one would expect that this type of additional information would be very useful to learned iterative reconstruction methods as well.

Several authors have explored this concept (Putzky and Welling 2017; Adler et al. 2017b; Adler and Öktem 2018b), typically having an extra “momentum” term in X^n for $n \approx 5$ which is updated alongside the reconstruction. These papers claim improvements, but it is also clear that many others opt not to use any further memory in their algorithms (Hammernik et al. 2018). It is hence not fully clear how large the benefit of using memory is.

Preconditioning

Since learned iterative reconstruction is inspired by optimization, it is perhaps not surprising that improvements to optimization schemes can be applied here as well. One particular such method is preconditioning, which is widely used to speed up optimization solvers. Several authors have investigated using such ideas in learned iterative reconstruction typically using preconditioners of the form of a regularized inverse (Gilton et al. 2019; Diamond et al. 2017; Aggarwal et al. 2018)

$$(\mathcal{T}^*\mathcal{T} + \lambda I)^{-1}.$$

However, this is only feasible when the above operator is easily computed, which is only really the case for image processing problems and Fourier inversion. Others have used approximations by, e.g., filtering (Hauptmann et al. 2019) or diagonal approximations to the Hessian (Ravishankar et al. 2019). Finally, some have investigated other optimization-based ways of speeding up convergence, e.g., Nesterov momentum (Li et al. 2018).

Learned Step Length

While learned iterative reconstruction exploits knowledge about the gradient or proximal of the data likelihood, the standard derivations typically give rise to algorithms with a step length that has to be selected by the user. Given that we're already learning large parts of the reconstruction, several authors have looked into learning this step length as well. There are two main ways of doing this. The most simple is to simply consider the step length as part of the learnable parameters and learn it along with the other parameters (Sun et al. 2016; Hammernik et al. 2018). A somewhat more intricate method is to learn to combine the gradient with the current iteration (Putzky and Welling 2017; Adler et al. 2017b). For example, in the learned proximal gradient scheme, one could use an update of the form

$$x^n \leftarrow \Lambda_\theta \left(x^{n-1}, \nabla_x \log P(y | x^{n-1}) \right)$$

where $\Lambda_\theta : X^2 \rightarrow X$ in this case. This should have some upsides in that the network could in theory learn, e.g., a preconditioner. Similar ideas can be applied to most proximal-based learned iterative schemes, e.g., learned primal-dual (Adler and Öktem 2018b).

Scalable Training

While learned iterative reconstruction schemes are at least an order of magnitude faster to evaluate than classical optimization-based reconstruction methods, training

them using the backpropagation algorithm (LeCun et al. 1989) is extremely memory intense since every step of the algorithm has to be stored in memory. For this reason, researchers have had significant issues in scaling the algorithms beyond slice-by-slice cases of roughly 512^2 pixels.

A method to train on full 3d volumes of about 512^3 voxels hence either needs a very expensive supercomputer (Laanait et al. 2019) or to be trained without standard backpropagation. Several researchers have investigated the latter. One such method is to train the network one iteration at a time, which significantly reduces the amount of memory needed (Hauptmann et al. 2018; Wu et al. 2019a). Another method is to use gradient checkpointing (Chen et al. 2016) which reduces the amount of memory used by recomputing on the fly. An extreme case of this is invertible networks (Dinh et al. 2014; Jacobsen et al. 2018) which totally remove the need for storing intermediate results, enabling 3d reconstruction (Putzky et al. 2019)

Putting It All Together

It is common to combine several, if not all, of the above ideas in a single algorithm. To give a more practical example in CT, let us assume that \mathcal{T} is the radon transform and that we have Gaussian noise, in which case $\log P(y | x) = \frac{1}{2} \|y - \mathcal{T}x\|^2$. A learned iterative reconstruction scheme for this inverse problem using the learned proximal gradient method can be obtained by combining pseudo-inverse initialization with initialization with avoiding parameter sharing, learned steps, extra memory, and preconditioning which should give a state-of-the-art reconstruction method. Most parts are straightforward, except for the choice of preconditioner. Here one could use that due to the Fourier slice theorem, the inverse Hessian $(\mathcal{T}^* \mathcal{T})^{-1}$ can be approximated by a convolution with a sharpening kernel K . Using this, we arrive at Algorithm 10 which is a state-of-the-art learned iterative reconstruction algorithm.

Algorithm 10 Learned Proximal Gradient with engineering improvements

```

1:  $x^0 \leftarrow [\mathcal{T}^\dagger y, 0, \dots, 0] \in X^M$ 
2: for  $n = 1, \dots, N$  do
3:    $x^n \leftarrow \Lambda_{\theta^n} \left( x^{n-1}, K \mathcal{T}^* (\mathcal{T} x_1^{n-1} - y) \right)$ 
4: end for
5:  $N_\theta(y) \leftarrow x_1^N$ 

```

Conclusions

Learned iterative reconstruction has attracted significant interest in just a few years, and research has quickly gone from a wild-west of architecture exploration to a more structured view. Given the enormous success of deep learning methods in general in solving supervised learning problems, research has started shifting toward new

frontiers. The first is moving into more practicably applicable domains, where we need to learn from large amounts of data without a ground truth and with various artifacts. The second frontier is the ability to solve previously unsolvable problems such as reconstructing the posterior distribution or integrating reconstruction with image analysis tasks. A final frontier is to gain a theoretical understanding of why these algorithms work so well. Some steps toward this has been taken (Effland et al. 2019; Mardani et al. 2019), but there is still a huge gap between theory and practice. I suspect that we will see an explosive development in this field in the coming years and can only hope that this chapter can serve as an introduction to its many possibilities in the future.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283 (2016)
- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Prob.* **33**(12), 124007 (2017)
- Adler, J., Öktem, O.: Deep Bayesian Inversion. arXiv1811.05910 (2018a)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018b)
- Adler, J., Kohr, H., Öktem, O.: ODL-A Python Framework for Rapid Prototyping in Inverse Problems. Royal Institute of Technology (2017a)
- Adler, J., Ringh, A., Öktem, O., Karlsson, J.: Learning to Solve Inverse Problems Using Wasserstein Loss. arXiv1710.10898 (2017b)
- Adler, J., Lunz, S., Verdier, O., Schönlieb, C.B., Öktem, O.: Task Adapted Reconstruction for Inverse Problems. arXiv1809.00948 (2018)
- Aggarwal, H.K., Mani, M.P., Jacob, M.: MoDL: model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imaging* **38**(2), 394–405 (2018)
- Alizadeh, K., Farhadi, A., Rastegari, M.: Butterfly Transform: An Efficient FFT Based Neural Architecture Design. arXiv1906.02256 (2019)
- Anonymous: Closed loop deep Bayesian inversion: uncertainty driven acquisition for fast MRI. In: Submitted to International Conference on Learning Representations (2020). <https://openreview.net/forum?id=BJIPOIBKDB>. Under review
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.B.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Arrow, K.J., Hurwicz, L., Uzawa, H.: *Studies in Linear and Non-linear Programming*. Stanford University Press, Stanford (1958)
- Aspri, A., Banert, S., Öktem, O., Scherzer, O.: A Data-Driven Iteratively Regularized Landweber Iteration. arXiv1812.00272 (2018)
- Banert, S., Ringh, A., Adler, J., Karlsson, J., Öktem, O.: Data-Driven Nonsmooth Optimization. arXiv1808.00946 (2018)
- Boink, Y.E., Manohar, S., Brune, C.: A Partially Learned Algorithm for Joint Photoacoustic Reconstruction and Segmentation. arXiv1906.07499 (2019)
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Wanderman-Milne, S.: JAX: composable transformations of Python+NumPy programs (2018). <http://github.com/google/jax>
- Cha, E., Jang, J., Lee, J., Lee, E., Ye, J.C.: Boosting CNN Beyond Label in Inverse Problems. arXiv1906.07330 (2019)

- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Chen, T., Xu, B., Zhang, C., Guestrin, C.: Training Deep Nets with Sublinear Memory Cost. arXiv1604.06174 (2016)
- Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., Wang, G.: LEARN: learned experts' assessment-based reconstruction network for sparse-data CT. *IEEE Trans. Med. Imaging* **37**(6), 1333–1347 (2018)
- Diamond, S., Sitzmann, V., Boyd, S., Wetzstein, G., Heide, F.: Dirty Pixels: Optimizing Image Classification Architectures for Raw Sensor Data. arXiv1701.06487 (2017)
- Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear Independent Components Estimation. arXiv1410.8516 (2014)
- Effland, A., Hölzel, M., Klatzer, T., Kobler, E., Landsberg, J., Neuhäuser, L., Pock, T., Rumpf, M.: Variational networks for joint image reconstruction and classification of tumor immune cell interactions in melanoma tissue sections. In: *Bildverarbeitung für die Medizin 2018*, pp. 334–340. Springer (2018)
- Effland, A., Kobler, E., Kunisch, K., Pock, T.: An Optimal Control Approach to Early Stopping Variational Methods for Image Restoration. arXiv preprint arXiv:1907.08488 (2019)
- Feliu-Faba, J., Fan, Y., Ying, L.: Meta-learning Pseudo-differential Operators with Deep Neural Networks. arXiv1906.06782 (2019)
- Gilton, D., Ongie, G., Willett, R.: Neumann Networks for Inverse Problems in Imaging. arXiv1901.03707 (2019)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, London (2016)
- Gupta, H., Jin, K.H., Nguyen, H.Q., McCann, M.T., Unser, M.: CNN-based projected gradient descent for consistent CT image reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1440–1453 (2018)
- Hammernik, K., Knoll, F.: Machine learning for image reconstruction. In: *Handbook of Medical Image Computing and Computer Assisted Intervention*, pp. 25–64. Elsevier, London (2020)
- Hammernik, K., Würfl, T., Pock, T., Maier, A.: A deep learning architecture for limited-angle computed tomography reconstruction. In: *Bildverarbeitung für die Medizin 2017*, pp. 92–97. Springer (2017)
- Hammernik, K., Klatzer, T., Kobler, E., Recht, M.P., Sodickson, D.K., Pock, T., Knoll, F.: Learning a variational network for reconstruction of accelerated mri data. *Magn. Reson. Med.* **79**(6), 3055–3071 (2018)
- Hauptmann, A., Lucka, F., Betcke, M., Huynh, N., Adler, J., Cox, B., Beard, P., Ourselin, S., Arridge, S.: Model-based learning for accelerated, limited-view 3-d photoacoustic tomography. *IEEE Trans. Med. Imaging* **37**(6), 1382–1393 (2018)
- Hauptmann, A., Adler, J., Arridge, S., Öktem, O.: Multi-Scale Learned Iterative Reconstruction. arXiv1908.00936 (2019)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
- Herrmann, F.J., Siahkoohi, A., Rizzuti, G.: Learned Imaging with Constraints and Uncertainty Quantification. arXiv1909.06473 (2019)
- Hershey, J.R., Roux, J.L., Weninger, F.: Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures. arXiv1409.2574 (2014)
- Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., Silver, D.: Rainbow: combining improvements in deep reinforcement learning. In: *Thirty-Second AAAI Conference on Artificial Intelligence* (2018)
- Hörmander, L.: Fourier integral operators. I. *Acta Math.* **127**(1), 79–183 (1971)
- Innes, M., Edelman, A., Fischer, K., Rackauckus, C., Saba, E., Shah, V.B., Tebbutt, W.: Zygote: A Differentiable Programming System to Bridge Machine Learning and Scientific Computing. arXiv1907.07587 (2019)

- Jacobsen, J.H., Smeulders, A., Oyallon, E.: i-Revnet: Deep Invertible Networks. arXiv1802.07088 (2018)
- Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **26**(9), 4509–4522 (2017)
- Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: *European Conference on Computer Vision*, pp. 694–711. Springer (2016)
- Kang, E., Min, J., Ye, J.C.: A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction. *Med. Phys.* **44**(10), e360–e375 (2017)
- Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv1412.6980 (2014)
- Knoll, F., Hammernik, K., Zhang, C., Moeller, S., Pock, T., Sodickson, D.K., Akcakaya, M.: Deep Learning Methods for Parallel Magnetic Resonance Image Reconstruction. arXiv1904.01112 (2019)
- Kobler, E., Muckley, M., Chen, B., Knoll, F., Hammernik, K., Pock, T., Sodickson, D., Otazo, R.: Variational deep learning for low-dose computed tomography. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6687–6691. IEEE (2018)
- Kofler, A., Haltmeier, M., Kolbitsch, C., Kachelrieß, M., Dewey, M.: A u-nets cascade for sparse view computed tomography. In: *International Workshop on Machine Learning for Medical Image Reconstruction*, pp. 91–99. Springer (2018)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
- Laanait, N., Romero, J., Yin, J., Young, M.T., Treichler, S., Starchenko, V., Borisevich, A., Sergeev, A., Matheson, M.: Exascale Deep Learning for Scientific Inverse Problems. arXiv1909.11150 (2019)
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**(4), 541–551 (1989)
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
- Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning Image Restoration Without Clean Data. arXiv1803.04189 (2018)
- Lei, K., Mardani, M., Pauly, J.M., Vasawanala, S.S.: Wasserstein GANs for MR Imaging: From Paired to Unpaired Training. arXiv1910.07048 (2019)
- Leuschner, J., Schmidt, M., Bager, D.O., Maaß, P.: The LoDoPaB-CT Dataset: A Benchmark Dataset for Low-Dose CT Reconstruction Methods. arXiv1910.01113 (2019)
- Li, H., Yang, Y., Chen, D., Lin, Z.: Optimization Algorithm Inspired Deep Neural Network Structure Design. arXiv1810.01638 (2018)
- Lønning, K., Putzky, P., Caan, M.W., Welling, M.: Recurrent Inference Machines for Accelerated MRI Reconstruction. arXiv (2018)
- Mardani, L.L.M.: Semi-supervised super-resolution GANs for MRI. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach (2017)
- Mardani, M., Gong, E., Cheng, J.Y., Vasanawala, S., Zaharchuk, G., Alley, M., Thakur, N., Han, S., Dally, W., Pauly, J.M., et al.: Deep Generative Adversarial Networks for Compressed Sensing Automates MRI. arXiv1706.00051 (2017a)
- Mardani, M., Monajemi, H., Pappas, V., Vasanawala, S., Donoho, D., Pauly, J.: Recurrent Generative Adversarial Networks for Proximal Learning and Automated Compressive Image Recovery. arXiv1711.10046 (2017b)
- Mardani, M., Gong, E., Cheng, J.Y., Vasanawala, S.S., Zaharchuk, G., Xing, L., Pauly, J.M.: Deep generative adversarial neural networks for compressive sensing MRI. *IEEE Trans. Med. Imaging* **38**(1), 167–179 (2018)
- Mardani, M., Sun, Q., Pappas, V., Vasanawala, S., Pauly, J., Donoho, D.: Degrees of Freedom Analysis of Unrolled Neural Networks. arXiv preprint arXiv:1906.03742 (2019)
- McCann, M.T., Unser, M.: Algorithms for Biomedical Image Reconstruction. arXiv1901.03565 (2019)
- Parikh, N., Boyd, S., et al.: Proximal algorithms. *Found. Trends Optim.* **1**(3), 127–239 (2014)

- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
- Putzky, P., Welling, M.: Recurrent Inference Machines for Solving Inverse Problems. arXiv1706.04008 (2017)
- Putzky, P., Karkaloulos, D., Teuwen, J., Miriakov, N., Bakker, B., Caan, M., Welling, M.: i-RIM Applied to the fastMRI Challenge. arXiv1910.08952 (2019)
- Ramzi, Z.: fastMRI reproducible benchmark. <https://github.com/zacharieramzi/fastmri-reproducible-benchmark> (2019)
- Raphan, M., Simoncelli, E.P.: Learning to be Bayesian without supervision. In: Advances in Neural Information Processing Systems, pp. 1145–1152 (2007)
- Ravishankar, S., Ye, J.C., Fessler, J.A.: Image Reconstruction: From Sparsity to Data-Adaptive Methods and Machine Learning. arXiv1904.02816 (2019)
- Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
- Schlemper, J., Caballero, J., Hajnal, J.V., Price, A., Rueckert, D.: A deep cascade of convolutional neural networks for mr image reconstruction. In: International Conference on Information Processing in Medical Imaging, pp. 647–658. Springer (2017)
- Schlemper, J., Salehi, S.S.M., Kundu, P., Lazarus, C., Dyvorne, H., Rueckert, D., Sofka, M.: Nonuniform variational network: deep learning for accelerated nonuniform MR image reconstruction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 57–64. Springer (2019)
- Schwab, J., Antholzer, S., Haltmeier, M.: Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Prob.* <https://iopscience.iop.org/article/10.1088/1361-6420/aaf14a> (2018)
- Soltanayev, S., Chun, S.Y.: Training deep learning based denoisers without ground truth data. In: Advances in Neural Information Processing Systems, pp. 3257–3267 (2018)
- Sun, J., Li, H., Xu, Z., et al.: Deep ADMM-Net for compressive sensing MRI. In: Advances in Neural Information Processing Systems, pp. 10–18 (2016)
- Syben, C., Michen, M., Stimpel, B., Seitz, S., Ploner, S., Maier, A.K.: PYRO-NN: Python Reconstruction Operators in Neural Networks. arXiv1904.13342 (2019)
- van Aarle, W., Palenstijn, W.J., De Beenhouwer, J., Altantzis, T., Bals, S., Batenburg, K.J., Sijbers, J.: The ASTRA Toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicroscopy* **157**, 35–47 (2015)
- Vishnevskiy, V., Sanabria, S.J., Goksel, O.: Image reconstruction via variational network for real-time hand-held sound-speed imaging. In: International Workshop on Machine Learning for Medical Image Reconstruction, pp. 120–128. Springer (2018)
- Vishnevskiy, V., Rau, R., Goksel, O.: Deep Variational Networks with Exponential Weighting for Learning Computed Tomography. arXiv1906.05528 (2019)
- Vogel, C., Pock, T.: A primal dual network for low-level vision problems. In: German Conference on Pattern Recognition, pp. 189–202. Springer (2017)
- Wang, G., Ye, J.C., Mueller, K., Fessler, J.A.: Image reconstruction is a new frontier of machine learning. *IEEE Trans. Med. Imaging* **37**(6), 1289–1296 (2018)
- Wu, D., Kim, K., Dong, B., El Fakhri, G., Li, Q.: End-to-end lung nodule detection in computed tomography. In: International Workshop on Machine Learning in Medical Imaging, pp. 37–45. Springer (2018)
- Wu, D., Kim, K., El Fakhri, G., Li, Q.: Computational-efficient cascaded neural network for CT image reconstruction. In: Medical Imaging 2019: Physics of Medical Imaging, vol. 10948, p. 109485Z. International Society for Optics and Photonics (2019a)
- Wu, D., Kim, K., Kalra, M.K., De Man, B., Li, Q.: Learned primal-dual reconstruction for dual energy computed tomography with reduced dose. In: 15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine, vol. 11072, p. 1107206. International Society for Optics and Photonics (2019b)

- Yang, G., Yu, S., Dong, H., Slabaugh, G., Dragotti, P.L., Ye, X., Liu, F., Arridge, S., Keegan, J., Guo, Y., et al.: DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1310–1321 (2017a)
- Yang, Q., Yan, P., Kalra, M.K., Wang, G.: CT Image Denoising with Perceptive Deep Neural Networks. *arXiv1702.07019* (2017b)
- Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M.K., Zhang, Y., Sun, L., Wang, G.: Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE Trans. Med. Imaging* **37**(6), 1348–1357 (2018)
- Yang, C., Lan, H., Gao, F.: Accelerated photoacoustic tomography reconstruction via recurrent inference machines. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6371–6374. IEEE (2019)
- Zbontar, J., Knoll, F., Sriram, A., Muckley, M.J., Bruno, M., Defazio, A., Parente, M., Geras, K.J., Katsnelson, J., Chandarana, H., et al.: FastMRI: An Open Dataset and Benchmarks for Accelerated MRI. *arXiv1811.08839* (2018)
- Zhang, H., Dong, B., Liu, B.: JSR-Net: a deep network for joint spatial-radon domain CT reconstruction from incomplete data. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3657–3661. IEEE (2019)
- Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R., Rosen, M.S.: Image reconstruction by domain-transform manifold learning. *Nature* **555**(7697), 487 (2018)



An Analysis of Generative Methods for Multiple Image Inpainting

20

Coloma Ballester, Aurélie Bugeau, Samuel Hurault,
Simone Parisotto, and Patricia Vitoria

Contents

Introduction	774
A Walk Through the Image Inpainting Literature	775
How to Achieve Multiple and Diverse Inpainting Results?	778
Generative Adversarial Networks	780
Variational Autoencoders and Conditional Variational Autoencoders	784
Autoregressive Models	788
Image Transformers	790
From Single-Image Evaluation Metrics to Diversity Evaluation	793
Experimental Results	795
Experimental Settings	796
Quantitative Performance	797
Qualitative Performance	803
Conclusions	809
Appendix	809
Additional Quantitative Results	809
Additional Qualitative Results	810
References	813

C. Ballester (✉) · P. Vitoria
IPCV, DTIC, University Pompeu Fabra, Barcelona, Spain
e-mail: coloma.ballester@upf.edu; patricia.vitoria@upf.edu

A. Bugeau
LaBRI, CNRS, Université de Bordeaux, Talence, France

Institut universitaire de France (IUF), Paris, France
e-mail: aurelie.bugeau@labri.fr

S. Hurault
Bordeaux INP, CNRS, IMB, Université de Bordeaux, Talence, France
e-mail: samuel.hurault@math.u-bordeaux.fr

S. Parisotto
DAMTP, University of Cambridge, Cambridge, UK
e-mail: sp751@cam.ac.uk

Abstract

Image inpainting refers to the restoration of an image with missing regions in a way that is not detectable by the observer. The inpainting regions can be of any size and shape. This is an ill-posed inverse problem that does not have a unique solution. In this work, we focus on learning-based image completion methods for multiple and diverse inpainting which goal is to provide a set of distinct solutions for a given damaged image. These methods capitalize on the probabilistic nature of certain deep generative models to sample various solutions that coherently restore the missing content. Throughout the chapter, we will analyze the underlying theory and analyze the recent proposals for multiple inpainting. To investigate the pros and cons of each method, we present quantitative and qualitative comparisons, on common datasets, regarding both the quality and the diversity of the set of inpainted solutions. Our analysis allows us to identify the most successful generative strategies in both inpainting quality and inpainting diversity. This task is closely related to the learning of an accurate probability distribution of images. Depending on the dataset in use, the challenges that entail the training of such a model will be discussed through the analysis.

Keywords

Inverse problems · Inpainting · Multiple inpainting · Diverse inpainting · Deep learning · Generative methods

Introduction

Image inpainting, also called *amodal completion* or *disocclusion* in early days, is an active research area in many fields including applied mathematics and computer vision, with foundations in the Gestalt theory of shape perception. Inpainting relates to the virtual reconstruction of missing content in images in a way that is non-detectable by the observer (Bertalmío et al. 2000). It is an ill-posed inverse problem that can have multiple plausible solutions. Indeed, the fact that the inpainted image is not unique can be understood both mathematically and also because the reconstruction quality is judged by independent humans. On top of that, it has a strong impact on many real-life applications, e.g., in medical imaging (sinograms (Tovey et al. 2019), CT scans (Chen et al. 2012)), 3D surface data (Biasutti et al. 2019; Bevilacqua et al. 2017; Hervieu et al. 2010; Parisotto et al. 2020), art conservation (frescoes (Batz et al. 2008), panel paintings (Ružić et al. 2011) and manuscripts (Calatroni et al. 2018)), image compression (Peter and Weickert 2015), camera artifact removal (Vitoria and Ballester 2019), and the restoration of old movies and videos (Grossauer 2006; Newson et al. 2014), just to name a few.

State-of-the-art image inpainting methods have achieved amazing results regarding the complex work of filling large missing areas in an image. However, most of

the methods generally attempt to generate one single result from a given image, ignoring many other plausible solutions. In this chapter, we focus on analyzing recent advances in the inpainting literature, concentrating on the learning-based approaches for *multiple* and *diverse* inpainting. The goal of those methods is to estimate multiple plausible inpainted solutions while being as much diverse as possible. Those methods mainly focus on the idea of exploiting image coherency at several levels along with the power of neural networks trained on large datasets of images. Unlike previous one-to-one methods, multiple-image inpainting offers the advantage of exploring a large space of possible solutions. This procedure gives capacity to the user to eventually choose the preferred fit under his/her judgment instead of leaving the task of singling out one solution to the algorithm itself.

This chapter is structured as follows: Section “[A Walk Through the Image Inpainting Literature](#)” provides a brief overview of both model-based and learning-based inpainting methods in the literature. Section “[How to Achieve Multiple and Diverse Inpainting Results?](#)” presents the underlying theory of several approaches for multiple and diverse inpainting together with a review of the most representative (to the best of our knowledge) state-of-the-art proposals using those particular strategies. Section “[From Single-Image Evaluation Metrics to Diversity Evaluation](#)” presents the evaluation metrics for both inpainting quality and diverse inpainting. The multiple inpainting results of the methods of section “[How to Achieve Multiple and Diverse Inpainting Results?](#)” are presented and compared in section “[Experimental Results](#)” both quantitatively and qualitatively, on common datasets and masks, concerning three aspects: proximity to ground truth, perceptual quality, and inpainting diversity. Finally, section “[Conclusions](#)” concludes the presented analysis.

A Walk Through the Image Inpainting Literature

In the literature, inpainting methods can fall under different categories, e.g., *local vs. nonlocal* depending on the ability to capture and exploit non-nearby content, or *geometric vs. exemplar-based methods* depending on the action on points or patches. For our purposes, it is more convenient to distinguish between learning- and model-based approaches, according to the usage or not of machine learning techniques. For extensive reviews of existing inpainting methods, we refer the reader to the works in Guillemot and Le Meur (2014), Schonlieb (2015), Buyskens et al. (2015), and Parisotto et al. (2022).

Model-Based Inpainting

Model-based inpainting methods are designed to manipulate an image by exploiting its regularity and coherency features with an explicit model governing the inpainting workflow. One approach for restoring geometric image content is to locally propagate the intensity values and regularity of the image level lines inward the inpainting domain with curvature-driven (Nitzberg et al. 1993; Masnou and

Morel 1998; Ballester et al. 2001; Chan and Shen 2001; Esedoglu and Shen 2002; Shen et al. 2003) and diffusion-based (Caselles et al. 1998; Shen and Chan 2002; Tschumperle and Deriche 2005) evolutionary partial differential equations (PDEs), possibly of fluid-dynamic nature (Bertalmío et al. 2000, 2001; Tai et al. 2007) or with coherent transport mechanisms (Bornemann and März 2007), also by invoking variational principles (Grossauer and Scherzer 2003; Bertozzi et al. 2007) and regularization (possibly of higher order) priors (Papafitsoros and Schönlieb 2013). The filling-in of geometry, especially of small scratches and homogeneous content in small inpainting domains, is the most effective scenario of these methods, which perform poorly in the recovery of texture. Such issue is overcome by considering a patch (a group of neighboring points in the image domain) as the imaging atom containing the essential texture element. The variational formulation of dissimilarity metrics based on the estimation of a correspondence map between patches (Efros and Leung 1999; Bornard et al. 2002; Demanet et al. 2003; Criminisi et al. 2004; Aujol et al. 2010) has led to the design of optimal copying-pasting strategies for inpainting large domains. However, these methods still fail, e.g., in the presence of different scale-space features. Thus, some researchers have exploited, also using a variational approach, the efficiency of PatchMatch (Barnes et al. 2009) in computing a probabilistic approximation of correspondence maps between patches to average the contribution of multiple-source patches during the synthesis step. For example, Arias et al. (2011) and Newson et al. (2014) use it in a non-local mean fashion (Wexler et al. 2004), to inpaint rescaled versions of the original image with results propagated from the coarser to the finer scale; Cao et al. (2011) to guide the inpainting with geometric-sketches; Sun et al. (2005) to guide structures; or Mansfield et al. (2011), Eller and Fornasier (2016), and Fedorov et al. (2016) to account for geometric transformations of patches. However, these mathematical and numerical advances may result to be computationally expensive while suffering from having only one single-imaging source as input, and dependence on the initialization quality and the selection of associated parameters (e.g., the size of the patch). Thus, it seems natural to study if image coherency, smoothness, and self-similarity patterns can be further exploited by augmenting the dataset of source images and eventually synthesize multiple inpainting solutions: this is where diverse inpainting with deep learning-based generative approaches is a significant step forward.

One of the earliest model-based inpainting works dealing with multiple-source images is Kang et al. (2002), where salient landmarks are extracted in a scene under different perspectives and then synthesized by interpolation, guiding the imaging restoration. As said, model-based models are sensitive to initializations and chosen parameters: One way to diminish these drawbacks is to perform inpainting of the input image multiple times, by varying parameters like the patch size, the number of pyramid scales, initializations, and inpainting methodologies. Thus, a final assembling step will produce an inpainted image, which encodes locally the most coherent content (Hays and Efros 2007; Le Meur et al. 2013; Kumar et al. 2016). Still, the computational effort of estimating several solutions with different parameters and their fine-tuning is a keypoint, leading to the need for a

one-encompassing strategy that can locally adapt the synthesis step from multiple-source images. This task can be solved with learning-based methods.

Learning-Based Methods

Learning-based methods address image inpainting by learning a mapping from a corrupted input to the estimated restoration by training on a large-scale dataset. Besides capturing local or non-local regularities and redundancy inside the image or the entire dataset, those methods also exploit high-level information inherent in the image itself, such as global regularities and patterns, or perceptual clues and semantics over the images.

Early learning-based methods tackled the problem as a blind inpainting problem (Ren et al. 2015; Cai et al. 2015) by minimizing the distance between the predicted image and the ground truth. This type of methods behaved as an image denoising algorithm and was limited to tiny inpainting domains. To deal with bigger and more realistic inpainting regions, later approaches incorporated in the model the information provided by the mask, e.g., Köhler et al. (2014), Ren et al. (2015), Pathak et al. (2016), and Lempitsky et al. (2018). Also, several modifications to vanilla convolutions have been proposed to explicitly use the information of the mask, like partial convolutions (Liu et al. 2018) and gated convolutions (Yu et al. 2019), where the output of those layers only depends on non-corrupted points. Additionally, attempts to increase the receptive field without increasing the number of layers have been proposed with dilated convolutions (Iizuka et al. 2017; Wang et al. 2018) and contextual attention (Yu et al. 2018, 2019). Learning to inpaint in a single step has shown to be a complex endeavor. Progressive learning approaches have also been introduced to split the learning into several steps: for instance, Zhang et al. (2018a) progressively fills the holes from outside to inside; similarly, Guo et al. (2019), Zeng et al. (2020), and Li et al. (2020) also learn how to update the inpainting mask for next iteration, and Li et al. (2019) learns jointly structure and feature information.

To train the network, early approaches minimized some distance between the ground-truth and the predicted image. But this approach takes into account just one of the several possible plausible solutions to the inpainting problem. Several approaches have been proposed to overcome this drawback. Some works use perceptual metrics based on generative adversarial networks (GANs) aiming to generate more perceptually realistic results (Pathak et al. 2016; Yeh et al. 2017; Iizuka et al. 2017; Yu et al. 2018; Vitoria et al. 2019, 2020; Dapogny et al. 2020; Liu et al. 2019; Lahiri et al. 2020). Other works tackle the problem in the feature space by minimizing distances at feature space level (Fawzi et al. 2016; Yang et al. 2017; Vo et al. 2018) by using an additional pre-trained network, or by directly inpainting those features (Yan et al. 2018; Zeng et al. 2019). Also, two-step approaches have been proposed. They are based on a first coarse inpainting (Yang et al. 2017; Yu et al. 2018; Liu et al. 2019), edge learning (Liao et al. 2018; Nazeri et al. 2019; Li et al. 2019), or structure prediction (Xiong et al. 2019; Ren et al. 2019) and followed by a refinement step adding finer texture details. Furthermore, Liu et al. (2020) aimed to ensure consistency between structure and texture generation. Another big problem

of early deep learning methods is that deep models treat input images with limited resolution. While first approaches were able to deal with images of maximum size 64×64 , the latest methods can deal with 1024×1024 resolution images by using, for example, a multiscale approach (Yang et al. 2017; Zeng et al. 2019), or even to $8K$ resolution by generating first a low-resolution solution and second its high-frequency residuals (Yi et al. 2020b).

Recent works (e.g., Zheng et al. 2019; Zhao et al. 2020b; Cai and Wei 2020; Peng et al. 2021; Wan et al. 2021; Liu et al. 2021) deal with the ill-posed nature of the problem by allowing more than one possible plausible solution to a given image. They aim to generate multiple and diverse solutions by using deep probabilistic models based on variational autoencoders (VAEs), GANs, autoregressive models, transformers, or a combination of them. Note that those types of methods have been also used for real case applications such as diverse fashion image inpainting (Han et al. 2019) and Cosmic microwave background radiation (CMB) image inpainting (Yi et al. 2020a). Besides, it is worth mentioning that there are several single-image generation methods that estimate complete images with some variations. For instance, SinGAN (Rott Shaham et al. 2019) produces several random images which are deviations of an input image by learning the distribution of its patches. Park et al. (2019) synthesizes new images by controlling style and semantics. However, these strategies do not completely fit within the multiple inpainting problems where regions of the image are known and should not be changed. In this chapter, we will focus on the study of multiple-image inpainting methods. More precisely, we will review, analyze, and compare, theoretically as well as experimentally, the different approaches proposed on the literature to generate inpainting diversity.

How to Achieve Multiple and Diverse Inpainting Results?

In this section, we will describe the different tools and methods that successfully addressed multiple image inpainting. Later in section “[Experimental Results](#)”, we will conduct a thorough experimental study comparing these methods visually and quantitatively.

As previously mentioned, image inpainting is an inverse problem with multiple plausible solutions. Generally, ill-posed problems are solved by incorporating some knowledge or priors into the solution. Mathematically, this is frequently done using a variational approach where a prior is added to a data-fidelity term to create an overall objective functional that is lastly optimized. The selected prior promotes the singling out of a particular solution. Traditionally, the incorporated priors were model-based, founded on properties of the expected solution.

More recently, data-driven proposals have emerged where the prior knowledge on the image distribution is implicitly or explicitly learned via neural networks optimization (we refer to the recent survey Arridge et al. 2019 and references therein). Among them, generative methods have been used to learn the underlying geometric and semantic priors of a set of non-corrupted images. Indeed, generative methods aim to estimate the probability distribution of a large set, \mathcal{X} , of data. In

other words, any $x \in \mathcal{X}$ is assumed to come from an underlying and unknown probability distribution $\mathbb{P}_{\mathcal{X}}$, and the goal is to learn it from the data in \mathcal{X} . Due to its capacity to produce several outcomes given a single output, some authors have proposed to address the multiplicity of solutions by leveraging the probabilistic nature of generative models.

Through this chapter, we will assume that \mathcal{X} is a set of images. Images will be assumed to be functions defined on a bounded domain $\Omega \subset \mathbb{R}^2$ with values in \mathbb{R}^C , with $C = 1$ for gray-level images and $C = 3$ for color images. With a slight abuse of notation, we will use the same notation to refer to the continuous setting, where $\Omega \subset \mathbb{R}^2$ is an infinite resolution image domain and $x : \Omega \rightarrow \mathbb{R}^C$ represents a continuous image, and to the discrete setting where Ω stands for a discrete domain given by a grid of $H \times W$ pixels, $H, W \in \mathbb{N}$, and x is a function defined on this discrete Ω and with values in \mathbb{R}^C . In the latter case, x is usually given in the form of a real-valued matrix of size $H \times W \times C$ representing the image values.

In the context of image inpainting, the inpainting domain, denoted here by O , represents the region of the image domain Ω where the image data is missing, and thus to be restored. Its complementary set, $O^c = \Omega \setminus O$, represents the region of Ω where the values of the image to be inpainted are known. The inpainting mask M will be defined as equal to 1 on the missing pixels of O and equal to 0 on $\Omega \setminus O$.

The space \mathcal{X} of (complete) natural images is a high-dimensional space, and its distribution can be very complex. However, natural images contain local regularities, non-local self-similarities, global coherency, and even semantic structure. This is one of the reasons that inspired the use of latent-based models. These models use latent variables $z \in \mathcal{Z}$ in a lower-dimensional space $\dim(\mathcal{Z}) \leq \dim(\mathcal{X})$, associated with a probability distribution $\mathbb{P}_{\mathcal{Z}}$. Generative latent-based models aim to learn a generative model $G_{\theta} : \mathcal{Z} \rightarrow \mathcal{X}$, with parameters θ , mapping a latent variable z to an image x . With a slight abuse of notation and if it is understood from the context, we will forget about the θ subindex and simply write G . The main goal of this strategy is twofold: (1) to be able to generate samples $G(z)$ hoping that $G(z) \in \mathcal{X}$ for $z \sim \mathbb{P}_{\mathcal{Z}}$ (or that \mathbb{P}_G is close to $\mathbb{P}_{\mathcal{X}}$) and (2) to use it for density estimation

$$p_{\mathcal{X}}(x) \approx p_G(x) = \int p(x|z)p_{\mathcal{Z}}(z)dz, \quad (1)$$

where p_G stands for the parametric density of $\mathbb{P}_G = G_{\#}\mathbb{P}_{\mathcal{Z}}$, the pushforward measure of $\mathbb{P}_{\mathcal{Z}}$ through G (defined in brief as $G_{\#}\mathbb{P}_{\mathcal{Z}}(B) = \mathbb{P}_{\mathcal{Z}}(\{z \in \mathcal{Z} | G(z) \in B\})$, for any B in the Borel σ -algebra associated to \mathcal{X}). Let us notice that the likelihood $p(x|z)$ depends on G and can be interpreted as a measure of how close $G(z)$ is to x .

Numerous strategies have been developed to parametrize G (or G_{θ}) as a neural network model and to learn the appropriate parameters θ by making \mathbb{P}_G as close as possible to $\mathbb{P}_{\mathcal{X}}$ for some probability distance $d(\mathbb{P}_G, \mathbb{P}_{\mathcal{X}})$. Among these strategies, we quote variational autoencoders, normalizing flows, generative adversarial networks, or autoregressive models.

The problem of image inpainting can also be naturally formulated in a probabilistic manner. Let y denote an observed incomplete image, which is unknown in O .

Table 1 Generative methods used in the analyzed state-of-the-art proposals for diverse inpainting

Method	VAE	Autoregressive	GAN	Transformers
PIC (Zheng et al. 2019)	✓		✓	
PiiGAN (Cai and Wei 2020)			✓	
UCTGAN (Zhao et al. 2020b)	✓		✓	
DSI-VQVAE (Peng et al. 2021)	✓	✓	✓	
ICT (Wan et al. 2021)			✓	✓
PD-GAN (Liu et al. 2021)			✓	
BAT (Yu et al. 2021)			✓	✓

We are interested in modeling the conditional distribution, $p(x|y)$, over the values of the variable x (corresponding to the complete image) conditioned on the value of the observed variable y . As possibly many plausible images are consistent with the same input image y , the distribution $p(x|y)$ will likely be multimodal. Then, each of the multiple solutions can be generated by sampling from that distribution using a given sampling strategy. Thus, the goal is not only to obtain a generative model that minimizes $d(\mathbb{P}_G, \mathbb{P}_{\mathcal{X}_s})$, where $\mathcal{X}_s \subset \mathcal{X}$ is the set of possible solutions, but also to design a mechanism able to sample the conditional distribution $p(x|y)$, i.e., for a given damaged incomplete image y , output a set of plausible completions x of y .

In this section, we will analyze the different families of generative models proposed in the literature to realize diverse image inpainting. We will in particular describe generative adversarial networks (GAN), variational autoencoders (VAE), autoregressive models, and transformers. We will also detail the different objective losses proposed to train these networks. Finally, for each family of models, we will review several state-of-the-art diverse inpainting methods that relate to this model. Table 1 lists all the methods that will be reviewed in this section.

Generative Adversarial Networks

Generative adversarial networks (GANs) are a type of generative models that have received a lot of attention since the seminal work of Goodfellow et al. (2014). The GAN strategy is based on a game theory scenario between two networks, a generator network and a discriminator network, that are jointly trained competing against each other in the sense of a Nash equilibrium. The generator maps a vector from the latent space, $z \sim \mathbb{P}_Z$, to the image space trying to trick the discriminator, while the discriminator receives either a generated or a real image and must distinguish between both. The parameters of the generator and the discriminator are learned jointly by optimizing a GAN objective by a min-max procedure. This procedure leads the probability distribution of the generated data to be as close as possible, for some distance, to the one of the real data. Several GAN variants have appeared. They mainly differ on the choice of the distance $d(\mathbb{P}_1, \mathbb{P}_2)$ between two probability

distributions \mathbb{P}_1 and \mathbb{P}_2 . The first GAN by Goodfellow et al. (2014) (also referred to as vanilla GAN) makes use of the Jensen–Shannon divergence, which is defined from the Kullback–Leibler divergence (KL), by

$$d_{JS}(\mathbb{P}_1, \mathbb{P}_2) = \frac{1}{2} \left[\text{KL} \left(\mathbb{P}_1 \parallel \frac{\mathbb{P}_1 + \mathbb{P}_2}{2} \right) + \text{KL} \left(\mathbb{P}_2 \parallel \frac{\mathbb{P}_1 + \mathbb{P}_2}{2} \right) \right], \tag{2}$$

where the KL is defined, for discrete probability densities, as

$$\text{KL}(\mathbb{P}_1, \mathbb{P}_2) = \sum_x \mathbb{P}_1(x) \log \left(\frac{\mathbb{P}_1(x)}{\mathbb{P}_2(x)} \right). \tag{3}$$

and, for continuous densities, as

$$\text{KL}(\mathbb{P}_1, \mathbb{P}_2) = \int_{\mathcal{X}} \mathbb{P}_1(x) \log \frac{\mathbb{P}_1(x)}{\mathbb{P}_2(x)} dx. \tag{4}$$

The Wasserstein GAN (Arjovsky et al. 2017) uses the Wasserstein-1 distance, given by

$$\mathbb{W}_1(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\pi \in \Pi(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_{x, y \sim \pi} (\|x - y\|), \tag{5}$$

where $\Pi(\mathbb{P}_1, \mathbb{P}_2)$ is the set of all joint distributions π whose marginals are, respectively, \mathbb{P}_1 and \mathbb{P}_2 . By Kantorovich–Rubenstein duality, the Wasserstein-1 distance can be computed as

$$\mathbb{W}_1(\mathbb{P}_1, \mathbb{P}_2) = \sup_{D \in \mathcal{D}} (\mathbb{E}_{x \sim \mathbb{P}_1} [D(x)] - \mathbb{E}_{y \sim \mathbb{P}_2} [D(y)]), \tag{6}$$

where \mathcal{D} denotes the set of 1-Lipschitz functions. In practice, the dual variable D is parametrized by a neural network and it represents the so-called discriminator.

Both the generator and discriminator are jointly trained to solve

$$\min_G \sup_{D \in \mathcal{D}} (\mathbb{E}_{x \sim \mathbb{P}_X} [D(x)] - \mathbb{E}_{y \sim \mathbb{P}_G} [D(y)]), \tag{7}$$

in the case of the Wasserstein GAN, and

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_X} [\log D(x)] + \mathbb{E}_{y \sim \mathbb{P}_G} [\log(1 - D(y))] \tag{8}$$

for the vanilla GAN. In (8), the discriminator D is simply a classifier that tries to distinguish samples in the training set \mathcal{X} (real samples) from the generated samples $G(z)$ (fake samples) by designing a probability $D(x) \in [0, 1]$ for its likelihood to be from the same distribution as the samples in \mathcal{X} .

GANs are sometimes referred to as implicit probabilistic models due to the fact that they are defined through a sampling procedure where the generator learns to generate new image samples. This is in contrast to variational autoencoders, autoregressive models, and methods that explicitly maximize the likelihood.

For the task of inpainting, several proposals set the problem as a conditioned one. The GAN approach is modified such that the input of the generator G is both an incomplete image y and a latent vector $z \sim \mathbb{P}_Z$, and G performs conditional image synthesis where the conditioning input is y . In the GAN-based works that we present in this section (Cai and Wei 2020; Liu et al. 2021), the authors focus on multimodal conditioned generation where the goal is to generate multiple plausible output images for the same given incomplete image.

Finally, let us mention that in these works, and in general in some works described in this chapter, the used generative methods are combined with consistency losses that encourage the inpainted images to be close to the ground truth. Examples of those consistency losses include value and feature reconstruction losses and perceptual losses. Nonetheless, multiple inpainting researchers acknowledge that it can be counterproductive to rely on consistency losses due to the fact that the ground truth is only one of the multiple solutions.

PiiGAN: Generative Adversarial Networks for Pluralistic Image Inpainting (Cai and Wei 2020)

One of the first methods that use GANs in order to generate pluralistic results is PiiGAN (Cai and Wei 2020). PiiGAN is a deep generative model that incorporates a style extractor that can extract the style features, in the form of a latent vector, from the ground-truth image.

To be more precise, the network is composed of one *generator* and *extractor* network. The training follows two different paths.

First, given a ground-truth image x_{gt} , the extractor is used to estimate the style feature z_{gt} . Once the style feature z_{gt} is obtained, the ground-truth image x_{gt} is masked and concatenated with the computed style feature z_{gt} and used as input of the *generator* network. The generator network will estimate the inpainted version x of the masked ground-truth image y . The estimated inpainted image $x_{out,1}$ is passed through the extractor network to estimate the corresponding style feature $z_{out,1}$. This path is supervised using the KL-divergence between the style features z_{gt} and $z_{out,1}$.

In parallel, another path estimates inpainted images from masked images without ground truth. That is, masked images without ground truth y_{raw} are fed to the generator with a random vector z_{raw} . An inpainted image $x_{out,2}$ is predicted followed by style feature prediction $z_{out,2}$. Additionally, they frame the inpainting of y_{raw} in an adversarial approach equipped with a local (that focuses just in the inpainted area) and a global discriminator applied to the inpainted image x_{raw} . This path is supervised using the L^1 norm of the difference between the style features x_{raw} and $x_{out,2}$ and adversarial loss applied to the inpainted image $x_{out,2}$ based on the Wasserstein loss (7) with gradient penalty as defined in Gulrajani et al. (2017).

The authors claim that their results are diverse and natural, especially for images with large missing areas. Figure 1 shows an overview of the algorithm pipeline.

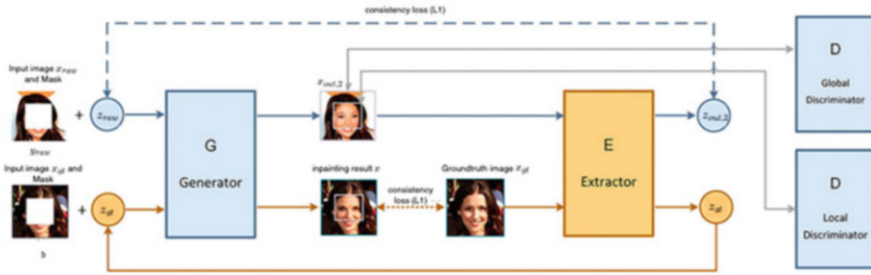


Fig. 1 Overview of the architecture of PiiGAN: Generative Adversarial Networks for Pluralistic Image Inpainting (Cai and Wei 2020). (Figure from Cai and Wei 2020)

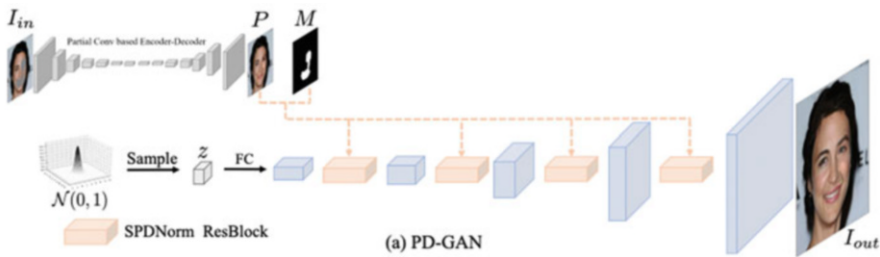


Fig. 2 Overview of the architecture of PD-GAN: Probabilistic Diverse GAN for Image Inpainting (Liu et al. 2021). (Figure from Liu et al. 2021)

PD-GAN: Probabilistic Diverse GAN for Image Inpainting (Liu et al. 2021)

The authors of Liu et al. (2021) propose a method to perform diverse image inpainting called PD-GAN. PD-GAN takes advantage of the benefits of GANs in generating diverse content from different random noise inputs. Figure 2 displays an overview of the algorithm pipeline. In contrast to the original vanilla GAN, in PD-GAN all the decoder deep features are modulated from coarse to fine by injecting prior information at each scale. This prior information is extracted from an initially restored image at a coarser resolution together with the inpainting mask. For that purpose, they introduce a probabilistic diversity normalization (SPDNorm) module based on the Spatially-adaptive denormalization (SPADE) module proposed in Park et al. (2019). SPDNorm works by modeling the probability of generating a pixel conditioned on the context information. It allows more diversity toward the center of the inpainted hole and more deterministic content around the inpainting boundary.

The objective loss is a combination of several losses, including a diversity loss, a reconstruction loss, an adversarial loss, and a feature matching loss (difference in the output feature layers computed with the learned discriminator). In general, in the context of multiple-image synthesis, *diversity losses* aim at ensuring that the different reconstructed images are diverse enough. In particular, the authors of PD-GAN (Liu et al. 2021) use the so-called *perceptual diversity loss*, defined as

$$\mathcal{L}_{pdiv}(x_{out_a}, x_{out_b}) = \frac{1}{\sum_l \|M \odot (\Phi_l(x_{out_a}) - \Phi_l(x_{out_b}))\|_1 + \epsilon}. \quad (9)$$

where x_{out_a} and x_{out_b} are two inpainted results, and M the inpainting mask (with 1 values on the missing pixels and 0 elsewhere). The minimization of (9) favors the maximization of the perceptual distance of inpainted regions in x_{out_a} and x_{out_b} . Notice that the non-masked pixels are not affected by this loss. A similar diversity loss was proposed in Mao et al. (2019).

Variational Autoencoders and Conditional Variational Autoencoders

Variational autoencoders (VAE) (Kingma and Welling 2013) are generative models for which the considered distance between probability distributions is the Kullback–Leibler divergence. Maximization of the log-likelihood criterion is equivalent to the minimization of a Kullback–Leibler divergence between the data and model distributions. In the VAE context, the generator G_θ is referred to as the *decoder*.

Let us first derive the vanilla VAE formulation in the general context of non-corrupted images $x \in \mathcal{X}$. Using Bayes rule, the likelihood $p_{G_\theta}(x)$, for $x \sim \mathbb{P}_\mathcal{X}$ and $z \sim \mathbb{P}_\mathcal{Z}$, is given by

$$p_\theta(x) = \frac{p_\theta(x, z)}{p_\theta(z|x)} = \frac{p_\theta(x|z)p_\mathcal{Z}(z)}{p_\theta(z|x)} \quad (10)$$

where, to simplify notations, we have denoted p_{G_θ} simply by p_θ . In order to bypass the intractability of the posterior $p_\theta(z|x)$, variational autoencoders introduce a second neural network, $q_\psi(z|x)$, to parametrize an approximation of the true posterior. This neural network is referred to as the *encoder*. Let us now derive the VAE objective function. Following Kingma et al. (2019),

$$\log p_\theta(x) = \mathbb{E}_{q_\psi(z|x)} \left[\log \left[\frac{p_\theta(x, z)}{q_\psi(z|x)} \right] \right] + \mathbb{E}_{q_\psi(z|x)} \left[\log \left[\frac{q_\psi(z|x)}{p_\theta(z|x)} \right] \right] \quad (11)$$

$$= \mathbb{E}_{q_\psi(z|x)} [\log p_\theta(x, z) - \log q_\psi(z|x)] + \text{KL}(q_\psi(z|x) || p_\theta(z|x)) \quad (12)$$

$$= \mathcal{L}_{\theta, \psi}(x) + \text{KL}(q_\psi(z|x) || p_\theta(z|x)). \quad (13)$$

$\mathcal{L}_{\theta, \psi}$ is the so-called evidence lower bound (ELBO). By positivity of the KL, it verifies

$$\mathcal{L}_{\theta, \psi}(x) = \log p_\theta(x) - \text{KL}(q_\psi(z|x) || p_\theta(z|x)) \leq \log p_\theta(x), \quad (14)$$

and $\mathcal{L}_{\theta, \psi}(x) = \log p_\theta(x)$ if and only if $q_\psi(z|x)$ is equal to $p_\theta(z|x)$.

VAE training consists in maximizing $\mathcal{L}_{\theta, \psi}$ in (14) with respect to the parameters $\{\theta, \psi\}$ of the encoder and of the decoder, simultaneously. The goal is to obtain a good approximation $q_{\psi}(z|x)$ of the true posterior $p_{\theta}(z|x)$ while maximizing the marginal likelihood $p_{\theta}(x)$.

The work Sohn et al. (2015) extends VAEs by proposing conditional variational autoencoders (CVAE). Their targeted distribution is the conditional distribution of x given an input “conditional” variable c and the maximization of the log-likelihood criterion becomes

$$\max_{\theta} \mathbb{E}_{x \sim \mathbb{P}_{\mathcal{X}}} \log p_{G_{\theta}}(x|c). \quad (15)$$

CVAE loss is obtained with a similar argument as in (11), (12), (13), and (14) by maximizing the conditional log-likelihood, which gives the variational lower bound of the conditional log-likelihood

$$\mathbb{E}_{q_{\psi}(z|x)} [\log p_{\theta}(x|c, z)] - \text{KL}(q_{\psi}(z|x, c) || p_{\theta}(z|x)) \leq \log p_{\theta}(x|c). \quad (16)$$

Then, the idea of the deep conditional generative modeling is simple: given an observation (input) x , z is drawn from a prior distribution $p_{\theta}(z|x)$. Then, the output is generated from the distribution $p_{\theta}(x|z, c)$. Bao et al. (2017) combines a CVAE with a GAN (CVAE-GAN) for fine-grained category image generation. Even if inpainting results are shown, the network is not trained explicitly for inpainting but for image generation conditioned on image labels.

In the context of multiple-image inpainting, or more generally of multiple-image restoration, a straightforward idea is to condition the generative model on the input degraded image y and to generate multiple images x sampling from $p_{\theta}(x|z, c = y)$. BicycleGAN (Zhu et al. 2017) uses this idea for diverse image-to-image translation. Their goal is to learn a bijective mapping between two image domains with a multimodal conditional distribution. They combine CVAE-GAN with latent regressors and show that their method can produce both diverse and realistic results across various image-to-image translation problems. However, their method is not explicitly applied for image inpainting. Moreover, as observed by several authors (see, e.g., Zheng et al. 2019; Wan et al. 2021), using standard conditional VAEs or CVAE-GAN for the specific task for image inpainting still leads to minimal diversity and quality. Several extensions of these models have recently appeared for diverse image inpainting. They are presented below with more details.

Finally, let us notice that VAE model has been extended in van den Oord et al. (2017) and Razavi et al. (2019) to the so-called vector quantized–variational autoencoder (VQ-VAE) that uses vector quantization to model discrete latent variables. Such discretization is done to avoid posterior collapse. The quantization codebook is trained at the same time as the autoencoder with an objective loss made of a reconstruction term and a regularization term that ensures that the embedding fits the encoder and outputs, respectively. The work Razavi et al. (2019) is a hierarchical extension of van den Oord et al. (2017). In particular, the authors of Razavi et al.

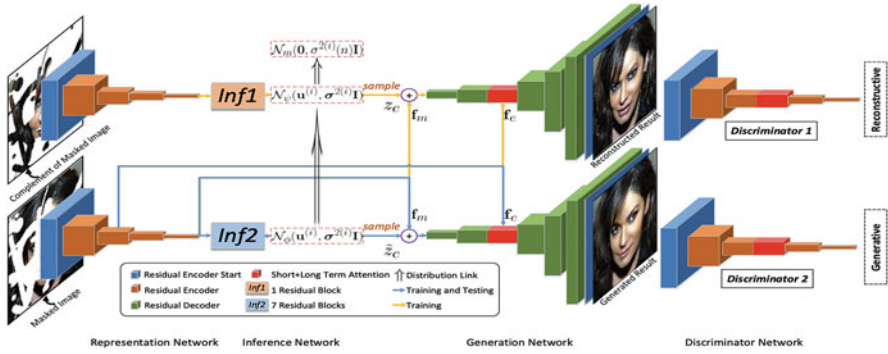


Fig. 3 Overview of the PIC architecture of pluralistic image completion (Zheng et al. 2019). (Figure from Zheng et al. 2019)

(2019) show that, by only considering two levels of a multiscale hierarchical organization of VQ-VAE (van den Oord et al. 2017), the information about image texture is disentangled from the information about the structure and geometry of the objects in an image. By combining the obtained hierarchical multiscale latent data with an autoregressive model as prior (see section “Autoregressive Models” below), they show an improved ability for generating high-resolution images.

Pluralistic Image Completion (Zheng et al. 2019)

The work Zheng et al. (2019) aims to estimate a probability distribution $p(x|y)$ from which to sample, where y represents an incomplete image and x one of its possible completions. They propose to use the conditional variational autoencoder (CVAE) (Sohn et al. 2015) approach described above which estimates a parametric distribution over a latent space, as in equation (16), from which sampling is possible. However, in Zheng et al. (2019), the authors observe that if they explicitly promote the inpainted output to be similar to the ground-truth image (either by any error-based loss such as, for instance, the L^1 distance, or as the authors show, by maximizing $\mathbb{E}_{q_\psi(z|x)} [\log p_\theta(x|y, z)]$ in (16) while $KL(q_\psi(z|x, y)||p_\theta(z|x))$ tends to zero), it results in a lack of diverse outputs. Alternatively, one could impose to fit the distribution of the training dataset by an adversarial approach including a discriminator as described in section “Generative Adversarial Networks”. However, this approach is highly unstable. Instead, they propose a probabilistic framework with a dual pipeline composed of two paths. See a detailed pipeline in Fig. 3. One is the *reconstructive path* which is a VAE-based model that utilizes the ground truth to get a prior distribution of missing parts, $x|O$, with the variance on the latent variables’ prior depending on the hole area, and rebuild the exact same ground-truth image from this distribution. The other is a *generative path* for which the conditional prior, based only on the visible regions, is coupled to the distribution obtained in the reconstructive path to generate multiple and diverse samples. Both parts are framed in an adversarial approach to fit the distribution of the training dataset. Accordingly,

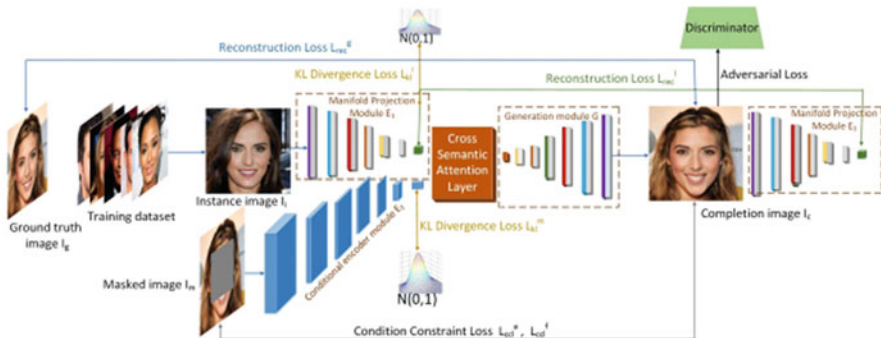


Fig. 4 Overview of the architecture of UCTGAN: Diverse inpainting based on unsupervised cross-space translation (Zhao et al. 2020b). (Figure from Zhao et al. 2020b)

the whole training loss is a combination of three types of terms. First, they use the KL divergences between the mentioned distributions. Second, the appearance terms based on the L^1 norm of the error, where in the generative path it only has into account the visible pixels. And lastly, the third term is an adversarial discriminator-based term. It is based on the L^1 difference among the discriminator features of the ground-truth and the reconstructed image for the reconstructive path and on the discriminator value on the generated image for the generative path. Additionally, to exploit the distant relation among the encoder and decoder, they use a modified self-attention layer that captures fine-grained features in the encoder and more semantic generative features in the decoder.

UCTGAN: Diverse Inpainting Based on Unsupervised Cross-Space Translation (Zhao et al. 2020b)

The authors of Zhao et al. (2020b) aim to produce multiple and diverse solutions conditioned by an instance image that guides the reconstruction, again aiming to maximize the conditional log-likelihood involving the variational lower bound (16) on the training dataset. They call their proposal UCTGAN. The pipeline is presented in Fig. 4. They use a two-encoder network that transforms the instance image and the corrupted image to a low-dimensional manifold space. A cross semantic attention layer combines the information in both low-dimensional spaces. Consecutively, a generator is used to compute the conditional reconstructed image.

The objective loss is composed of four terms. First, a constraint loss in the uncorrupted pixels is applied by minimizing the L^1 norm of the difference both at pixel and feature levels. Second, the KL divergence is used to project the low-dimensional manifold space of the instance image and masked image into a multivariate normal distribution space. Additionally, the L^1 norm of the difference in the low-dimensional manifold space of the instance image and the ground-truth image is added. Finally, all the training is framed in an adversarial approach using the vanilla GAN (8), where the discriminator works in the image space.

Generating Diverse Structure for Image Inpainting with Hierarchical VQ-VAE (Peng et al. 2021)

The multiple inpainting proposal in Peng et al. (2021) leverages three generative strategies, namely, variational autoencoders, generative adversarial methods, and autoregressive models. We first review below the main ideas of autoregressive models and then describe the proposal (Peng et al. 2021).

Autoregressive Models

In autoregressive models (Van Oord et al. 2016; Oord et al. 2016; Chen et al. 2018), the likelihood $p_\theta(x)$ is learned by choosing an order of the data variables $x = (x_1, x_2, \dots, x_n) \in \mathcal{X}$, frequently related to values on the n pixels of an image, and exploiting the fact that the joint distribution can be decomposed as

$$p(x) = p(x_1, x_2, \dots, x_n) = p(x_1) \prod_{i=2}^n p(x_i | x_1, \dots, x_{i-1}). \quad (17)$$

More generally, a similar decomposition to (17) can be obtained by splitting the set of variables in smaller disjoint subsets. In this case, and considering the variable order of x_1, x_2, \dots, x_n to be represented by a directed and noncyclic graph, one has

$$p(x) = p(x_1, x_2, \dots, x_n) = p(x_1) \prod_{i=2}^m p(x_i | S(x_i)), \quad (18)$$

where $S(x_i)$ is the set of parent variables of variable i and $m \leq n$.

Autoregressive models have been used to learn a probability distribution or a conditional distribution, for instance, in the context of VAEs (cf. section “[Variational Autoencoders and Conditional Variational Autoencoders](#)”) to model the prior or the decoder and also to tackle several problems in imaging such as image generation (e.g., Razavi et al. 2019), super resolution (e.g., Dahl et al. 2017), inpainting (e.g., Peng et al. 2021) or image colorization (e.g., Zhao et al. 2020a; Guadarrama et al. 2017; Royer et al. 2017), and also for other types of data such as audio and speech synthesis (e.g., Oord et al. 2018) or text (e.g., Bowman et al. 2015).

Generating Diverse Structure for Image Inpainting with Hierarchical VQ-VAE (Peng et al. 2021)

Inspired by the hierarchical vector quantized variational autoencoder (VQ-VAE) (Razavi et al. 2019) whose hierarchical architecture disentangles structural and textural information, the authors of Peng et al. (2021) propose a two-stage pipeline (cf. Fig. 5). As already pointed out by Razavi et al. (2019), by using a two-step approach instead of directly computing the final inpainted image, they aim to generate richer structure and texture images than previous VAE-based methods that often produce a distorted structure or blurry textures.

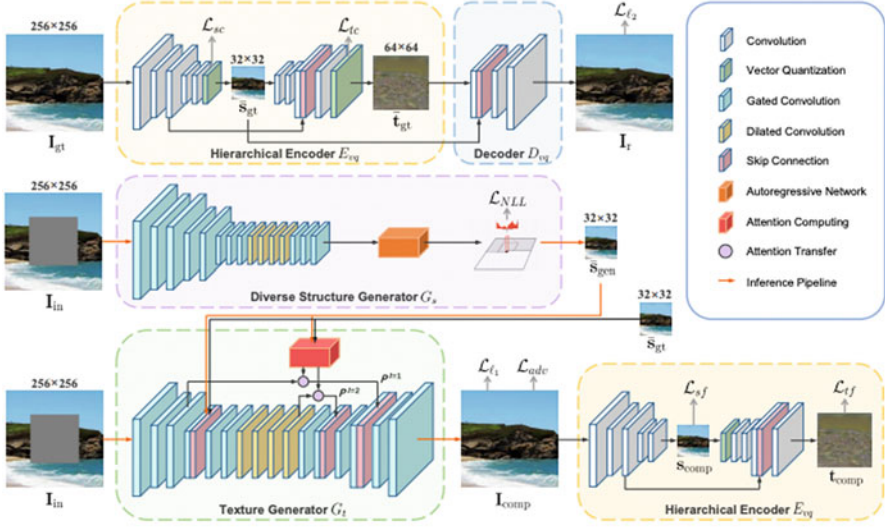


Fig. 5 Overview of the architecture of generating diverse structure for image inpainting with hierarchical VQ-VAE (DSI-VQVAE) (Peng et al. 2021). (Figure from Peng et al. 2021)

The first stage of Peng et al. (2021), known as *diverse structure generator*, generates multiple low-resolution results, each of which has a different structure by sampling from a conditional autoregressive distribution. The second stage, known as *texture generator*, uses an encoder–decoder architecture with a structural attention module that refines each low-resolution result separately by augmenting texture. The structural information module facilitates the capture of distant correlations. They further reuse the VQ-VAE to calculate two feature losses, which help improve structure coherence and texture realism, respectively.

The authors first train the hierarchical VQ-VAE and, afterward, the diverse structure generator (G_s depending on parameters θ) and the texture generator (G_t depending on parameters φ) are trained separately. These generators are later on used for inference. The structure generator G_s is constructed via a conditional autoregressive network for the distribution over structural features. In inference, it will generate different structural features via sampling. Its objective loss is defined as the negative log-likelihood

$$\mathcal{L}_l(\theta) = -\mathbb{E}_{x_{gt} \sim \mathbb{P}_X}[\log(p_\theta(s_{gt}|y, M))] \tag{19}$$

where y is the input image to be inpainted on the points of O where the hole mask M is equal to 1, \mathbb{P}_X denotes the distribution of the training dataset, s_{gt} denote the vector quantized structural features of the ground truth at the coarser scale given by the hierarchical VQ-VAE, and θ the parameters of G_s .

Besides, the objective loss for the texture generator G_t is composed by: (i) the L^1 norm comparing the inpainted solution to the ground truth at pixel level, (ii) an adversarial loss using the discriminator trained with the SN-PatchGAN hinge version (Yu et al. 2019) applied to the resulting image and, moreover, (iii) a structural feature loss $\mathcal{L}_{sf}(\varphi)$, and (iv) a textural feature loss $\mathcal{L}_{tt}(\varphi)$. These last two losses are defined similarly using a multiclass cross-entropy loss. In particular, the structural feature loss is defined as

$$\mathcal{L}_{sf}(\varphi) = - \sum_{k,j} \alpha_{k,j} \log(\text{softmax}(\lambda_2 \delta_{k,j})), \quad (20)$$

where $\delta_{k,j}$ denotes the truncated distance similarity score between the k -th feature vector of s_{comp} (computed from the inpainted image using the trained encoder) and the j -th prototype vector of the structural codebook of VQ-VAE, λ_2 is a parameter set to 10, and $\alpha_{k,j}$ is an indicator of the prototype vector class. That is, $\alpha_{k,j} = 1$ when the k -th feature vector of s_{gt} belongs to the j -th class of the structural codebook; otherwise, $\alpha_{k,j} = 0$. The authors define the textural feature loss $\mathcal{L}_{tt}(\varphi)$ in an analogous way.

As mentioned, in section “[Experimental Results](#)”, we will experimentally analyze this method. It will be denoted there as DSI-VQVAE.

Image Transformers

Self-attention-based architectures, in particular transformers (Vaswani et al. 2017) are well-explored architectures in natural language processing (NLP). Transformers use a self-attention mechanism to model long-range relationships between the elements of an input sequence (for instance, in a text) that has shown to be more efficient than recurrent neural networks. They have achieved state-of-the-art results in several tasks not only in the field of NLP but also more recently for computer vision problems. The vanilla transformer (Vaswani et al. 2017) and its variants have been successfully applied in computer vision to, e.g., inpainting (Wan et al. 2021; Yu et al. 2021), object detection (Carion et al. 2020), image classification (Dosovitskiy et al. 2020), colorization (Kumar et al. 2021), and super resolution (Yang et al. 2020).

Instead of using inductive local biases like CNNs, transformers in imaging aim to have a global receptive field. For this, the image is first transformed by, as in the most basic approach, flattening the spatial dimensions of the input feature map into a sequence of features of size $M \times N \times F$, where $M \times N$ represents the flattened spatial dimensions and F the depth of a feature map. Then self-attention is applied over the extracted sequence. To ease the associated high computational cost, some authors substitute spatial pixels by patches. The attention mechanism looks at the input sequence and decides for each position which other parts of the sequence or image are important. More specifically, the transformers will transform the set of inputs, called *tokens*, using sequential blocks of multiheaded self-attention, which

relate embedded inputs to each other. It is worth noticing that transformers will maintain the number of tokens throughout all computations. If tokens were related to pixels, each pixel would have a one-to-one correspondence with the output, thus, maintaining the spatial resolution of the original input image. Since transformers are set-to-set functions, they do not intrinsically retain the information of the spatial position for each individual token; thus, the embedding is concatenated to a learnable position embedding to add the positional information to the representation.

One advantage of using a transformer for image restoration is that it naturally supports pluralistic outputs by directly optimizing the underlying data distribution. One drawback is the computational complexity that increases quadratically with the input length, thus making it difficult to directly synthesize high-resolution images.

High-Fidelity Pluralistic Image Completion with Transformers (Wan et al. 2021)

The authors of Wan et al. (2021) exploit the benefits of both transformers and CNNs. The use of transformers will enforce a global structural understanding and pluralism support in the inpainted region, at a coarse resolution. On the other hand, the use of CNNs will allow working with high-resolution images without a high computational cost due to its capacity of estimating local textures efficiently.

Concretely, in this work, image completion is performed in two steps as shown in Fig. 6. In the first step, given a corrupted image, the authors use transformers to produce the probability distribution of structural appearance of complete images given the incomplete one. Low-resolution results can be obtained by sampling from this distribution with diversities that recover pluralistic coherent image structures. In the second step, guided by the computed image structures together with the available pixels of the input image, another upsampling CNN model is used to render high-fidelity textures for missing regions meanwhile ensuring coherence with neighboring pixels.

If X_{Π} denotes the set of masked tokens x_{π_k} (where $\Pi = \{\pi_1, \dots, \pi_K\}$ denote their indexes), and $X_{-\Pi}$ denotes the set of unmasked tokens (corresponding to the

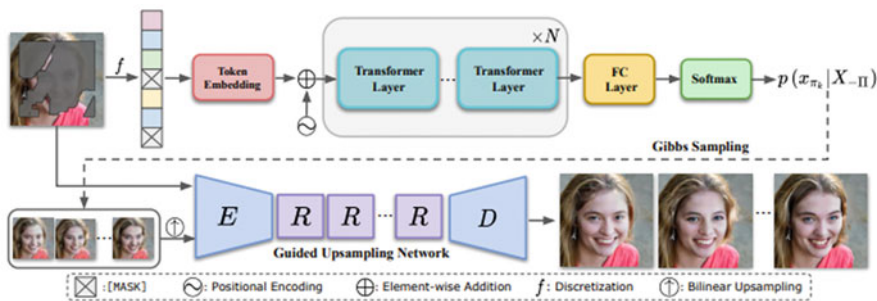


Fig. 6 Overview of the architecture of high-fidelity pluralistic image completion with transformers (Wan et al. 2021), referred to as ICT. (Figure from Wan et al. 2021)

visible regions), then the transformer is optimized by minimizing the negative log-likelihood of the masked tokens x_{π_k} , conditioned the visible regions $X_{-\Pi}$, that is,

$$\mathcal{L}_{\text{MLM}}(\theta) = \mathbb{E}_X \frac{1}{K} \sum_{k=1}^K -\log p(x_{\pi_k} | X_{-\Pi}; \theta) \quad (21)$$

where θ contains the parameters of the transformer and the subindex MLM stands for the *masked language model* which is similar to the one in BERT (Devlin et al. 2018). One particularity of the ICT model is that each token attends simultaneously to all positions thanks to bidirectional attention. This enables the generated distribution to capture the full context, thus leading to a consistency between generated contents and unmasked region.

Once the transformer is trained, instead of directly sampling the entire set of masked positions which would lead to non-plausible results due to the independence property, they apply Gibbs sampling to iteratively sample tokens at different locations. To do so, in each iteration, a grid position is sampled from $p(x_{\pi_k} | X_{-\Pi}, X_{<\pi_k}, \theta)$ with the top- \mathcal{K} predicted elements, where $X_{<\pi_k}$ denotes the previous generated tokens.

The second step is to perform texture refinement at the original resolution using a CNN, which is optimized by minimizing the L^1 loss between the predicted image and the ground truth, together with an adversarial loss based on the vanilla GAN (cf. (8) in section “[Generative Adversarial Networks](#)”).

Diverse Image Inpainting with Bidirectional and Autoregressive Transformers (Yu et al. 2021)

This proposal exploits, as in Wan et al. (2021), a two-step strategy where transformers will encode global structure understanding and high-level semantics at the first stage, followed by a CNN-based generation of additional texture. While Wan et al. (2021) leverages bidirectional attention with the masked language model (MLM) as in BERT (Devlin et al. 2018), the authors of Yu et al. (2021) propose BAT-Fill that combines autoregressive models and bidirectional models (cf. Fig. 7). The first transformer-based step estimates the distribution of inpainted low-resolution structures from which to sample, from an input damaged image, a set $\{s_1, \dots, s_J\}$ of plausible complete structures. Instead of only using a masked language model like BERT and Wan et al. (2021) (see above) that use bidirectional contextual information but predicts each masked token separately and independently (which can result in inconsistency in the generated result), BAT-Fill incorporates autoregressive modeling (factorizing the predicted tokens with the product rule). The input sequence of tokens is sorted by first having the visible tokens (permuted) and then the missing tokens (with the original order). In this way, the autoregressive model starts at the position of the first missing pixel. The BAT training objective is given by

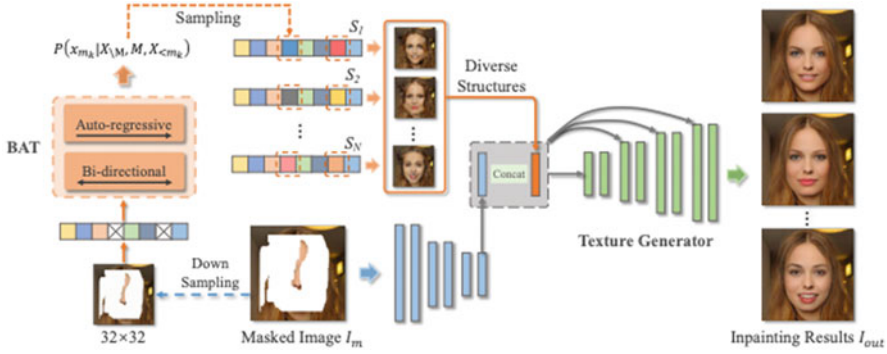


Fig. 7 Overview of the BAT architecture of diverse image inpainting with bidirectional and autoregressive transformers (Yu et al. 2021). (Figure from Yu et al. 2021)

$$\mathcal{L}_{\text{BAT}}(\theta) = \mathbb{E}_X \frac{1}{K} \sum_{k=1}^K -\log p(x_{\pi_k} | X_{-\pi}, M, X_{<\pi_k}; \theta). \tag{22}$$

where we have used the same notations as in (21), namely, K is the length of masked tokens, and $X_{-\pi}$ are all the unmasked tokens (corresponding to the visible regions). Finally, $X_{<\pi_k}$ denote the previous predicted tokens, and M the masked positions.

Finally, they construct a texture generator based on CNN-based synthesis, which is optimized by minimizing the L^1 loss between the predicted image and the ground truth together with an adversarial loss and a perceptual loss (Johnson et al. 2016).

In inference, each masked token is predicted bidirectionally and autoregressively. As in Wan et al. (2021), they iteratively use top- \mathcal{K} sampling to randomly sample from the \mathcal{K} most likely next tokens.

From Single-Image Evaluation Metrics to Diversity Evaluation

Currently, there is no consensus on automatic evaluation methods for single or diverse inpainting. As the problem is to recover a visually plausible image, performing quantitative evaluation is not trivial as the solution is not unique and the plausibility is a subjective term. Nevertheless, several evaluation metrics have been proposed through the years. We first detail those used for evaluating inpainting methods, one image at a time, before presenting the metric used as diversity scores.

Full-reference metrics compare the ground-truth image with the inpainted result. Famous measures in this category include L^1 , L^2 distances, PSNR, or SSIM (Wang et al. 2004). These metrics analyze the ability of the model to reconstruct the original image content. Nevertheless, it is easy to demonstrate that they do not well characterize the realism of an image. Being close to the ground-truth image

does not ensure being realistic. Other perceptual metrics have been proposed and are supposed to be more consistent with human judgment. In particular, Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018b) has been demonstrated to correlate well with the human perceptual similarity. It relies on the observation that hidden activations in CNNs trained for image classification are indeed a space where distance can strongly correlate with human judgment. Precisely, LPIPS computes a weighted L^2 norm between deep features of pair of images:

$$\text{LPIPS}(x, x_{gt}) = \sum_l \frac{1}{M_l N_l} \sum_{ij} \|w_l \odot (\Phi_r^l(i, j) - \Phi_{gt}^l(i, j))\|_2^2 \quad (23)$$

where x is the reconstructed image, x_{gt} is the ground truth, l is a layer number, (i, j) is a pixel, w_l are weights for each features, and Φ^l and $\Phi_{gt}^l \in \mathbb{R}^{M_l \times N_l \times C_l}$ are features unit-normalized in the channel dimension. LPIPS has been used in the context of inpainting when generating one image (e.g., Zheng et al. 2021). In Kettunen et al. (2019), it was shown that standard adversarial attack techniques can easily fool LPIPS. Therefore, a slightly different metric called E-LPIPS (Ensemble LPIPS) is proposed by applying random simple image transformations and dropout. Nonetheless, up to our knowledge, it has never been used in the context of inpainting.

When, apart from the set of images, there is available corresponding image categories, other metrics, that are also supposed to be following human judgment, can be used. The inception score (IS) (Salimans et al. 2016) was designed to measure how realistic the output from a GAN is. This score measures the variety of a set of generated images as well as the probability distribution of each image classification. This is done by comparing the class distribution of each image, which should have a low entropy, with the marginal distribution of the whole set, which should have high entropy:

$$\text{IS}(G) = \exp\left(\mathbb{E}_{x \sim p_g} \text{KL}(p(y|x) || p(y))\right) \quad (24)$$

where p_g is the model distribution of the whole set given by the generative model G ; x , an image sampled from p_g ; $p(y|x)$, the conditional class distribution; KL, the Kullback–Leibler divergence; and $p(y)$, the marginal class distribution. As detailed in Barratt and Sharma (2018), inception score has its own limitations: sensitivity to small changes in network weights, misleading results when used beyond the ImageNet dataset (Rosca et al. 2017), and adversarial examples when used for model optimization. The IS score was adapted to diverse inpainting in Zhao et al. (2020b), leading to the Modified Inception Score (MIS). When performing inpainting, there is only one kind of image, and so $p(y)$ can be removed. The MIS is then defined as

$$\text{MIS(G)} = \exp \left(\mathbb{E}_{x \sim p_g} \sum_i (p(y_i|x) \log p(y_i|x)) \right), \quad (25)$$

where y_i of is the class label of the i th generated sample. Another improvement of the IS is the Fréchet Inception Distance (FID) (Heusel et al. 2017) that compares the statistics of generated images to the ones of original images. FID uses the inception pre-trained model to extract the feature vectors of real images and fake images and compare their feature-wise means (μ_r, μ_f) and covariances (Σ_r, Σ_f):

$$\text{FID} = \|\mu_r - \mu_f\|^2 + \text{Tr}(\Sigma_r + \Sigma_f + 2(\Sigma_r \Sigma_f)^{1/2}). \quad (26)$$

Fréchet Inception Distance has been widely used for validating single and diverse inpainting results in recent papers (e.g., Peng et al. 2021; Liu et al. 2021; Yu et al. 2021).

Measuring Diversity

In the context of pluralistic inpainting, following the idea proposed for image-to-image translation in Zhu et al. (2017), LPIPS has been used as a *diversity score* to measure how perceptually different the generated images are (Cai and Wei 2020; Zhao et al. 2020b; Liu et al. 2021). The higher the LPIPS, the more diversity is present in the results. For instance, in Cai and Wei (2020), they compute the average distance between the 10,000 pairs randomly generated from the 1000 center-masked image samples. LPIPS is computed on the full-inpainting results and mask-region inpainting results, respectively.

Experimental Results

In this section, we present a quantitative and qualitative comparison of several existing methods for multiple-image inpainting. We include an assessment of both the quality and the diversity of the inpainted solutions. All the results shown in this section are thanks to publicly available code together with pre-trained weights provided by the authors. In Table 2, we summarize, for all the methods previously reviewed, the conditions in which the experiments were conducted. Note that, among these methods, only PIC (Zheng et al. 2019), PiiGAN (Cai and Wei 2020), DSI-VQVAE (Peng et al. 2021), ICT (Wan et al. 2021), and BAT (Yu et al. 2021) provide source code and pre-trained models. In the rest of this section, we describe the experimental settings in section “[Experimental Settings](#)” including datasets and used masks; quantitative results in section “[Quantitative Performance](#)” including proximity to ground truth, perceptual quality, and inpainting diversity; and finally, a qualitative analysis is provided in section “[Qualitative Performance](#)”.

Table 2 Generative methods for diverse inpainting: experimental conditions. Random regular and irregular masks are generated as in Zheng et al. (2019)

Method	Input size	Train datasets	Training masks	Code
PIC	256×256	Celeba-HQ ImageNet Paris Places2	Regular (center 128×128 + random) Irregular (random)	✓
PiiGAN	128×128	CelebA Mauflex Agricultural Disease	center 64×64	✓
UCTGAN	256×256	Celeba-HQ ImageNet Paris Places2	Regular (center 128×128 + random) Irregular (random)	✗
DSI-VQVAE	256×256	Celeba-HQ ImageNet Places2	Regular (center 128×128 + random) Irregular (random)	✓
ICT	256×256	FFHQ ImageNet Places2	Irregular Pconv (Liu et al. 2018)	✓
PD-GAN	256×256	Celeba-HQ Paris StreetView Places2	Irregular Pconv (Liu et al. 2018)	✗
BAT	256×256	CelebA-HQ Paris StreetView Places2	Irregular Pconv (Liu et al. 2018)	✓

Experimental Settings

Table 2 lists all the explained methods together with the training dataset and corresponding training masks. Aiming for a fair comparison, we compare and test the methods trained on the same training images, i.e., the VAE-based model PIC (Zheng et al. 2019), the VQVAE-based model DSI-VQVAE (Peng et al. 2021), and the two transformer-based models ICT (Wan et al. 2021) and BAT (Yu et al. 2021). Notice that we do not analyze the performance of PiiGAN (Cai and Wei 2020), as the training datasets and size images are different from the other methods.

Datasets

We evaluate the methods on the three datasets Celeba-HQ (Karras et al. 2018), Places2 (Zhou et al. 2017), and ImageNet (Russakovsky et al. 2015). All the evaluated models take as input images of resolution 256×256 . Due to the long inference time of DSI-VQVAE and ICT methods (see Table 7), quantitative experiments are made on 100 randomly selected images from each training dataset. For each kind of mask (see below) and for each image, we sample 25 different results.



Fig. 8 Example for each kind of mask considered for evaluation. In gray are the hidden pixels. From left to right: center, random regular, random irregular, and irregular Pconv masks from Liao et al. (2018) with $<20\%$, $[40\%, 60\%]$ and $[40\%, 60\%]$ hidden pixels

For Celeba-HQ, the 1024×1024 resolution images are resized to 256×256 . For Places2 and ImageNet, the compared methods were trained on 256×256 patches either by resizing the input images (PIC), by cropping them, randomly (DSI) or to the center patch (BAT), or by both cropping and resizing (ICT). We will both consider center-cropped and resized versions of the input images to ensure a fair comparison among the trained models.

Note that ICT is not trained on Celeba-HQ but on the FFHQ face dataset (Karras et al. 2019). FFHQ contains higher variation than Celeba-HQ in terms of age, ethnicity, and image background. It also has a good coverage of accessories. Images from both datasets are, however, similarly aligned and cropped. Therefore, we still give the results of the ICT method tested on the Celeba-HQ dataset, but the reader should remember this difference when analyzing the results.

Inpainting Masks

We use the following type of masks: center, random regular, random irregular, and irregular masks from Liu et al. (2018) with different proportions of hidden pixels. Figure 8 shows an example of each kind of mask. The random masks are generated once for each test image so that all the methods are evaluated on the same degradation.

We would like to highlight that the methods PIC and DSI-VQVAE train a different model for regular and irregular holes. Testing on centered or random regular masks is realized with the former model, and testing on irregular masks with the latter. The transformer-based methods ICT and BAT only train on “irregular Pconv” holes given by Liu et al. (2018). Testing on each type of mask will be done with this unique model.

Quantitative Performance

We first analyze the numerical performance of each method. Table 3 shows quantitative results on Celeba-HQ dataset. Additionally, results on Places2 and ImageNet are, respectively, shown in Tables 4 and 5.

In Tables 4 and 5, we give our results obtained by center-cropping the images on Places2 and ImageNet, respectively. For fair comparison, we also give, in Appendix

Table 3 Quantitative comparison of four pluralistic image inpainting methods (PIC, DSI-VQVAE, ICT and BAT) on **Celeba-HQ** and for different kind of masks (central, random regular, random irregular and from Liu et al. 2018). Best and second best results by column are in bold and italics, respectively

Mask	Method	Similarity to GT			Realism		Diversity
		PSNR \uparrow	SSIM \uparrow	L ¹ \downarrow	MIS \uparrow	FID \downarrow	LPIPS \uparrow
Irregular <20%	PIC	34.63	0.964	1.17	0.0206	16.8	0.0009
	DSI-VQVAE	<i>35.49</i>	<i>0.968</i>	1.41	0.0216	11.0	<i>0.0081</i>
	ICT	34.72	<i>0.968</i>	2.09	0.0200	9.84	0.0084
	BAT	36.25	0.974	<i>1.20</i>	<i>0.0208</i>	<i>9.90</i>	0.0056
Irregular 20%–40%	PIC	26.69	0.879	4.19	<i>0.0216</i>	34.2	0.0091
	DSI-VQVAE	27.36	0.888	<i>4.06</i>	0.0223	28.8	<i>0.0357</i>
	ICT	26.83	<i>0.891</i>	4.71	0.0189	26.7	0.0383
	BAT	27.28	0.900	3.85	0.0214	20.7	0.0269
Irregular 40%–60%	PIC	21.47	0.745	10.36	0.0153	65.4	0.0527
	DSI-VQVAE	22.53	0.770	<i>9.01</i>	<i>0.0156</i>	51.9	<i>0.0916</i>
	ICT	21.92	<i>0.773</i>	9.82	0.0153	50.7	0.0970
	BAT	22.35	0.787	8.91	0.0183	39.7	0.0731
Central 128 × 128	PIC	24.46	0.868	5.26	<i>0.0212</i>	23.8	0.0288
	DSI-VQVAE	25.25	<i>0.880</i>	5.08	0.0210	<i>21.7</i>	0.0243
	ICT	24.45	0.872	6.06	0.0170	27.3	0.0486
	BAT	<i>25.10</i>	0.882	<i>5.21</i>	0.0218	21.5	<i>0.0365</i>
Random regular	PIC	24.16	0.840	7.23	0.0188	33.4	0.0402
	DSI-VQVAE	24.98	0.850	6.46	<i>0.0200</i>	30.5	<i>0.0642</i>
	ICT	24.51	<i>0.852</i>	7.24	0.0180	31.3	0.0665
	BAT	<i>24.85</i>	0.860	6.52	0.0209	24.6	0.0541
Random irregular	PIC	23.47	0.759	8.45	0.0161	73.5	0.0280
	DSI-VQVAE	<i>24.27</i>	<i>0.785</i>	<i>7.56</i>	<i>0.0167</i>	58.8	<i>0.0744</i>
	ICT	23.26	0.781	9.26	0.0148	52.2	0.0855
	BAT	24.36	0.810	7.13	0.0186	40.8	0.0495
Average	PIC	25.65	0.843	6.11	0.0189	41.2	0.0266
	DSI-VQVAE	<i>26.65</i>	<i>0.857</i>	<i>5.60</i>	<i>0.0195</i>	33.8	<i>0.0497</i>
	ICT	25.95	0.855	6.53	0.0173	33.0	0.0575
	BAT	26.70	0.869	5.47	0.0203	26.2	0.0410

Table 4 Quantitative comparison of four pluralistic image inpainting methods (PIC, DSI-VQVAE, ICT and BAT) on 256×256 **center-cropped** images from **Places2**, for different kind of masks (central, random regular, random irregular and from Liu et al. 2018)

Mask	Method	Similarity to GT		Realism		Diversity	
		PSNR \uparrow	SSIM \uparrow	$L^1 \downarrow$	MIS \uparrow	FID \downarrow	LPIPS \uparrow
Irregular <20%	PIC	30.48	0.937	2.02	0.0507	36.8	0.0050
	DSI-VQVAE	31.58	0.952	2.11	0.0482	19.3	0.0187
	ICT	29.86	0.943	3.64	0.0463	22.8	0.0198
	BAT	32.20	0.957	1.83	0.0463	14.2	0.0158
Irregular 20%–40%	PIC	23.88	0.820	6.46	0.0378	97.6	0.0344
	DSI-VQVAE	24.20	0.844	6.14	0.0438	63.6	0.0707
	ICT	23.08	0.831	8.05	0.0428	70.0	0.0769
Irregular 40%–60%	PIC	19.92	0.667	13.75	0.0326	156.1	0.1309
	DSI-VQVAE	20.34	0.703	12.52	0.0398	110.2	0.1566
	ICT	19.49	0.686	14.66	0.0371	128.7	0.1668
	BAT	19.98	0.705	13.10	0.0364	107.0	0.1610
Central 128×128	PIC	20.98	0.812	9.00	0.0435	96.8	0.1080
	DSI-VQVAE	21.41	0.819	8.85	0.0416	79.8	0.1234
	ICT	20.93	0.812	10.22	0.0476	92.2	0.1204
	BAT	21.20	0.822	8.76	0.0442	81.8	0.1190
Random regular	PIC	21.70	0.783	10.14	0.0425	103.8	0.1124
	DSI-VQVAE	22.36	0.805	9.21	0.0412	75.8	0.1167
	ICT	21.75	0.796	10.77	0.0405	87.1	0.1237
	BAT	22.34	0.808	9.15	0.0436	76.6	0.1200
Random irregular	PIC	20.86	0.658	12.80	0.0255	165.4	0.0979
	DSI-VQVAE	21.18	0.701	11.78	0.0360	114.4	0.1450
	ICT	20.07	0.681	14.14	0.0334	131.9	0.1548
	BAT	20.85	0.708	12.00	0.0374	103.2	0.1454
Average	PIC	22.97	0.780	9.02	0.0388	109.2	0.0814
	DSI-VQVAE	23.51	0.804	8.44	0.0418	76.9	0.1052
	ICT	22.53	0.792	10.25	0.0413	88.9	0.1107
	BAT	23.44	0.809	8.97	0.0417	72.7	0.1047

(Tables 8 and 9), the results on these two datasets for resized images. The ICT method is run, as proposed in the original paper, with its top- \mathcal{K} parameter (cf. section “Image Transformers”) set to 50. We investigate the influence of the top- \mathcal{K} parameter in Table 6. Note that for fair quantitative comparison, unlike Zheng

Table 5 Quantitative comparison of three pluralistic image inpainting methods (PIC, DSI-VQVAE and ICT) on 256×256 **center-cropped** images from **ImageNet**, for different kind of masks (central, random regular, random irregular and from Liu et al. 2018)

Mask	Method	Similarity to GT			Realism		Diversity
		PSNR \uparrow	SSIM \uparrow	L ¹ \downarrow	MIS \uparrow	FID \downarrow	LPIPS \uparrow
Irregular <20%	PIC	<i>30.33</i>	<i>0.941</i>	2.02	0.2416	20.2	0.0036
	DSI-VQVAE	30.44	0.946	2.38	<i>0.2361</i>	<i>12.1</i>	0.0199
	ICT	29.23	0.940	3.98	0.2323	10.7	<i>0.0185</i>
Irregular 20%–40%	PIC	23.02	0.797	7.37	0.1709	83.7	0.0289
	DSI-VQVAE	22.98	0.809	7.56	0.2015	53.4	0.0855
	ICT	22.24	<i>0.802</i>	9.23	<i>0.1970</i>	24.9	<i>0.0771</i>
Irregular 40%–60%	PIC	18.33	0.623	<i>16.34</i>	0.0792	183.9	0.1269
	DSI-VQVAE	18.92	0.651	14.82	<i>0.1192</i>	<i>126.3</i>	0.1907
	ICT	<i>18.52</i>	<i>0.646</i>	16.41	0.1329	101.7	<i>0.1700</i>
Central 128 \times 128	PIC	19.87	0.794	9.77	0.1591	95.8	0.1067
	DSI-VQVAE	<i>20.06</i>	0.795	9.99	0.1754	85.6	0.1291
	ICT	20.34	<i>0.795</i>	10.76	<i>0.1753</i>	73.8	<i>0.1162</i>
Random regular	PIC	19.81	0.737	13.24	0.0934	129.2	0.1027
	DSI-VQVAE	20.54	0.756	11.52	<i>0.1305</i>	89.3	0.1540
	ICT	<i>20.32</i>	<i>0.752</i>	<i>12.76</i>	0.1420	77.6	<i>0.1360</i>
Random irregular	PIC	<i>19.51</i>	0.598	<i>14.78</i>	0.0645	193.0	0.0982
	DSI-VQVAE	19.81	0.636	14.04	<i>0.1147</i>	<i>136.8</i>	0.1757
	ICT	19.02	<i>0.628</i>	16.08	0.1363	108.5	<i>0.1574</i>
Average	PIC	<i>21.81</i>	0.748	<i>10.59</i>	0.1347	117.6	0.0735
	DSI-VQVAE	22.13	0.765	10.07	<i>0.1629</i>	83.9	0.1258
	ICT	21.61	<i>0.761</i>	11.54	0.1693	66.2	<i>0.1125</i>

et al. (2019), we do not use any discriminator score to select the best generated samples.

To measure inpainting quality, we take into account three factors: the similarity to the ground truth, the realism of inpainting outputs, and the diversity of those outputs. Definitions and details on the metrics for each factor can be found in section “[From Single-Image Evaluation Metrics to Diversity Evaluation](#)”. Note that, contrary to Zheng et al. (2019), we do not use here any discriminator score to select the best samples before evaluation.

In each table, the best and second-best results by column are in bold and italics, respectively.

Table 6 Influence of the top- \mathcal{K} parameter on the ICT results. Results obtained on Places2 dataset, with central mask

top- \mathcal{K}	Similarity to GT			Realism		Diversity
	PSNR \uparrow	SSIM \uparrow	$L^1 \downarrow$	MIS \uparrow	FID \downarrow	LPIPS \uparrow
5	21.76	0.820	6.52	0.0510	87.6	0.0854
25	21.16	0.813	10.03	0.0495	90.2	0.1146
50	20.93	0.812	10.22	0.0476	92.2	0.1204

Proximity to Ground Truth

First, to measure the similarity between the inpainting results and the ground truth (GT), we use the following metrics : peak signal-to-noise ratio (PSNR), L^1 loss, and structural similarity (SSIM). For each input image to be inpainted, those metrics are averaged on the set of inpainted results.

Note that all the compared methods enforce somehow, in their training loss, similarity between the reconstructed image and the ground truth, either at pixel or feature levels. From the results in Tables 3, 4, 5, 8 and 9, we observe that, on all datasets, ICT and PIC obtained slightly lower scores than BAT and DSI-VQVAE in terms of GT similarity. A possible explanation for this performance gap is that, these two methods, contrary to the two others, consider a reconstruction loss only at the image level and not at the feature level. Being similar at feature levels encourages generating images having similar low-level (pixels, contours, etc.) and higher-level semantics to the ground truth.

Perceptual Quality

Second, to measure realism in the outputs, we measure perceptual quality by using Modified Inception Score (MIS) and Fréchet Inception Distance (FID) metrics (defined by (25) and (26), respectively). These two metrics are computed directly on the whole sets of generated or ground truth images.

BAT, ICT, and DSI-VQVAE are the methods that provide the best scores on average on all datasets. On the opposite, PIC gives the worst results quantitatively and, as we will see later, also qualitatively. We argue that a possible reason for the superior performance of BAT, ICT, and DSI-VQVAE is that, with different strategies, they separate the tasks of texture and structure recovery. Each task is handled with a specific subnetwork, first reconstructing structures that then guide the texture recovery. From a more practical point of view, BAT and ICT use transformers for global structure understanding and high-level semantics at a coarse resolution and CNNs for generating textures at the original resolution. DSI-VQVAE incorporates the multiscale hierarchical organization of VQ-VAE where the information corresponding to the texture is disentangled from the one about structure and geometry. Accordingly, DSI-VQVAE incorporates two different generators respectively devoted to both levels (cf. section “[How to Achieve Multiple and Diverse Inpainting Results?](#)”). Although DSI-VQVAE and PIC are VAE-based methods, DSI-VQVAE has the advantage that first, at low resolution, it proposes

diverse completions of structure inside the hole. These different structures then guide the completion of texture at high resolution. PIC does not have this global structure completion (at least, not explicitly). All in all, splitting the estimation of coarse and fine details in two distinct steps seems like a successful approach for high-quality image inpainting.

Note also that BAT is the method that achieves the best scores in terms of realism. Indeed, as explained before, autoregressive transformers have the ability to model longer dependencies across the image than CNN-based methods, which can be crucial for image inpainting. Note that BAT outperforms the other transformer-based method ICT, especially on irregular masks and large holes. As explained in section “[Autoregressive Models](#)”, one can explain this difference by the fact that BAT was trained, not only with bidirectional attention but also with autoregressive sampling. Therefore, it creates better consistency of the reconstructed structures, especially for large missing regions. The very good results of the DSI-VQVAE method also prove that autoregressive modeling performs well for realistic image inpainting.

Finally, one can observe the influence of the complexity of the training dataset on the performance. Notice that the underlying probability distribution of CelebA-HQ dataset is semantically less complex and diverse than the one of Places2 and ImageNet, and, thus, training is more difficult in the latter cases. We hypothesize that this affects both inpainting quality and inpainting diversity. Regarding quality the average FID score on all the studied methods trained on CelebA is equal to 33.55, while in the case of Places2 and Imagenet, it is equal to 86.92 and 128.43, respectively. This gives us an idea of the difference in complexity for each particular dataset.

Inpainting Diversity

To measure diversity, we rely on the LPIPS metric. The higher the LPIPS is, the more diverse are the outputs. For each generated sample, we compute the LPIPS distance with another sample randomly selected from the other 24 results from the same corrupted image. The reported LPIPS score corresponds to this distance averaged over the 2500 selected pairs.

First and foremost, from the range of LPIPS values on the different datasets, one can again observe the influence of the complexity of the training dataset. CelebA-HQ dataset is semantically more constraint and less complex than the one of Places2 and ImageNet, leading to lower diversity in the outputs. Indeed, the LPIPS is, in average, ~ 2 times smaller on CelebA-HQ than on Places2 or ImageNet. Similarly, as expected, all the methods create more diverse samples on larger holes than on smaller holes.

These observations argue for the existence of a trade-off between inpainting quality and inpainting diversity. The harder the inpainting problem gets (on a more complex dataset or for a larger hole), the more diverse outputs will be created. This trade-off, already highlighted in Yu et al. (2021), also arises when parametrizing a method itself. We study in Table 6 the influence of the top- \mathcal{K} parameter on the

performance of the ICT algorithm. One can observe that using a smaller \mathcal{K} creates outputs that are, on the one hand, closer to GT and more realistic but, on the other hand, less diverse.

PIC is the method giving the less diverse results on all datasets. One reason could be the aforementioned disentanglement of structure and texture of BAT, DSI-VQVAE, and ICT. In practice, these three methods first attempt to produce a multiplicity of coherent structures and then fill each of the sampled structure with a deterministic texture generator. This divide-and-conquer approach makes easier the creation of diversity as it is only performed on low-resolution structures and not on the whole reconstructed output.

ICT slightly outperforms DSI-VQVAE and BAT in terms of LPIPS on the Celeba-HQ testing images. Recall that for this experiment ICT was trained on the more diverse FFHQ dataset. This observation highlights again the influence of the training dataset on the capacity of the model to create diverse outputs.

Qualitative Performance

Similar to Zheng et al. (2019), Peng et al. (2021), Wan et al. (2021), and Yu et al. (2021), for *qualitative* comparison, we select for each method the 5 samples with the highest discriminator score out of the 25 generated samples. We use pretrained discriminators given by each of the models, i.e., for PIC, the discriminator of the generative pipeline; for DSI-VQVAE, the discriminator of the texture generation module; and for ICT and BAT, the discriminator of the upsampling module. We perform this comparison on a representative selection of testing images and masks. Figures 9, 10, 11, and 12 show some results on CelebA-HQ, Places2, and ImageNet datasets for the methods PIC, DSI-VQVAE, ICT, and BAT. BAT does not provide weights for ImageNet. Remember that ICT was not trained on CelebA-HQ but on FFHQ. Additional visual results are also given in the Appendix.

At first glance, we observe that DSI-VQVAE, ICT, and BAT provide more plausibly visual results than PIC. PIC struggles to recover information on less constrained datasets, like Places2 and Imagenet, and creates strong artifacts when applied to large missing regions (see second examples in Figs. 10 and 12). Among these methods, BAT and ICT propose the most realistic outputs. For instance, in Fig. 9, PIC generates results that do not maintain the proportions and harmony of a face (see the second example). DSI-VQVAE does not have a full understanding of the image either: for example, in the second example in Fig. 9 and the third example in Fig. 12, one eye is visible in the input image, but the other is not. On the opposite, transformer-based methods are able to reconstruct a left eye similar to the right. This can be explained by the capability of transformers to have a global structure understanding and high-level semantics. Other examples strengthening this observation are the first images of Fig. 10, where the inpainting of the snow is sometimes not realistic, and all the ImageNet results in Fig. 12.

When images contain strong structures, like Figs. 10 and 11, transformer-based methods again estimate more realistic reconstructions. This can be explained by

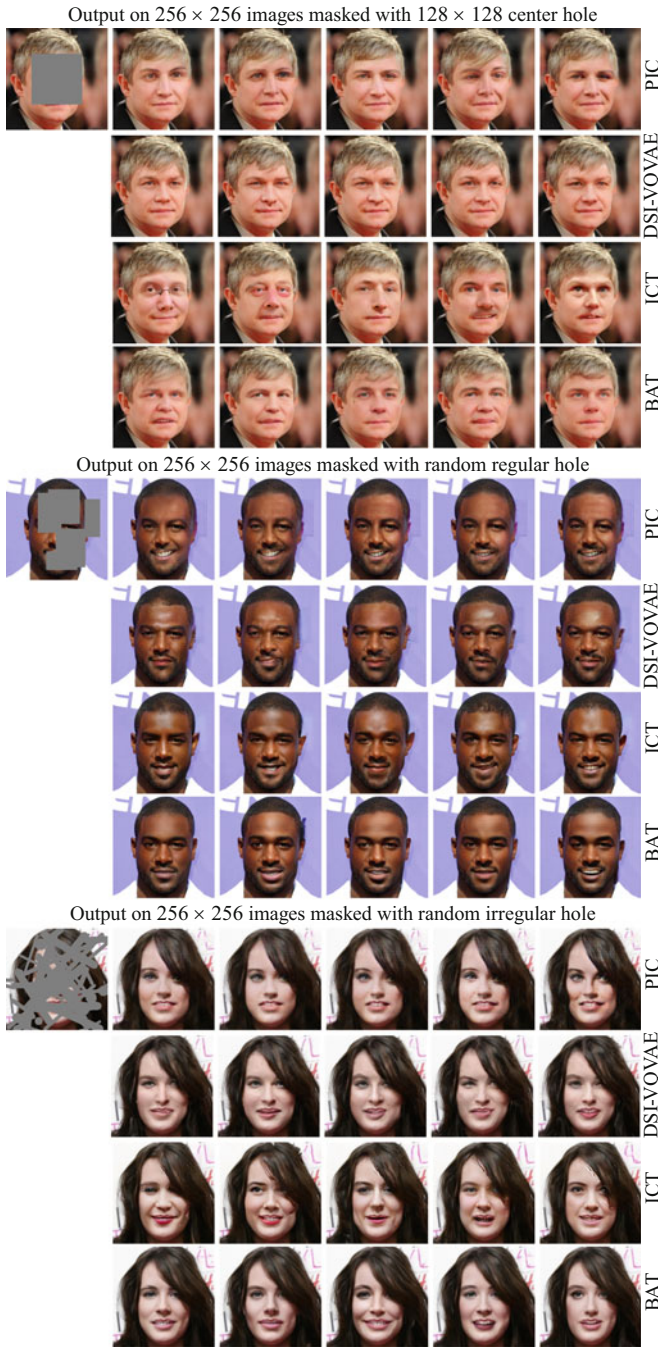


Fig. 9 Diverse inpainting output on 256×256 images from Celeba dataset with center, random regular, and random irregular masks. For each method, out of 25 generated samples, the five samples with highest discriminator score are displayed

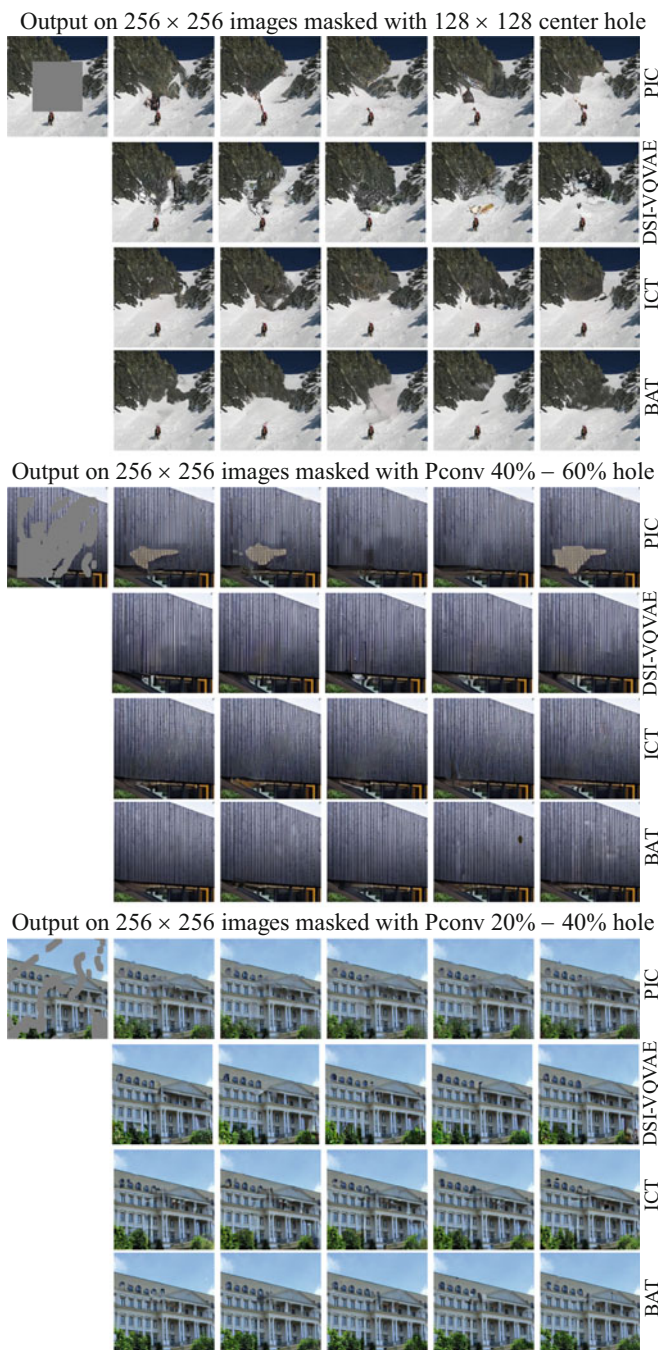


Fig. 10 Diverse inpainting output on 256×256 images from Places2 dataset with center and irregular masks with various proportion of hidden pixels. For each method, out of 25 generated samples, the 5 samples with highest discriminator score are displayed

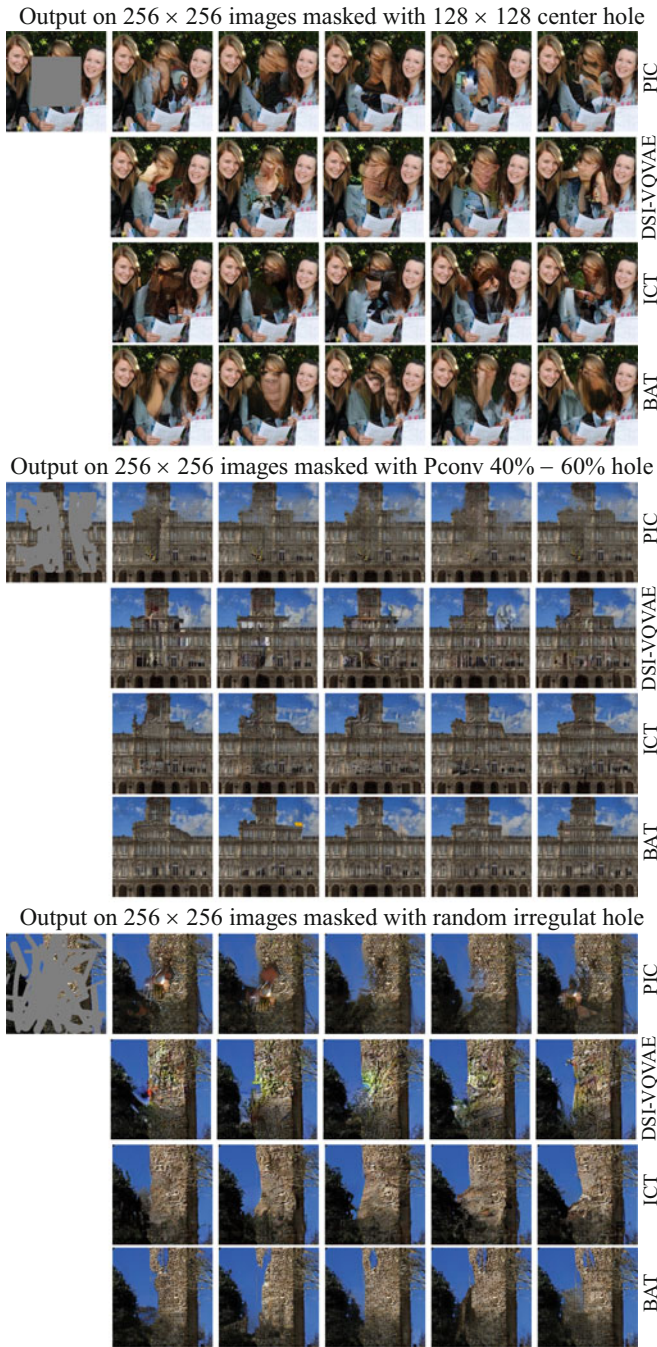


Fig. 11 Diverse inpainting output on 256×256 images from Places2 dataset with center and irregular masks with various proportion of hidden pixels. For each method, out of 25 generated samples, the five samples with highest discriminator score are displayed

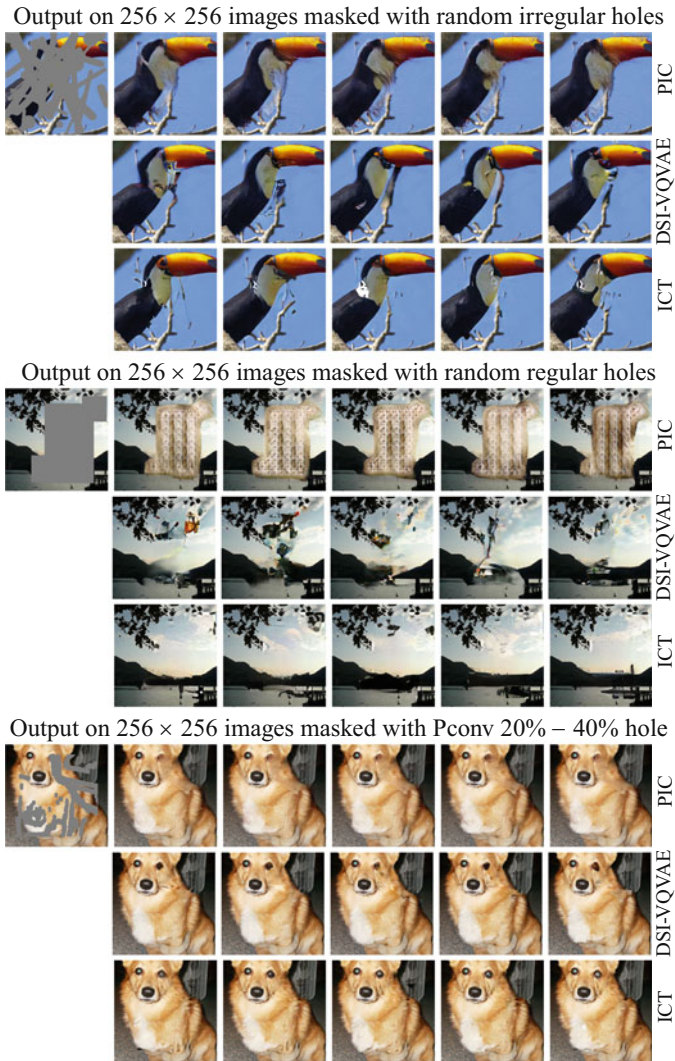


Fig. 12 Diverse inpainting output on 256×256 images from ImageNet dataset with center and irregular masks with various proportion of hidden pixels. For each method, out of 25 generated samples, the five samples with highest discriminator score are displayed

the fact that they include previously predicted tokens in the training objective, and thus, global consistency is imposed over the results. This consistency shall avoid problems in the center of big holes. In some situations, such as the middle example in Fig. 10, the structure and texture disentanglement of DSI-VQVAE also provides good reconstructions.

In terms of diversity, transformer-based methods are visually more diverse. For example, in Fig. 9, each transformer-based inpainted face corresponds to a different expression or different person, while in the case of DSI-VQVAE all generated faces are very similar. Also, in Fig. 11, even if one could imagine the result quite deterministic, ICT and BAT aim to propose multiple possibilities. Note the multiplicity of structures obtained by ICT compared to DSI-VQVAE and PIC in the chest of the dog or the skyline in Fig. 12.

Regarding the difference across datasets, while methods trained on CelebA-HQ all obtain satisfactory results (Fig. 9), results on Places2 (Fig. 11) and ImageNet (Fig. 12) are often not visually satisfactory. Also, as already noticed numerically, diversity is less visible on these two datasets. This is probably because the models have difficulties learning the underlying multimodal distribution of these complex and diverse datasets. This demonstrates the need for further research on the topic to be able to deal with real inpainting scenarios.

Influence of the Occlusion Type. We summarize here our observations related to the influence of the shape of the missing region on the reconstruction quality and diversity. First, as expected, for all methods, when the hole is larger, the generated reconstructions are more diverse but farther away to the ground-truth image. Additionally, we visually observed that PIC may produce strong artifacts when tested on large missing regions, which is also quantitatively attested by its bad realism (MIS,FID) scores Table 5 on irregular and central holes. The superior performance of BAT on this kind of degradation seem to acknowledge the advantage of autoregressive sampling for filling large missing regions. Also notice that, although ICT and BAT were only trained on irregular masks, we do not observe a drop in performance while performing inpainting on regular masks, for instance, on the central one. This shows the capacity of those methods to generalize to unseen type of missing regions.

Computational Time. Despite image quality, an important aspect that should be considered when choosing an inpainting method is its inference time. In Table 7, for the four analyzed methods, we give the average runtime to sample one inpainting result from a central hole on a 256×256 input image. We run the experiments on a single P100 GPU. Despite showing lower inpainting quality or diversity (see before), PIC is tremendously faster to run than all the other methods (~ 100 times faster than DSI-VQVAE and ICT and ~ 50 times faster than BAT). While providing good results, inference time on autoregressive or transformer-based methods can be prohibitive for time-restricted applications.

Table 7 Average runtime to sample one inpainting result on a single P100 GPU for the four compared methods. Experiments conducted for **central** masks

Method	Time (s)
PIC	0.4
DSI-VQVAE	55
ICT	43
BAT	21

Conclusions

In this chapter, we have tackled the question of whether generative methods are a suitable strategy to obtain multiple solutions to problems that do not have a unique solution. By focusing on the inpainting problem, we have reviewed the main generative models and recent learning-based image completion methods for multiple and diverse inpainting. We have compared the methods with available code and model weights on three public datasets. We have shown that the transformer-based method BAT (or BAT-Fill) and the VQ-VAE-based method DSI-VQVAE provide the best results in both inpainting quality and multiple inpainting diversity. This is true both quantitatively and qualitatively. Our analysis highlights that their advantageous results are due to their strategy that consists in, first, sampling multiple structures inside the missing regions, and, second, generating textures at higher resolution in a deterministic way. The PIC method is, however, computationally way faster than the concurrence. Moreover, our analysis shows that the multiple inpainting problem is not solved yet, as the results lack of diversity or in general visually satisfactory results. The difficulty of learning the probability distribution depending on the training dataset is also evident from our study. Therefore, we argue that most efforts should be made on improving and exploring new generative strategies to enhance both the quality and diversity of the solutions of such ill-posed inverse problem with multiple solutions. For instance, following the spirit of structure/texture division, one could further separate the problem into different subtasks or tackle different regions of the scene separately. Another way to improve inpainting quality would be to have a control of the solution by bounding it through an input condition such as the semantic of the object you want to fill-in or by a reference image, among others. Finally, the computational burden of some of the transformer-based or autoregressive methods is prohibitive for sampling a high number of solutions in reasonable time. We think that this limitation has been overlooked for the purpose of image quality but should be now primarily addressed.

Appendix

Additional Quantitative Results

We provide in this section additional quantitative results on Places2 and ImageNet. Results from Tables 8 and 9 were conducted in the same conditions as Tables 4 and 5 but with 256×256 **resized** images instead of center-cropped images. Note that, in average, on both Places2 and ImageNet, the difference between methods is very similar when computed on resized or cropped images. The modification in aspect ratio due to the resize operation does not impede the results, even for models that were trained on “real” aspect ratios. The main reason for this is that the aspect ratio is not drastically changed when resizing Places2 and ImageNet images.

Table 8 Quantitative comparison of three pluralistic image inpainting methods (PIC, DSI-VQVAE, ICT) on 256×256 **resized** images from **Places2**

Mask	Method	Similarity to GT			Realism		Diversity
		PSNR \uparrow	SSIM \uparrow	L ¹ \downarrow	MIS \uparrow	FID \downarrow	LPIPS \uparrow
Irregular <20%	PIC	29.86	0.934	2.14	0.0489	32.3	0.0055
	DSI-VQVAE	30.64	0.948	2.30	0.0533	20.0	0.0214
	ICT	29.05	0.939	3.83	0.0450	23.0	0.0224
Irregular [20%, 40%]	PIC	22.98	0.808	7.06	0.0394	91.0	0.0375
	DSI-VQVAE	23.04	0.832	6.92	0.0443	64.4	0.0789
	ICT	22.11	0.818	8.81	0.0423	72.8	0.0831
Irregular [40%, 60%]	PIC	19.01	0.649	14.71	0.0273	144.2	0.1357
	DSI-VQVAE	19.15	0.684	13.90	0.0287	115.0	0.1700
	ICT	18.50	0.669	15.78	0.0330	127.4	0.1755
Central 128 \times 128	PIC	19.50	0.797	10.27	0.0335	104.5	0.1129
	DSI-VQVAE	19.46	0.797	10.60	0.0387	94.6	0.1364
	ICT	19.42	0.796	11.72	0.0352	101.0	0.1284
Random regular	PIC	20.80	0.773	10.95	0.0359	93.8	0.1152
	DSI-VQVAE	21.15	0.791	10.48	0.0426	79.0	0.1233
	ICT	21.03	0.787	11.51	0.0382	84.3	0.1239
Random irregular	PIC	19.91	0.640	13.85	0.0246	157.7	0.1023
	DSI-VQVAE	20.05	0.682	12.98	0.0329	116.5	0.1539
	ICT	19.10	0.662	15.41	0.0285	131.4	0.1607
Average	PIC	22.01	0.767	9.83	0.0349	103.9	0.0848
	DSI-VQVAE	22.25	0.789	9.53	0.0401	81.6	0.1140
	ICT	21.54	0.779	11.18	0.0370	90.0	0.1157

Another explanation is that the training datasets are large enough and the models have enough capacity for being robust to such a transformation.

Additional Qualitative Results

In Figs. 13 and 14 we show additional inpainting visual results on Celeba-HQ and ImageNet datasets.

Table 9 Quantitative comparison of three pluralistic image inpainting methods (PIC, DSI-VQVAE, ICT) on 256×256 **resized** images from **ImageNet**

Mask	Method	Similarity to GT			Realism		Diversity
		PSNR \uparrow	SSIM \uparrow	L ¹ \downarrow	MIS \uparrow	FID \downarrow	LPIPS \uparrow
Irregular <20%	PIC	31.37	0.944	1.82	0.1885	21.5	0.0028
	DSI-VQVAE	31.83	0.952	2.08	0.1913	12.8	0.0175
	ICT	30.21	0.946	3.41	0.2002	12.2	0.0203
Irregular [20%, 40%]	PIC	23.13	0.807	6.91	0.1401	93.7	0.0323
	DSI-VQVAE	23.45	0.825	6.72	0.1617	61.8	0.0790
	ICT	22.36	0.817	8.34	0.1739	52.1	0.0810
Irregular [40%, 60%]	PIC	18.39	0.636	15.84	0.0497	198.0	0.1314
	DSI-VQVAE	18.95	0.672	14.14	0.0737	147.9	0.1901
	ICT	18.34	0.663	15.85	0.0822	120.4	0.1764
Central 128 \times 128	PIC	19.31	0.795	10.35	0.0583	153.9	0.1091
	DSI-VQVAE	19.47	0.800	10.25	0.0700	172.1	0.1293
	ICT	19.91	0.796	11.27	0.0790	120.3	0.1247
Random regular	PIC	19.63	0.745	13.13	0.0690	150.5	0.1071
	DSI-VQVAE	20.13	0.769	11.59	0.1048	113.8	0.1457
	ICT	20.13	0.766	12.56	0.1028	101.7	0.1376
Random irregular	PIC	19.70	0.618	14.04	0.0457	194.6	0.1021
	DSI-VQVAE	20.11	0.665	12.85	0.0642	155.9	0.1648
	ICT	18.94	0.649	15.33	0.0859	131.2	0.1652
Average	PIC	21.92	0.758	10.35	0.0919	135.4	0.0949
	DSI-VQVAE	22.32	0.781	9.61	0.1110	107.7	0.1211
	ICT	21.64	0.773	11.13	0.1207	89.7	0.1175

Acknowledgments PV, CB, and AB acknowledge the EU Horizon 2020 research and innovation program NoMADS (Marie Skłodowska-Curie grant agreement No 777826). SP acknowledges the Leverhulme Trust Research Project Grant “Unveiling the invisible: Mathematics for Conservation in Arts and Humanities.” CB and PV also acknowledge partial support by MICINN/FEDER UE project, ref. PGC2018-098625-B-I00, and RED2018-102511-T. AB also acknowledges the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01). SH acknowledges the French Ministry of Research through a CDSN grant of ENS Paris-Saclay.

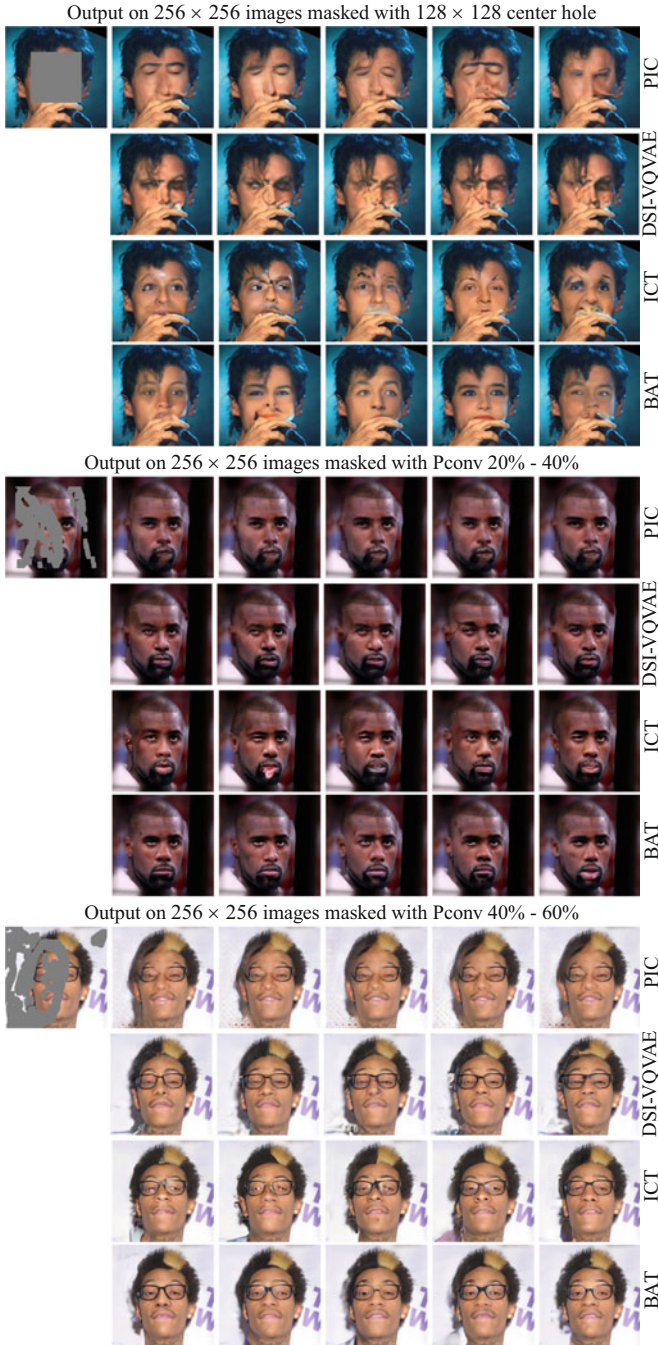


Fig. 13 Diverse inpainting output on 256×256 images from Celeba dataset with center, and irregular masks. For each method, out of 25 generated samples, the 5 samples with highest discriminator score are displayed

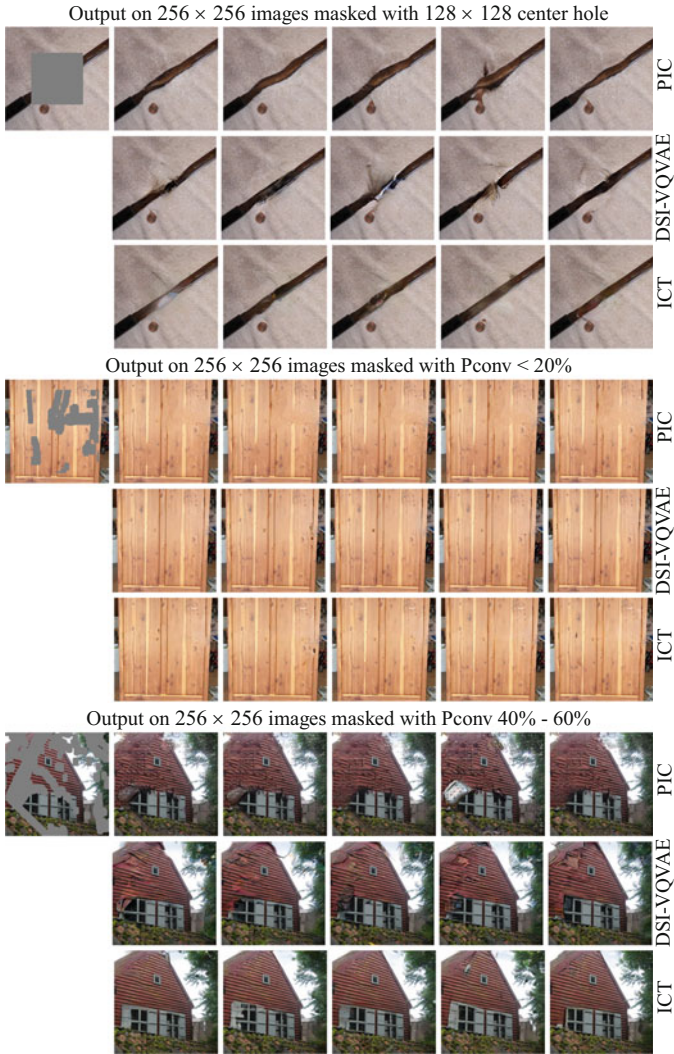


Fig. 14 Diverse inpainting output on 256×256 images from ImageNet dataset with centered and irregular masks with different hidden proportions. For each method, out of 25 generated samples, the 5 samples with highest discriminator score are displayed

References

Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A variational framework for exemplar-based image inpainting. *Int. J. Comput. Vis.* **93**(3), 319–347 (2011)

Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: *International Conference on Machine Learning*, pp. 214–223. PMLR (2017)

- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Aujol, J.-F., Ladjal, S., Masnou, S.: Exemplar-based inpainting from a variational point of view. *SIAM J. Math. Anal.* **42**(3), 1246–1285 (2010)
- Baatz, W., Fornasier, M., Markowich, P.A., bibiane Schönlieb, C.: Inpainting of ancient austrian frescoes. In: *Conference Proceedings of Bridges*, pp. 150–156 (2008)
- Ballester, C., Bertalmío, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.* **10**(8), 1200–1211 (2001)
- Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Cvae-gan: fine-grained image generation through asymmetric training. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2745–2754 (2017)
- Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch. In: *ACM SIGGRAPH 2009 papers on – SIGGRAPH’09*. ACM Press (2009)
- Barratt, S., Sharma, R.: A note on the inception score (2018). arXiv preprint arXiv:1801.01973
- Bertalmío, M., Bertozzi, A., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE Computer Society (2001)
- Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH’00*, pp. 417–424. ACM Press/Addison-Wesley Publishing Co (2000)
- Bertozzi, A.L., Esedoglu, S., Gillette, A.: Inpainting of binary images using the cahn–hilliard equation. *IEEE Trans. Image Process.* **16**(1), 285–291 (2007)
- Bevilacqua, M., Aujol, J.-F., Biasutti, P., Brédif, M., Bugeau, A.: Joint inpainting of depth and reflectance with visibility estimation. *ISPRS J. Photogram. Rem. Sens.* **125**, 16–32 (2017)
- Biasutti, P., Aujol, J.-F., Brédif, M., Bugeau, A.: Diffusion and inpainting of reflectance and height LiDAR orthoimages. *Comput. Vis. Image Underst.* **179**, 31–40 (2019)
- Bornard, R., Lecan, E., Laborelli, L., Chenot, J.-H.: Missing data correction in still images and image sequences. In: *Proceedings of the Tenth ACM International Conference on Multimedia – MULTIMEDIA’02*. ACM Press (2002)
- Bornemann, F., März, T.: Fast image inpainting based on coherence transport. *J. Math. Imag. Vis.* **28**(3), 259–278 (2007)
- Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space (2015). arXiv preprint arXiv:1511.06349
- Buyssens, P., Daisy, M., Tschumperle, D., Lezoray, O.: Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. *IEEE Trans. Image Process.* **24**(6), 1809–1824 (2015)
- Cai, N., Su, Z., Lin, Z., Wang, H., Yang, Z., Ling, B.W.-K.: Blind inpainting using the fully convolutional neural network. *Vis. Comput.* **33**(2), 249–261 (2015)
- Cai, W., Wei, Z.: Piigan: generative adversarial networks for pluralistic image inpainting. *IEEE Access* **8**, 48451–48463 (2020)
- Calatroni, L., d’Autume, M., Hocking, R., Panayotova, S., Parisotto, S., Ricciardi, P., Schönlieb, C.-B.: Unveiling the invisible: mathematical methods for restoring and interpreting illuminated manuscripts. *Herit. Sci.* **6**(1), 56 (2018)
- Cao, F., Gousseau, Y., Masnou, S., Pérez, P.: Geometrically guided exemplar-based inpainting. *SIAM J. Imag. Sci.* **4**(4), 1143–1179 (2011)
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: *European Conference on Computer Vision*, pp. 213–229. Springer (2020)
- Caselles, V., Morel, J.-M., Sbert, C.: An axiomatic approach to image interpolation. *IEEE Trans. Image Process.* **7**(3), 376–386 (1998)
- Chan, T.F., Shen, J.: Nontexture inpainting by curvature-driven diffusions. *J. Vis. Commun. Image Rep.* **12**(4), 436–449 (2001)

- Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: Pixlsnail: An improved autoregressive generative model. In: International Conference on Machine Learning, pp. 864–872. PMLR (2018)
- Chen, Y., Li, Y., Guo, H., Hu, Y., Luo, L., Yin, X., Gu, J., Toumoulin, C.: CT metal artifact reduction method based on improved image segmentation and sinogram in-painting. *Math. Probl. Eng.* **2012**, 1–18 (2012)
- Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **13**(9), 1200–1212 (2004)
- Dahl, R., Norouzi, M., Shlens, J.: Pixel recursive super resolution. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5439–5448 (2017)
- Dapogny, A., Cord, M., Pérez, P.: The missing data encoder: cross-channel image completion with hide-and-seek adversarial network. *Proc. AAAI Conf. Artif. Intell.* **34**(07), 10688–10695 (2020)
- Demanet, L., Song, B., Chan, T.: Image inpainting by correspondence maps: a deterministic approach. *Appl. Comput. Math.* **1100**, 217–50 (2003)
- Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint arXiv:1810.04805
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale (2020). arXiv preprint arXiv:2010.11929
- Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE (1999)
- Eller, M., Fornasier, M.: Rotation invariance in exemplar-based image inpainting. In: Variational Methods: In Maitine, B., Gabriel, P., Christoph, S., Jean-Baptiste, C., Thomas, H. (eds.), *Imaging and Geometric Control*, pp. 108–183. De Gruyter, Berlin, Boston (2017). <https://doi.org/10.1515/9783110430394-004>
- Esedoglu, S., Shen, J.: Digital inpainting based on the mumford–shah–euler image model. *Eur. J. Appl. Math.* **13**(04), 353–370 (2002)
- Fawzi, A., Samulowitz, H., Turaga, D., Frossard, P.: Image inpainting through neural networks hallucinations. In: IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop, pp. 1–5. IEEE (2016)
- Fedorov, V., Arias, P., Facciolo, G., Ballester, C.: Affine invariant self-similarity for exemplar-based inpainting. In: Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. SCITEPRESS – Science and Technology Publications (2016)
- Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27**, 2672–2680 (2014)
- Grossauer, H.: Inpainting of movies using optical flow. In: *Mathematics in Industry*, pp. 151–162. Springer, Berlin/Heidelberg (2006)
- Grossauer, H., Scherzer, O.: Using the complex ginzburg–landau equation for digital inpainting in 2d and 3d. In: *Scale Space Methods in Computer Vision*, pp. 225–236. Springer, Berlin/Heidelberg (2003)
- Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., Murphy, K.: Pixcolor: Pixel recursive colorization (2017). arXiv preprint arXiv:1705.07208
- Guillemot, C., Le Meur, O.: Image inpainting: overview and recent advances. *IEEE Sig. Process. Mag.* **31**(1), 127–144 (2014)
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans (2017). arXiv preprint arXiv:1704.00028
- Guo, Z., Chen, Z., Yu, T., Chen, J., Liu, S.: Progressive image inpainting with full-resolution residual network. In: Proceedings of the 27th ACM International Conference on Multimedia, MM’19, New York, pp. 2496–2504. Association for Computing Machinery (ACM) (2019)

- Han, X., Wu, Z., Huang, W., Scott, M.R., Davis, L.S.: Finet: compatible and diverse fashion image inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4481–4491 (2019)
- Hays, J., Efros, A.A.: Scene completion using millions of photographs. *ACM Trans. Graph.* **26**(3), 87–94 (2007)
- Hervieu, A., Papadakis, N., Bugeau, A., Gargallo, P., Caselles, V.: Stereoscopic image inpainting: distinct depth maps and images inpainting. In: 2010 20th International Conference on Pattern Recognition, pp. 4101–4104. IEEE (2010)
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inf. Process. Syst.* **30**, 6629–6640 (2017)
- Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. In: *ACM Transactions on Graphics (ToG)*, vol. **36**(4), pp. 1–14. ACM, New York, NY, USA (2017)
- Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711. Springer (2016)
- Kang, S.H., Chan, T., Soatto, S.: Inpainting from multiple views. In: Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission. IEEE Computer Society (2002)
- Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (2018)
- Karras, T., Laine, S., and Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
- Kettunen, M., Härkönen, E., Lehtinen, J.: E-lpips: robust perceptual image similarity via random transformation ensembles (2019). arXiv preprint arXiv:1906.03973
- Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2013)
- Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. *Found. Trends@ Mach. Learn.* **12**(4), 307–392 (2019)
- Köhler, R., Schuler, C., Schölkopf, B., Harmeling, S.: Mask-specific inpainting with deep neural networks. In: Jiang, X., Hornegger, J., Koch, R. (eds.) *Pattern Recognition*, pp. 523–534, Springer International Publishing, Cham (2014)
- Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer (2021). arXiv preprint arXiv:2102.04432
- Kumar, V., Mukherjee, J., Mandal, S.K.D.: Image inpainting through metric labeling via guided patch mixing. *IEEE Trans. Image Process.* **25**(11), 5212–5226 (2016)
- Lahiri, A., Jain, A.K., Agrawal, S., Mitra, P., Biswas, P.K.: Prior guided GAN based semantic inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13696–13705 (2020)
- Le Meur, O., Ebdelli, M., Guillemot, C.: Hierarchical super-resolution-based inpainting. *IEEE Trans. Image Process.* **22**(10), 3779–3790 (2013)
- Lempitsky, V., Vedaldi, A., Ulyanov, D.: Deep image prior. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9446–9454. IEEE (2018)
- Li, J., He, F., Zhang, L., Du, B., Tao, D.: Progressive reconstruction of visual structure for image inpainting. In: 2019 IEEE/CVF International Conference on Computer Vision. IEEE (2019)
- Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7760–7768 (2020)
- Liao, L., Hu, R., Xiao, J., Wang, Z.: Edge-aware context encoder for image inpainting. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3156–3160. IEEE (2018)
- Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: European Conference on Computer Vision, pp. 89–105 (2018)

- Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: *Computer Vision – ECCV 2020*, pp. 725–741. Springer International Publishing (2020)
- Liu, H., Jiang, B., Xiao, Y., Yang, C.: Coherent semantic attention for image inpainting. In: *2019 IEEE/CVF International Conference on Computer Vision*. IEEE (2019)
- Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J.: Pd-gan: probabilistic diverse gan for image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9371–9381 (2021)
- Mansfield, A., Prasad, M., Rother, C., Sharp, T., Kohli, P., Gool, L.V.: Transforming image completion. In: *Proceedings of the British Machine Vision Conference 2011*. British Machine Vision Association (2011)
- Mao, Q., Lee, H.-Y., Tseng, H.-Y., Ma, S., Yang, M.-H.: Mode seeking generative adversarial networks for diverse image synthesis. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1429–1437 (2019)
- Masnou, S., Morel, J.-M.: Level lines based disocclusion. In: *Proceedings 1998 International Conference on Image Processing*. ICIP98 (Cat. No.98CB36269). IEEE Computer Society (1998)
- Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: EdgeConnect: generative image inpainting with adversarial edge learning. In: *The IEEE International Conference on Computer Vision Workshops* (2019)
- Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Video inpainting of complex scenes. *SIAM J. Imag. Sci.* **7**(4), 1993–2019 (2014)
- Nitzberg, M., Mumford, D., Shiota, T.: *Filtering, Segmentation and Depth*. Springer, Berlin/Heidelberg (1993)
- Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G., Lockhart, E., Cobo, L., Stimberg, F., et al.: Parallel wavenet: Fast high-fidelity speech synthesis. In: *International Conference on Machine Learning*, pp. 3918–3926. PMLR (2018)
- Oord, A.V.D., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with pixelcnn decoders. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4797–4805 (2016)
- Papafitsoros, K., Schönlieb, C.B.: A combined first and second order variational approach for image reconstruction. *J. Math. Imag. Vis.* **48**(2), 308–338 (2013)
- Parisotto, S., Lellmann, J., Masnou, S., Schönlieb, C.-B.: Higher-order total directional variation: imaging applications. *SIAM J. Imag. Sci.* **13**(4), 2063–2104 (2020)
- Parisotto, S., Vitoria, P., Ballester, C., Bugeau, A., Reynolds, S., Schönlieb, C.-B.: *The Art of Inpainting – A Monograph on Mathematical Methods for the Virtual Restoration of Illuminated Manuscripts* (2022) (submitted)
- Park, T., Liu, M.-Y., Wang, T.-C., Zhu, J.-Y.: Semantic image synthesis with spatially-adaptive normalization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2337–2346 (2019)
- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544. IEEE (2016)
- Peng, J., Liu, D., Xu, S., Li, H.: Generating diverse structure for image inpainting with hierarchical vq-vae. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10775–10784 (2021)
- Peter, P., Weickert, J.: Compressing images with diffusion- and exemplar-based inpainting. In: *Lecture Notes in Computer Science*, pp. 154–165. Springer International Publishing (2015)
- Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: *Advances in Neural Information Processing Systems*, pp. 14866–14876 (2019)
- Ren, J.S., Xu, L., Yan, Q., Sun, W.: Shepard convolutional neural networks. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. *NIPS'15*, Cambridge, MA, vol. 1, pp. 901–909. The MIT Press (2015)

- Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: StructureFlow: image inpainting via structure-aware appearance flow. In: 2019 IEEE/CVF International Conference on Computer Vision, pp. 181–190. IEEE (2019)
- Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks (2017). arXiv preprint arXiv:1706.04987
- Rott Shaham, T., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: International Conference on Computer Vision (2019)
- Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization (2017). arXiv preprint arXiv:1705.04258
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
- Ružić, T., Cornelis, B., Platiša, L., Pižurica, A., Dooms, A., Philips, W., Martens, M., Mey, M.D., Daubechies, I.: Virtual restoration of the ghent altarpiece using crack detection and inpainting. In: *Advanced Concepts for Intelligent Vision Systems*, pp. 417–428. Springer, Berlin/Heidelberg (2011)
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: *Advances in Neural Information Processing Systems* (2016)
- Schonlieb, C.-B.: *Partial Differential Equation Methods for Image Inpainting*. Cambridge University Press, New York (2015)
- Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**(3), 1019–1043 (2002)
- Shen, J., Kang, S.H., Chan, T.F.: Euler’s elastica and curvature-based inpainting. *SIAM J. Appl. Math.* **63**(2), 564–592 (2003)
- Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. *Adv. Neural Inf. Process. Syst.* **28**, 3483–3491 (2015)
- Sun, J., Yuan, L., Jia, J., Shum, H.-Y.: Image completion with structure propagation. *ACM Trans. Graph.* **24**(3), 861–868 (2005)
- Tai, X.-C., Osher, S., Holm, R.: Image inpainting using a TV-stokes equation. In: *Image Processing Based on Partial Differential Equations*, pp. 3–22. Springer, Berlin/Heidelberg (2007)
- Tovey, R., Benning, M., Brune, C., Lagerwerf, M.J., Collins, S.M., Leary, R.K., Midgley, P.A., Schönlieb, C.-B.: Directional sinogram inpainting for limited angle tomography. *Inverse Probl.* **35**(2), 024004 (2019)
- Tschumperle, D., Deriche, R.: Vector-valued image regularization with PDEs: a common framework for different applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(4), 506–517 (2005)
- van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6309–6318 (2017)
- Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: *International Conference on Machine Learning*, pp. 1747–1756. PMLR (2016)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)
- Vitoria, P., Ballester, C.: Automatic flare spot artifact detection and removal in photographs. *J. Math. Imag. Vis.* **61**(4), 515–533 (2019)
- Vitoria, P., Sintés, J., Ballester, C.: Semantic image inpainting through improved Wasserstein generative adversarial networks. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. VISAPP*, vol. 4, pp. 249–260. INSTICC, SciTePress (2019)
- Vitoria, P., Sintés, J., Ballester, C.: Semantic image completion through an adversarial strategy. In: *Communications in Computer and Information Science*, pp. 520–542. Springer International Publishing (2020)

- Vo, H.V., Duong, N.Q.K., Pérez, P.: Structural inpainting. In: 2018 ACM Multimedia Conference on Multimedia Conference, MM'18, New York, pp. 1948–1956. Association for Computing Machinery (ACM) (2018)
- Wan, Z., Zhang, J., Chen, D., Liao, J.: High-fidelity pluralistic image completion with transformers (2021). arXiv preprint arXiv:2103.14031
- Wang, Z.B., Alan, C.S., Hamid, R.S.: Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process* **13**(4), 600–612 (2004)
- Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 329–338. Curran Associates Inc., Montréal, Canada (2018)
- Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004. IEEE (2004)
- Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5840–5848. IEEE (2019)
- Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: image inpainting via deep feature rearrangement. In: Computer Vision – ECCV 2018, pp. 3–19. Springer International Publishing (2018)
- Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 6721–6729. IEEE (2017)
- Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5791–5800 (2020)
- Yeh, R.A., Chen, C., Lim, T.Y., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 5485–5493. IEEE (2017)
- Yi, K., Guo, Y., Fan, Y., Hamann, J., Wang, Y.G.: Cosmovae: variational autoencoder for CMB image inpainting (2020a). arXiv preprint arXiv:2001.11651
- Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7505–7514. IEEE (2020b)
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Free-form image inpainting with gated convolution. In: International Conference on Computer Vision, pp. 4470–4479 (2019)
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5505–5514. IEEE (2018)
- Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., Miao, C.: Diverse image inpainting with bidirectional and autoregressive transformers (2021). arXiv preprint arXiv:2104.12335
- Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1486–1494. IEEE (2019)
- Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: European Conference on Computer Vision, pp. 1–17. Springer (2020)
- Zhang, H., Hu, Z., Luo, C., Zuo, W., Wang, M.: Semantic image inpainting with progressive generative networks. In: 2018 ACM Multimedia Conference on Multimedia Conference, MM'18, pp. 1939–1947. ACM Press (2018a)
- Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Conference on Computer Vision and Pattern Recognition (2018b)

- Zhao, J., Han, J., Shao, L., Snoek, C.G.: Pixelated semantic colorization. *Int. J. Comput. Vis.* **128**(4), 818–834 (2020a)
- Zhao, L., Mo, Q., Lin, S., Wang, Z., Zuo, Z., Chen, H., Xing, W., Lu, D.: UCTGAN: diverse image inpainting based on unsupervised cross-space translation. In: *Conference on Computer Vision and Pattern Recognition*, pp. 5741–5750 (2020b)
- Zheng, C., Cham, T.-J., Cai, J.: Pluralistic image completion. In: *Conference on Computer Vision and Pattern Recognition*, pp. 1438–1447 (2019)
- Zheng, C., Cham, T.-J., Cai, J.: Tfill: image completion via a transformer-based architecture (2021). arXiv preprint arXiv:2104.00845
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(6), 1452–1464 (2017)
- Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Multimodal image-to-image translation by enforcing bi-cycle consistency. In: *Advances in Neural Information Processing Systems*, pp. 465–476 (2017)



Analysis of Different Losses for Deep Learning Image Colorization

21

Coloma Ballester, Hernan Carrillo, Michaël Clément,
and Patricia Vitoria

Contents

Introduction	822
Losses in the Colorization Literature	823
Error-Based Losses	824
Generative Adversarial Network-Based Losses	826
Distribution-Based Losses	828
Proposed Colorization Framework	831
Detailed Architecture	831
Quantitative Evaluation Metrics Used in Colorization Methods	833
Experimental Analysis	836
Quantitative Evaluation	836
Qualitative Evaluation	838
Generalization to Archive Images	844
Conclusion	844
References	844

Abstract

Image colorization aims to add color information to a grayscale image in a realistic way. Recent methods mostly rely on deep learning strategies. While learning to automatically colorize an image, one can define well-suited objective functions related to the desired color output. Some of them are based on a specific type of error between the predicted image and ground truth one, while other

C. Ballester · P. Vitoria
University Pompeu Fabra, Barcelona, Spain
e-mail: coloma.ballester@upf.edu; patricia.vitoria@upf.edu

H. Carrillo · M. Clément (✉)
LaBRI, CNRS, Bordeaux INP, Université de Bordeaux, Bordeaux, France
e-mail: hernan.carrillo-lindado@u-bordeaux.fr; michael.clement@labri.fr

losses rely on the comparison of perceptual properties. But, is the choice of the objective function that crucial, i.e., does it play an important role in the results? In this chapter, we aim to answer this question by analyzing the impact of the loss function on the estimated colorization results. To that goal, we review the different losses and evaluation metrics that are used in the literature. We then train a baseline network with several of the reviewed objective functions, classic L1 and L2 losses, as well as more complex combinations such as Wasserstein GAN and VGG-based LPIPS loss. Quantitative results show that the models trained with VGG-based LPIPS provide overall slightly better results for most evaluation metrics. Qualitative results exhibit more vivid colors when trained with Wasserstein GAN plus the L2 loss or again with the VGG-based LPIPS. Finally, the convenience of quantitative user studies is also discussed to overcome the difficulty of properly assessing on colorized images, notably for the case of old archive photographs where no ground truth is available.

Keywords

Image colorization · Deep learning · Loss functions · Color spaces

Introduction

Color is acknowledged to be captured by the human visual system at the first milliseconds. Color perception allows to highly increase the perceived diversity of real scenes since more than 2 million colors are identified by humans. Besides, although humans are interested in color and have used it since the dawn of humanity, full comprehension of the chromatic aspect of color is still an open problem. Color images capturing a real scene indeed include both structure information (edges, textures) which is mostly contained in the so-called black-and-white component of the image and chromatic information which, when added to the achromatic black-and-white component, provides the rich color vision of the scene image. This achromatic and chromatic dichotomy is also palpable in works of art: artists often slide between drawing strength from the massive richness of the variations on black and white and exploiting the infinite power of color, even using it as an actor on its own.

Image colorization aims to hallucinate the missing color information of a given grayscale image by, as in the case of learning-based methods, directly learning a mapping from the grayscale to the color information by minimizing a chosen objective function. The objective function favors the desired properties the estimated colorization should satisfy. Due to the ill-posed nature of the problem, in most cases, one does not aim to recover the actual ground truth color – that is, the real color of the actual scene captured in the grayscale image – but rather to produce a plausible colorization for a human observer. Accordingly, choosing the right way to train such networks is not trivial. The network could end up penalizing a good solution far away from the ground truth data or estimating an average of

all possible correct solutions. Alternatively, instead of directly learning the per-pixel chrominance information, some methods learn a per-pixel color distribution to, afterward, sample from it the color at each pixel. In principle, this could encourage the mapping to be one to many, which can be desirable. However, how to properly capitalize and train such networks to account for the different possible solutions having, both, geometric and semantic meaning remains an open problem.

This chapter aims to analyze the influence of the optimized objective function on the results of automatic deep learning methods for image colorization. Some of the chosen objective functions favor colorization results perceptually as plausible as the associated color ground truth image, no matter the pixel-wise color differences between them, while others aim to recover the ground truth values. To the best of our knowledge, there is currently no study about their influence over the results.

Additionally, besides the selected objective function used to train the model, another important choice is the color space we will work on. Almost all colorization methods work either on a Luminance–Chrominance or on the RGB color space. Only a few of them, such as Larsson et al. (2016), work on Hue–Saturation-based color spaces. Thus, together with this chapter, another chapter of the current handbook, called ▶ [Chap. 22, “Influence of Color Spaces for Deep Learning Image Colorization”](#) has been added for completeness. It focuses on the influence of color spaces. It also contains a more detailed review of the literature on image colorization and of the used datasets. We refer the reader to the mentioned chapter for these reviews.

The rest of this chapter is organized as follows. In Section [“Losses in the Colorization Literature,”](#) we first make a review of the loss functions that have been used in the field of image colorization while connecting them with the colorization-related works. Section [“Proposed Colorization Framework”](#) details the framework used to analyze the influence of the different losses, including both the chosen architecture and evaluation metrics. Finally, in Section [“Experimental Analysis,”](#) we present quantitative and qualitative colorization results on a classical image dataset, and Section [“Generalization to Archive Images”](#) shows extended results on archive images. Conclusions can be found in Section [“Conclusion.”](#)

Losses in the Colorization Literature

The objective loss function summarizes the desired properties that we want the estimated outcome to satisfy. In this section, we review the losses and evaluation methods used in the literature.

Along this chapter, a color image is assumed to be defined on a bounded domain Ω , a subset of \mathbb{R}^2 . With a slight abuse of notation, we will both use the same notation to refer to the continuous setting, where $\Omega \subset \mathbb{R}^2$ is an infinite resolution image domain and $u : \Omega \rightarrow \mathbb{R}^C$, and to the discrete setting, where Ω represents a discrete domain given by a grid of $M \times N$ pixels, $M, N \in \mathbb{N}$, and u is a function defined on this discrete Ω and with values in \mathbb{R}^C . In the latter case, u is usually given by a real-valued matrix of size $M \times N \times C$ representing the image values. Finally, C

Table 1 Losses used to train deep learning methods for image colorization. CE stands for cross-entropy and KL for Kullback–Leibler divergence

		Using GANs	Histogram prediction	User guided	Diverse	Object aware	Survey
	Cheng et al. (2015)						
	Iizuka et al. (2016)						
	Vitoria et al. (2020)						
	Nazeri et al. (2018)						
	Cao et al. (2017)						
	Yoo et al. (2019)						
	Antic (2019)						
	Larsson et al. (2016)						
	Zhang et al. (2016)						
	Mouzon et al. (2019)						
	Zhang et al. (2017)						
	He et al. (2018)						
	Deshpande et al. (2017)						
	Guadarrama et al. (2017)						
	Royer et al. (2017)						
	Kumar et al. (2021)						
	Su et al. (2020)						
	Pucci et al. (2021)						
	Kong et al. (2021)						
	winner of Gu et al. (2019)						
MAE							
smooth-L1							
MSE							
GANs							
KL on distributions							
CE on distributions							
KL for classification							
CE for classification							
neg log-likelihood							
Perceptual							

can be either equal to 3 if u is a color image or equal to 2 if the goal is to reconstruct the two chrominance channels and, thus, the input grayscale image is not modified during colorization.

Error-Based Losses

In the following, the different losses used in the literature of image colorization are described and related to some representative works that capitalize on them. Table 1 summarizes it.

MSE or squared L2 loss. Given two functions u and v defined on Ω and with values in \mathbb{R}^C , $C \in \mathbb{N}$, the so-called Mean Square Error (MSE) between u and v is defined as the squared L2 loss of their difference. That is

$$\text{MSE}(u, v) = \|u - v\|_{L^2(\Omega; \mathbb{R}^C)}^2 = \int_{\Omega} \|u(x) - v(x)\|_2^2 dx, \tag{1}$$

where $\|\cdot\|_2$ denotes the Euclidean norm in \mathbb{R}^C . In the discrete setting, it is equal to the sum of the square differences between the image values, that is

$$\text{MSE}(u, v) = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^C (u_{i,j,k} - v_{i,j,k})^2. \quad (2)$$

It has been extensively used for image colorization methods (Cheng et al. 2015; Larsson et al. 2016; Zhang et al. 2016; Iizuka et al. 2016; Isola et al. 2017; Nazari et al. 2018; Vitoria et al. 2020) (see also Table 1), where $C = 3$ if u and v are color images (usually the predicted and the ground truth data) or $C = 2$ in the case that u and v are chrominance images. Although while the training with this loss can lead to a more stable solution, it is not robust to outliers in the data and penalizes large errors while being more tolerant to small errors.

MAE or L1 loss with l^1 -coupling. The Mean Absolute Error is defined as the L1 loss with l^1 -coupling, that is

$$\text{MAE}(u, v) = \int_{\Omega} \|u(x) - v(x)\|_{l^1} dx = \int_{\Omega} \sum_{k=1}^C |u_k(x) - v_k(x)| dx. \quad (3)$$

In the discrete setting, it coincides with the sum of the absolute differences $|u_{i,j,k} - v_{i,j,k}|$. Some authors use a l^2 -coupled version of it:

$$\text{MAE}^c(u, v) = \sum_{i=1}^M \sum_{j=1}^N \sqrt{\sum_{k=1}^C (u_{i,j,k} - v_{i,j,k})^2}. \quad (4)$$

Both MAE and MAE^c losses are robust to outliers.

To ease the non-differentiability issue in the minimization of the MAE and MAE^c , some authors use the **Smooth L1** or **Huber loss**. It is simply defined by substituting the absolute value $|\cdot|$ in (3) by

$$l_H(g) = \begin{cases} \frac{1}{2}g^2 & \text{if } |g| \leq \delta \\ \delta(|g| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (5)$$

for $g \in \mathbb{R}$. Several works Su et al. (2020), Cao et al. (2017), Yoo et al. (2019), Zhang et al. (2017), He et al. (2018), and Guadarrama et al. (2017) use MAE, MAE^c , or Smooth L1 losses either alone or combined with other losses (cf. Table 1).

Previous error-based losses aim to find a solution close to the ground truth. This is counterproductive to the idea that image colorization has multiple possible solutions. Additionally, both metrics are poorly related to perceptual quality. Nonetheless, both metrics are the most used ones to train deep learning approaches. In Section “[Experimental Analysis](#),” we present some numerical results together with a comparison with other kinds of losses.

Aiming at favoring a solution keeping from the ground truth not the exact values but more perceptual or style features, the following error losses have been proposed and used for colorization purposes.

Feature Loss. The feature reconstruction loss (Gatys et al. 2016; Johnson et al. 2016) is a perceptual loss that encourages images to have similar feature representations as the ones computed by a pretrained network, denoted here by Φ . Let $\Phi_l(u)$ be the activation of the l -th layer of the network Φ when processing the image u ; if l is a convolutional layer, then $\Phi_l(u)$ will be a feature map of size $C_l \times W_l \times H_l$. The *feature reconstruction* loss is the normalized squared Euclidean distance between feature representations, that is

$$\mathcal{L}_{\text{feat}}^l(u, v) = \frac{1}{C_l W_l H_l} \|\Phi_l(u) - \Phi_l(v)\|_2^2. \quad (6)$$

It penalizes the output reconstructed image when it deviates in feature content from the target.

In our experimental analysis in Section “[Experimental Analysis](#),” we analyze the influence of the perceptual loss given by the VGG-based LPIPS (21), which was introduced in Ding et al. (2021) as a generalization of the perceptual loss above (Johnson et al. 2016).

Generative Adversarial Network-Based Losses

Aiming to favor more diverse and perceptually plausible colorization results, losses based on *Generative Adversarial Networks* (GANs) (Goodfellow et al. 2014) have been introduced in the colorization literature (Isola et al. 2017; Cao et al. 2017; Nazeri et al. 2018; Yoo et al. 2019; Vitoria et al. 2020). GANs are a kind of generative methods where the goal is to learn the probability distribution of the considered dataset by learning to generate new samples as if they were coming from that dataset. In the case of GANs, the learning is done by an adversarial learning strategy.

Vanilla GAN. The first GAN proposal by Goodfellow et al. (2014) is based on a game theory scenario between two networks competing one against another. The first network called generator, denoted by G , aims to generate samples of data as similar as possible to the ones of real data \mathcal{P}_r . The second network, called discriminator, aims to classify between real and generated data. To do so, the discriminator, denoted here by D , is trained to maximize the probability of correctly distinguishing between real examples and samples created by the generator. On the other hand, G is trained to fool the discriminator by generating realistic examples. The adversarial loss of the vanilla GAN is defined as

$$\mathcal{L}_{\text{adv}}(G_\theta, D_\phi) = \mathbb{E}_{u \sim \mathcal{P}_r} [\log D_\phi(u)] + \mathbb{E}_{v \sim \mathcal{P}_{G_\theta}} [\log(1 - D_\phi(v))], \quad (7)$$

and the min-max adversarial optimization problem is

$$\min_{G_\theta} \max_{D_\phi} \mathcal{L}_{\text{adv}}(G_\theta, D_\phi). \quad (8)$$

Wasserstein GAN. Although vanilla GANs have achieved good results in many domains, they have some drawbacks like convergence, vanishing gradients, and mode collapse problems. Therefore, some modifications from the original GAN have been proposed. For example, the *Wasserstein GAN* (WGAN), proposed by Arjovsky et al. (2017), replaces the underlying Jensen–Shannon divergence from the original proposal with the Wasserstein–1 distance (or Earth Mover distance) between two probability distributions. Then, the WGAN loss, $\mathcal{L}_{\text{adv,wgan}}$, and WGAN optimization problem can be defined as

$$\min_{G_\theta} \max_{D_\phi \in \mathcal{D}} \mathcal{L}_{\text{adv,wgan}}(G_\theta, D_\phi) = \min_{G_\theta} \max_{D_\phi \in \mathcal{D}} \left(\mathbb{E}_{u \sim \mathcal{P}_r} [D_\phi(u)] - \mathbb{E}_{v \sim \mathcal{P}_{G_\theta}} [D_\phi(v)] \right) \quad (9)$$

where \mathcal{D} denotes the set of 1-Lipschitz functions. To enforce the 1-Lipschitz condition, in Gulrajani et al. (2017), the authors propose a *Gradient Penalty* (GP) term constraining the L2 norm of the gradient while optimizing the original WGAN during training. The resulting loss for the WGAN-GP can be defined as

$$\min_{G_\theta} \max_{D_\phi} \left(\mathbb{E}_{u \sim \mathcal{P}_r} [D_\phi(u)] - \mathbb{E}_{v \sim \mathcal{P}_{G_\theta}} [D_\phi(v)] - \lambda \mathbb{E}_{\hat{u} \sim \hat{\mathcal{P}}} [(\|\nabla_{\hat{u}} D(\hat{u})\|_2 - 1)^2] \right) \quad (10)$$

where \hat{u} is a sample defined as

$$\hat{u} = tu + (1 - t)v,$$

with t uniformly sampled in $[0, 1]$ and $u \sim \mathcal{P}_r$, $v \sim \mathcal{P}_{G_\theta}$. The last term in (10) provides a tractable approximation to enforce the norm of the gradient of D to be less than 1. The authors of Gulrajani et al. (2017) motivated it by a theoretical result showing that the optimal discriminator D contains straight lines connecting samples in the ground truth space and samples in the space of generated data. Moreover, they experimentally observed that this technique exhibits good performance in practice. Finally, let us observe that the minus before the gradient penalty term in (10) corresponds to the fact that the WGAN min-max objective (10) implies maximization with respect to the discriminator parameters.

In our experimental results in Section “[Experimental Analysis](#),” we will present a comparison of several losses, and we will include a combination of WGAN loss and a VGG-based LPIPS loss. To the best of our knowledge, it has not been proposed yet.

Distribution-Based Losses

As mentioned in Section “[Introduction](#),” some authors colorize an image after learning a certain probability distribution such as a color probability distribution (Larsson et al. 2016; Zhang et al. 2016, 2017; Royer et al. 2017), or a distribution of semantic classes (Vitoria et al. 2020), or directly using it for classification purposes (Iizuka et al. 2016). The remaining of this section describes the corresponding measures of the difference between two probability distributions that have been used in the mentioned related work (see also Table 1).

Kullback–Leibler loss. The *Kullback–Leibler* (KL) loss is the directed divergence between two probability densities ρ and $\hat{\rho}$ defined in the same space \mathcal{Y} . It is defined as the relative entropy from $\hat{\rho}$ to ρ which, for discrete probability densities, is given by

$$KL(\rho||\hat{\rho}) = \sum_{y \in \mathcal{Y}} \rho(y) \log \frac{\rho(y)}{\hat{\rho}(y)}. \quad (11)$$

Here, ρ is usually taken as the ground truth density (sometimes as a Dirac delta or a one-hot vector on the ground truth value, or a regularized one) and $\hat{\rho}$ the predicted one.

Some works predict a color distribution density per pixel where the color bins are associated to a fixed 2D grid in a chrominance space (e.g., CIE Lab in Zhang et al. 2016). In Zhang et al. (2016), the final color of each pixel in the inferred color image is given by the expectation (sum over the color bin centroids weighted by the histogram). Others, such as Larsson et al. (2016), learn Hue-Saturation-based color distributions. More precisely, Larsson et al. (2016) learn the marginal distributions $\hat{\rho}^{\text{Hue}}$ and $\hat{\rho}^{\text{Chroma}}$ of Hue and Chroma, per pixel, where chroma is related to saturation by the formula $\text{Saturation} = \frac{\text{Chroma}}{\text{Value}}$ and $\text{Value} = \text{Luminance} + \frac{\text{Chroma}}{2}$. They use the KL divergence to measure the deviation between the estimated distributions and the ground truth ones. The marginal ground truth distributions, ρ^{Chroma} , ρ^{Hue} , are again defined as either a one-hot vector on the bin associated to the ground truth color or regularized version of it. Then, their loss is

$$\mathcal{L}(\rho||\hat{\rho}) = KL(\rho^{\text{Chroma}}||\hat{\rho}^{\text{Chroma}}) + \lambda c KL(\rho^{\text{Hue}}||\hat{\rho}^{\text{Hue}}) \quad (12)$$

where $c \in [0, 1]$ is the ground truth Chroma of the considered pixel and $\lambda = 5$ in Larsson et al. (2016). The authors introduce this weight depending on the Chroma multiplying the KL term on ρ^{Hue} to avoid Hue instability issues when Chroma approaches zero. For inference and to sample a color value per pixel from the estimated marginal distributions, they experimentally tested that a median-based selection (a periodically modified version in the case of Hue) gives the best results.

Besides, Vitoria et al. (2020) uses the KL loss (11) to learn, for each image, the distribution density of semantic classes, for a fixed number of classes. It provides information about the semantic content and objects present in the image.

In particular, they define the ground truth probability density ρ of semantic classes to be the output distribution of a pre-trained VGG-16 model applied to the grayscale image and $\hat{\rho}$ the estimated class distribution density.

Cross-Entropy Loss. Cross-entropy loss is used for classification problems, and it is sometimes referred to as logistic loss. For discrete densities, it is defined as

$$CE(\rho, \hat{\rho}) = - \sum_{y \in \mathcal{Y}} \rho(y) \log \hat{\rho}(y), \quad (13)$$

where, again, ρ is usually taken as the ground truth density and $\hat{\rho}$ the predicted one. In the classification context, ρ is often a one-hot vector equal to 1 on the ground truth class, or a regularized version of it. Let us also note, from (11) and (13), that there is a relationship between the Kullback–Leibler and the cross-entropy losses given by

$$CE(\rho, \hat{\rho}) = E(\rho) + KL(\rho || \hat{\rho}), \quad (14)$$

where $E(\rho)$ denotes the entropy of ρ .

Cross-entropy is used as a classification loss in Iizuka et al. (2016) where the network is trained on a large-scale dataset. The architecture is made of two encoding networks that learn local and global features and a decoder that learns the color image from these features. The classification loss is used to guide the training of the global feature network from image label estimation. It is combined with a MSE loss that compares estimated color image with the ground truth.

In Zhang et al. (2016, 2017), CE is applied on color distributions. Zhang et al. (2016) treat the colorization problem as multinomial classification by learning a mapping from the input grayscale image to a probability distribution over possible discrete chrominance values. CE compares the estimated distribution with the one of the ground truth. Zhang et al. (2017) build upon this framework and incorporate user interaction. Finally, Mouzon et al. (2019) and Pierre and Aujol (2020) stem from the resulting distributions from Zhang et al. (2016) that, in a subsequent step, are incorporated in a variational approach (Pierre et al. 2015).

Log-likelihood Maximization for Diversity. Some works propose to generate several possible colorizations, for the same input gray-level image, by sampling over possible color distributions that are often learned by maximizing the log likelihood conditioned to the grayscale image (Guadarrama et al. 2017; Royer et al. 2017; Kumar et al. 2021).

The work *Pixcolor: Pixel recursive colorization* (Guadarrama et al. 2017) colorizes an image by first learning the color distribution of images conditioned to a grayscale input. It stems from autoregressive models (Van Oord et al. 2016; Oord et al. 2016; Chen et al. 2018) that exploit the fact that a color probability distribution $p(u)$ can be in principle learned by choosing an order of the data

variables $u = (u_1, u_2, \dots, u_n) \in \mathcal{X}$, associated with the color values of a discrete color image u at its n pixels (where \mathcal{X} denotes the space of discrete color images), and exploiting the fact that the joint distribution can be decomposed as

$$p(u) = p(u_1, u_2, \dots, u_n) = p(u_1) \prod_{i=2}^n p(u_i | u_1, \dots, u_{i-1}). \quad (15)$$

As claimed by Guadarrama et al. (2017), this ordering tends to capture dependencies between pixels to ensure that, at inference, colors will be consistently selected. By working in the $YCbCr$ color space and by discretizing the Cb and Cr channels separately into 32 bins, they propose to model the conditional distribution of u given the grayscale image Y by

$$p(u^{b,r} | Y) = \prod_i p(u_i^r | u_1^{b,r}, \dots, u_{i-1}^{b,r}, Y) p(u_i^b | u_i^r, u_1^{b,r}, \dots, u_{i-1}^{b,r}, Y), \quad (16)$$

where u_i^b denotes the Cb value for pixel i , u_i^r its Cr value, and $u_i^{b,r}$ its (Cb , Cr) chrominance. They train the model using maximum likelihood, with a cross-entropy loss per pixel. Afterward, they perform high-resolution refinement to upscale the chrominance image at the dimensions of the original grayscale image.

In Royer et al. (2017), a feed-forward network followed by an autoregressive network is used to predict for each pixel a probability distribution over all possible chrominances conditioned to the luminance. They work in the Lab color space. $p(u^{a,b} | L)$ is factorized again as in (15) and (16) as the product of terms of the form $p(u_i^{a,b} | u_1^{a,b}, \dots, u_{i-1}^{a,b}, L)$, which are learned on a set of training images D by minimizing negative log-likelihood of the chrominance channels in the training data:

$$\arg \min - \sum_{u \in D} \log p(u^{a,b} | L). \quad (17)$$

L and $u^{a,b}$ denote the luminance and chrominance channels, respectively. In order to speed up the learning, Royer et al. (2017) approximate each distribution $p(u_i^{a,b} | u_1^{a,b}, \dots, u_{i-1}^{a,b}, L)$ with a mixture of ten logistic distributions.

Kumar et al. (2021) also address the generation of multiple outputs for a given grayscale image, in this case using transformers. They use a conditional autoregressive transformer (a conditional variant of Axial Transformer particular self-attention with Ho et al. 2019) to first produce a low-resolution colorization of the grayscale image (both spatial and color low resolution) that is then upsampled with two parallel networks for upsampling the spatial and color resolutions. The model is trained to minimize the negative log-likelihood of the distributions that are estimated by each network.

Several works combine distribution-based losses with error-based ones. For instance, aiming to learn the distribution of color images conditioned to a grayscale version $p(u|L)$, Deshpande et al. (2017) uses a VAE approach and log-likelihood

maximization to learn a low-dimensional (latent variables) embedding of color images, combined with error losses on the output of the decoder that favor to keep color specificity (with a L2 loss that compares the projection of the generated color and ground truth images along a top-k principal components), colorfulness (with a loss that encourages rare colors to appear), and similar gradients to the ground truth color image (with a loss that compares the gradients of the generated images with the ones of the ground truth). Moreover, the conditional distribution $p(z|L)$ of the latent variables given the grayscale image is assumed to be a Gaussian mixture and learned minimizing the conditional negative log likelihood.

The authors of Pucci et al. (2021) capitalize on capsule networks (Sabour et al. 2017) to learn a color distribution over a set of quantized colors. To that goal, they use a weighted cross-entropy loss where the weights are used to weight more rare colors, with a MSE loss on the (a, b) channels.

Kong et al. (2021) propose a multitask network in an adversarial manner that uses a MSE loss on hue, saturation, and lightness channels to perform colorization and a cross-entropy loss to learn a semantic segmentation.

Finally, it is worth mentioning that Ding et al. (2021) compare different cost functions to train a deep neural network on four low-level vision tasks, denoising, blind image deblurring, single image super resolution, and lossy image compression, although it is not done for image colorization.

In the following sections, we will present a comparison of the different loss functions for the colorization task. To do so, we propose a baseline colorization network architecture (presented in the next section) and show experimental results for the different loss functions on the same dataset.

Proposed Colorization Framework

In this section, we present the framework used to study the influence of the chosen objective loss on the estimated images colorization results. First we detail the architecture and second the dataset used for both training and testing. Note that the same architecture and training procedure is used in ► [Chap. 22, “Influence of Color Spaces for Deep Learning Image Colorization”](#) of this handbook.

Detailed Architecture

The architecture used in our experiments is an encoder–decoder U-Net composed of five stages. Figure 1 displays a summary of the whole architecture. All convolutional blocks are composed of two 2D convolutional layers with kernels of kernel size equal to 3×3 , each one followed by 2D batch normalization and a ReLU activation. For the encoder, downsampling is done by using a max pooling operator after each convolutional block. After downsampling, the number of filters is doubled in the following block. For the decoder, upsampling is done by using 2D transpose convolutions (with 4×4 kernels with stride 2). At a given stage, the corresponding

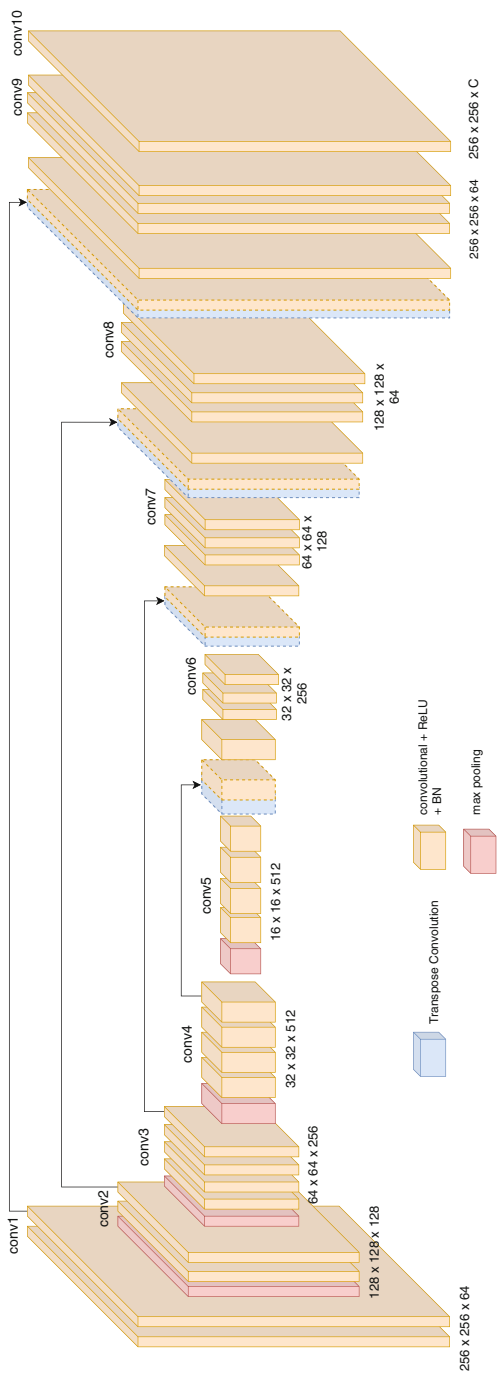


Fig. 1 Summary of the baseline U-Net architecture used in our experiments. It outputs a $256 \times 256 \times C$ image, where C stands for the number of channels, being equal to 2 when estimating the missing chrominance channels and to 3 when estimating the RGB components

Table 2 Detailed architecture and output resolution for each block

Layer type	Output resolution
Input	$3 \times H \times W$
Conv1 + Max-pooling	$64 \times H/2 \times W/2$
Conv2 + Max-pooling	$128 \times H/4 \times W/4$
Conv3 + Max-pooling	$256 \times H/8 \times W/8$
Conv4 + Max-pooling	$512 \times H/16 \times W/16$
Conv5 + Conv. Transpose (I)	$512 \times H/8 \times W/8$
Conv6 + Conv. Transpose (II)	$256 \times H/4 \times W/4$
Conv7 + Conv. Transpose (III)	$128 \times H/2 \times W/2$
Conv8 + Conv. Transpose (IV)	$64 \times H \times W$
Conv9	$64 \times H \times W$
Conv10	$C \times H \times W$

encoder and decoder blocks are linked with skip connections: feature maps from the encoder are concatenated with the ones from the corresponding upsampling path and fused using 1×1 convolutions. More details can be found in Table 2.

The encoder architecture is identical to the CNN part of a VGG network (Simonyan and Zisserman 2015). It allows us to start from pretrained weights initially used for ImageNet classification.

The training settings are described as follows:

- Optimizer: Adam
- Learning rate: $2e-5$.
- Batch size: 16 images (10–11 GB RAM on Nvidia Titan V).
- All images are resized to 256×256 for training which enables using batches. In practice, to keep the aspect ratio, the image is resized such that the smallest dimension matches 256. If the other dimension remains larger than 256, we then apply a random crop to obtain a square image. Note that the random crop is performed using the same seed for all trainings.

More details regarding this framework are given in ► [Chap. 22, “Influence of Color Spaces for Deep Learning Image Colorization”](#).

Quantitative Evaluation Metrics Used in Colorization Methods

For the last 20 years, colorization methods have mostly been evaluated with MAE, MSE, Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) metrics (Wang et al. 2004).

In the context of colorization, the PSNR measures the ratio between the maximum value of a color target image $u : \Omega \rightarrow \mathbb{R}^C$ and the Mean Square Error (MSE) between u and a colorized image $v : \Omega \rightarrow \mathbb{R}^C$ with $\Omega \in \mathbb{Z}^2$ a discrete grid of size $M \times N$. That is

$$\text{PSNR}(u, v) = 20 \log_{10}(\max u) - 10 \log_{10} \left(\frac{1}{CMN} \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^N (u(i, j, k) - v(i, j, k))^2 \right), \quad (18)$$

where $C = 3$ when working in the RGB color space and $C = 2$ in any luminance–chrominance color space as YUV, Lab, and YCbCr. The PSNR score is considered as a reconstruction measure tending to favor methods that will output results as close as possible to the ground truth image in terms of the MSE.

SSIM intends to measure the perceived change in structural information between two images. It combines three measures to compare images, color (l), contrast (c), and structure (s):

$$\text{SSIM}(u, v) = l(u, v)c(u, v)s(u, v) = \frac{(2\mu_u\mu_v) + c_1}{\mu_u^2 + \mu_v^2 + c_1} \frac{(2\sigma_u\sigma_v + c_2)}{\sigma_u^2 + \sigma_v^2 + c_2} \frac{(\sigma_{uv} + c_3)}{\sigma_u\sigma_v + c_3} \quad (19)$$

where μ_u (resp. σ_u) is the mean value (resp. the variance) of image u values and σ_{uv} the covariance of u and v . c_1 , c_2 , and c_3 are regularization constants that are used to stabilize the division for images with mean or standard deviation close to zero.

More recently, other perceptual metrics based on deep learning have been proposed: the Fréchet Inception Distance (FID) (Heusel et al. 2017) and a Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018). They have been widely used in image editing for their ability to correlate well with human perceptual similarity. FID (Heusel et al. 2017) is a quantitative measure used to evaluate the quality of the outputs' generative model and which aims at approximating human perceptual evaluation. It is based on the Fréchet distance (Dowson and Landau 1982) which measures the distance between two multivariate Gaussian distributions. FID is computed between the feature-wise mean and covariance matrices of the features extracted from an Inception v3 neural network applied to the input images (μ_r, Σ_r) and those of the generated images (μ_g, Σ_g):

$$\text{FID}((\mu_r, \Sigma_r), (\mu_g, \Sigma_g)) = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\Sigma_r\Sigma_g)^{1/2}. \quad (20)$$

LPIPS (Zhang et al. 2018) computes a weighted L2 distance between deep features of a pair of images u and v :

$$\text{LPIPS}(u, v) = \sum_l \frac{1}{H_l W_l} \sum_{i=1}^{H_l} \sum_{j=1}^{W_l} \|\omega_l \odot (\Phi_l(u)_{i,j} - \Phi_l(v)_{i,j})\|_2^2, \quad (21)$$

where H_l (resp. W_l) is the height (resp. the width) of feature map Φ_l at layer l and ω_l are weights for each features. Note that features are unit-normalized in the channel dimension.

Other quantitative metrics can be found in the literature for image colorization. Accuracy (Nazeri et al. 2018) measures the ratio between the number of pixels that have the same color information as the source and the total number of pixels. Raw accuracy (AuC) (Zhang et al. 2016) computes the percentage of predicted pixel colors within a threshold of the L2 distance from the ground truth in ab color space. The result is then swept across thresholds from 0 to 150 to produce a cumulative mass function. Deshpande et al. (2017) evaluate colorfulness as the MSE on histograms. Royer et al. (2017) verify if the framework produces vivid colors by computing the average perceptual saturation (Lübbe 2010). Other works evaluate the capability of a classification network to infer the right class to the generated image (Zhang et al. 2016; He et al. 2018). Zhang et al. (2016) feed the generated image to a classification network and observe if the classifier performs well.

Table 3 Evaluation metrics used by deep learning methods for image colorization

	Quantitative							User study		
	L1/MAE	L2/MSE	PSNR	SSIM	LPIPS	FID	Other	AMT Fooling Rate	Naturalness	Other
Cheng et al. (2015)				•						
Iizuka et al. (2016)							•		•	
Using GANS										
Vitoria et al. (2020)			•						•	
Nazeri et al. (2018)	•						•			
Cao et al. (2017)		•	•							•
Yoo et al. (2019)					•					•
Histograms Prediction										
Larsson et al. (2016)		•	•				•			
Zhang et al. (2016)							•	•		
User Guided										
Zhang et al. (2017)			•					•		
He et al. (2018)			•				•	•		
Diverse										
Deshpande et al. (2017)		•								•
Guadarrama et al. (2017)				•				•		
Royer et al. (2017)										•
Kumar et al. (2021)						•		•		
Object Aware										
Su et al. (2020)			•	•	•					
Pucci et al. (2021)			•		•					
Kong et al. (2021)			•	•			•			
Survey										
Gu et al. (2019)			•	•						•

Note that all models that are trained with a L2 loss will more likely get better PSNR or MSE as the L2 loss is correlated with the evaluation.

Table 3 summarizes the quantitative evaluation metrics more generally used in the literature of image colorization. In our experiments, we choose to rely on the more generally used and more recent ones, namely, L1 (MAE), L2 (MSE), PSNR, SSIM, LPIPS, and FID.

Experimental Analysis

To compare the influence of the objective loss in the resulting colorization results, we train the network described in Section “[Proposed Colorization Framework](#)” by changing the objective loss. In particular, we train the network with the L1 loss, the L2 loss, the VGG-based LPIPS, the combination of WGAN plus L2 losses, and the combination of WGAN and VGG-based LPIPS. To the best of our knowledge, the combination of the VGG-based LPIPS loss with a WGAN training procedure is novel and has not been proposed in the recent literature.

For each of these losses, depending on the chosen color space, we estimate:

- either the two (a, b) chrominance channels given the luminance channel L as input;
- or the three (R, G, B) color channels given a grayscale image as input.

In this section, we present a quantitative and qualitative comparison for all of these combinations. Note that to compute the VGG-based LPIPS loss, the output colorization always has to be converted to RGB (in a differentiable way), even for Lab color space, because this loss is computed with a pre-trained VGG expecting RGB images as input. To this end, we have used the Kornia implementation of differentiable color space conversions (Riba et al. 2020).

Throughout our experiments, we use the COCO dataset (Lin et al. 2014), containing various natural images of different sizes. COCO is divided into three sets that approximately contain 118k, 5k, and 40k images that, respectively, correspond to the training, validation, and test sets. Note that we carefully remove all grayscale images, which represent around 3% of the overall amount of each set. Although larger datasets such as ImageNet have been regularly used in the literature, COCO offers a sufficient number and a good variety of images so we can efficiently train and compare numerous models. While the training is done on batches of square 256×256 images, for testing, we apply the network to images at their original resolution.

Quantitative Evaluation

Table 4 shows the quantitative results comparing five losses, namely, the L1 loss, the L2 loss, the VGG-based LPIPS, the combination of WGAN plus L2 losses, and

Table 4 Quantitative evaluation of colorization results for different loss functions. Metrics are used to compare ground truth to every images in the 40k test set. Best and second best results by column are in bold and italics, respectively

Color space	Loss function	MAE ↓	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Lab	L1	0.04407	0.00589	22.3020	0.9268	0.1587	8.8109
Lab	L2	0.04488	0.00585	22.3283	<i>0.9250</i>	0.1613	8.1517
Lab	LPIPS	0.04374	0.00566	22.4699	0.9228	0.1403	<i>3.2221</i>
Lab	WGAN+L2	0.04459	0.00582	22.3512	0.9243	0.1609	7.6127
Lab	WGAN+LPIPS	<i>0.04383</i>	<i>0.00568</i>	<i>22.4541</i>	0.9223	<i>0.1406</i>	3.1045
RGB	L1	0.04385	0.00587	22.3119	0.9268	0.1583	8.0125
RGB	L2	<i>0.04458</i>	<i>0.00587</i>	<i>22.3136</i>	<i>0.9255</i>	0.1606	7.4223
RGB	LPIPS	0.04573	0.00577	22.3892	0.9196	0.1429	<i>3.0576</i>
RGB	WGAN+L2	0.05256	0.00651	21.8667	0.8559	0.2469	15.4780
RGB	WGAN+LPIPS	0.04901	0.00679	21.6806	0.9137	<i>0.1495</i>	2.6719

the combination of WGAN and VGG-based LPIPS (denoted in Table 4 as L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS, respectively). The first five rows display this assessment when the used color space is Lab (i.e., the model estimates the two ab chrominance channels), while for the last five rows, the used color space is RGB (i.e., the model estimates the three RGB color channels). In particular, let us remark that the quantitative evaluations are always performed in the final RGB color space. Thus, even when the model is trained to estimate the ab chrominance channels, the resulting Lab color image is converted to the RGB color space to compute the evaluation metrics.

From the results in Table 4, we observe that for the analyzed dataset, the models trained with the VGG-based LPIPS loss function provide overall better quantitative results, for both Lab and RGB color spaces. This is especially true for the perceptual metrics LPIPS and FID, as they are strongly correlated to this loss function. The fact that the VGG-based LPIPS training loss is computed on RGB color space (as this loss is computed with a pre-trained VGG expecting RGB images as input) and also is a quantitative result might be related to the performance (see also ► [Chap. 22 “Influence of Color Spaces for Deep Learning Image Colorization”](#)). In the same spirit, we can observe a slight correlation between the used training loss and the quantitative metric. For instance, when training with L1, MAE results are better. However, we can see that L2 loss is not at the top in any of the metrics, while we could have expected in the case of MSE or PSNR, but this is not the case.

Nevertheless, no strong tendency clearly emerges from this table: for many metrics, the different losses do not differ so much from one another and could be in the margin of error. From our analysis, we hypothesize that, apart from the chosen objective function, the network architecture design, and the training process, may play a very important role as a prior on the colorization operator. Further analysis will be done on that matter.

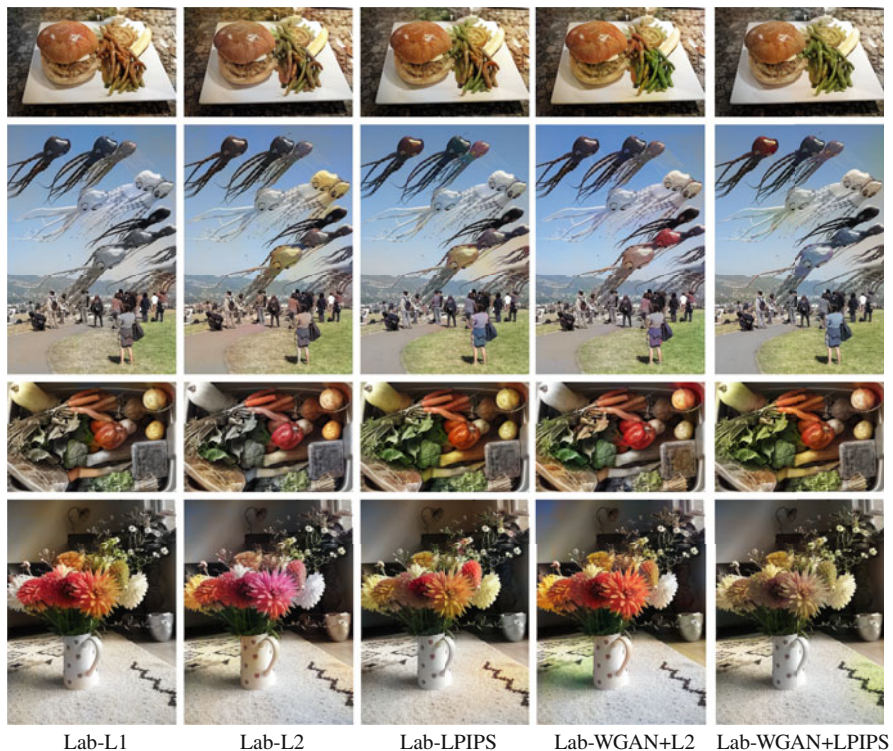


Fig. 2 Examples where multiple objects are in the same image. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS. The used color space is Lab for all the cases (i.e., the model estimates two ab chrominance channels)

Finally, let us mention the importance of user-based quantitative studies to properly assess colorizations results. It is not just important in cases where no ground truth is available, such as the ones of old archive photographs, but also due to the fact that multiple colorizations are always possible. Several works propose different user-based metrics, e.g., *naturalness* or *fooling rate*. Nevertheless, efforts should be made on a widely accepted protocol and a widespread user study metric.

Qualitative Evaluation

Figures 2, 3, and 4 show a qualitative experimental comparison of the five losses, namely, L1, L2, VGG-based LPIPS, WGAN+L2, and WGAN+VGG-based LPIPS. In all cases, the models were trained on the Lab color space. Still, we recall that any model based on VGG-based LPIPS loss requires to convert the predicted image to the RGB color space in a differentiable way (i.e., with Kornia (Riba et al. 2020)).

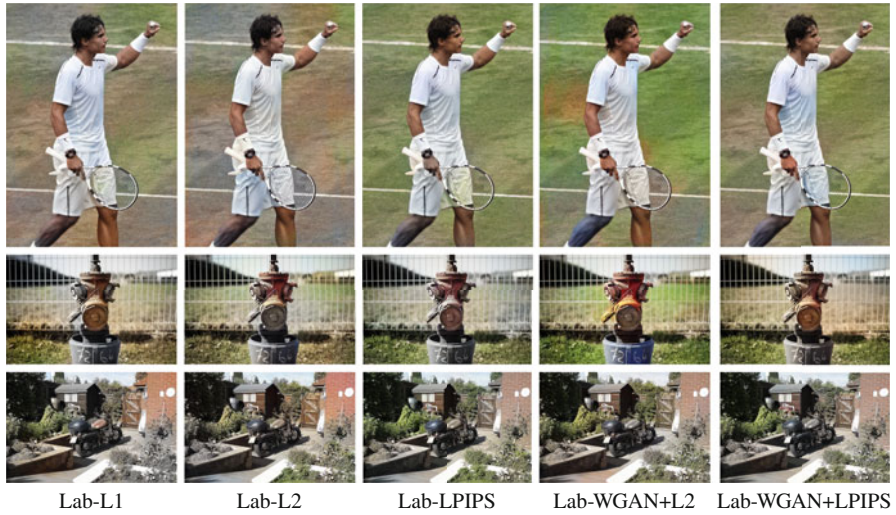


Fig. 3 Examples to evaluate shininess of the results. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS. The used color space is Lab for all the cases

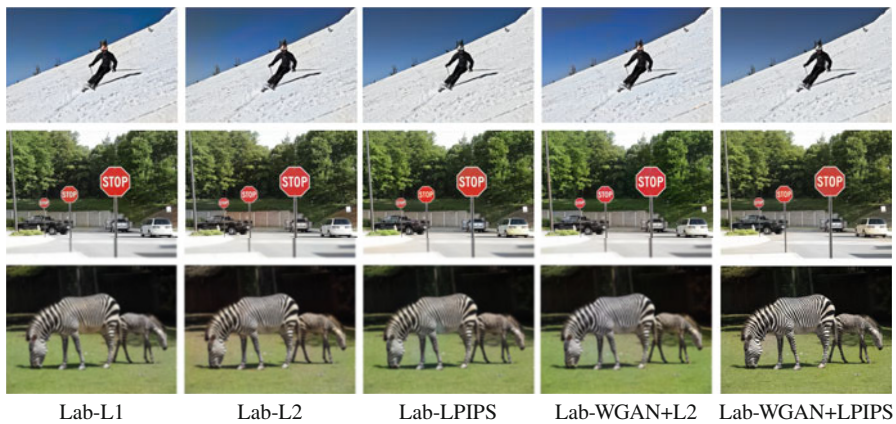


Fig. 4 Colorization results on images that contain objects have strong structures and that have been seen many times in the training set. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS. The used color space is Lab for all the cases

In Fig. 2, we can see some results obtained for each of the studied losses in images with multiple objects. We can observe that each loss brings slightly different colors to objects. Overall, VGG-based LPIPS and WGAN losses generate shinier and more colorful images (it can be seen, for instance, in the sky, grass, and vegetables), although we can observe colorful examples in the case of the L2 loss in the example of the flowers or vegetables. However, WGAN hallucinates more unrealistic colors as can be seen on the table or the wall on the image with a flower



Fig. 5 Examples where multiple objects are in the same image. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS perceptual. The used color space is RGB for all the cases (i.e., the model estimates three RGB color channels)

of the last row of Fig. 2. This effect can be reduced by improving architecture and semantic features (e.g., Vitoria et al. 2020) or by introducing spatial localization (e.g., Su et al. 2020). Besides, by comparing the two last columns obtained with the models trained with the adversarial strategy WGAN combined with, respectively, the L2 or the VGG-based LPIPS, one can observe that WGAN+VGG-based LPIPS tends to homogenize colors (e.g., some of the balloons take similar color to the sky on the second row; the flowers on the fifth have grayish colors, more similar to the wall). WGAN+VGG-based LPIPS also tends to have less bleeding than WGAN+L2.

The generation of more vivid colors with VGG-based LPIPS and WGAN losses is also visible on Fig. 3. The grass and bushes are more green and look more natural. However, none of the losses give consistency to all the limbs of the tennis player on the first row (e.g., the right leg).

Figure 4 shows results on objects, here zebra and stop sign, with strong contours that were highly present in the training set. The colorization of this object is impressive for any loss. None of the losses manage to properly colorize the person

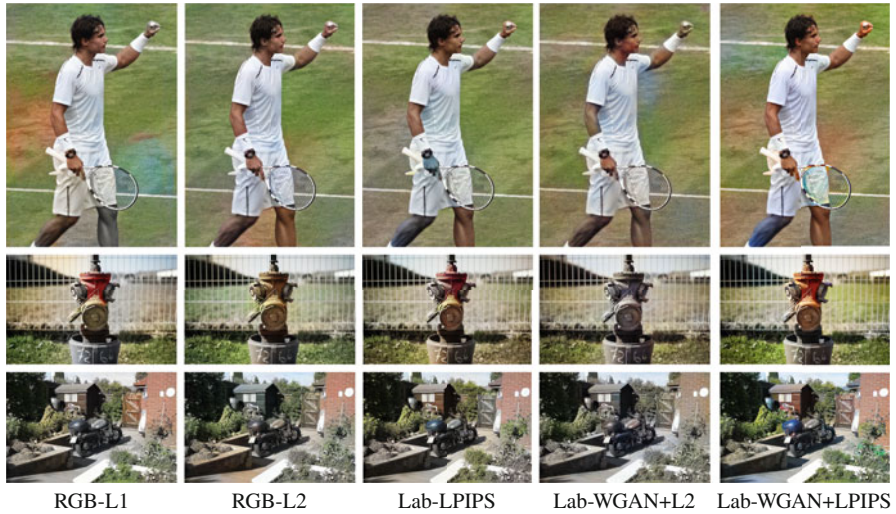


Fig. 6 Examples to evaluate shininess of the results. Five losses are compared, namely, L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS perceptual. The used color space is RGB for all the cases

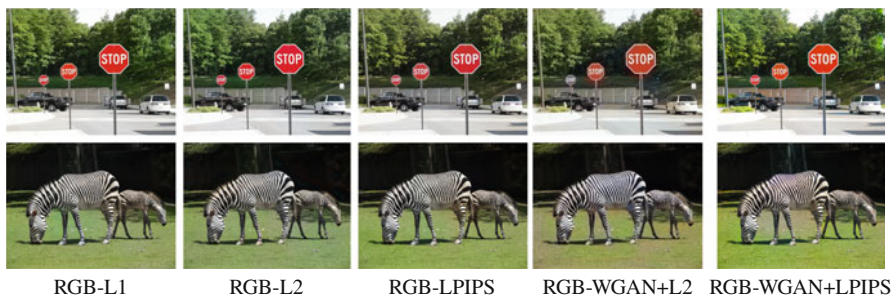


Fig. 7 Colorization results on images that contain objects which have strong structures and that have been seen many times in the training set. Four losses are compared, namely, L1, L2, LPIPS, and WGAN+L2. The used color space is RGB for all the cases

near the center car on the first row. This type of examples could be improved by learning high-level semantics on the image content.

Figures 5, 6, and 7 show an additional experimental comparison of five losses, namely, L1, L2, VGG-based LPIPS, WGAN+L2, and WGAN+VGG-based LPIPS, but when the network is trained to learn the three RGB color channels for all the cases. For these test images, more realistic and consistent results are obtained in general for this configuration. Let us notice from the results in these three figures that more colorful images are obtained compared to the ones of Figs. 2, 3, and 4, although less textured. Further analysis on the influence of the chosen color space can be found in ► [Chap. 22, “Influence of Color Spaces for Deep Learning Image Colorization”](#).

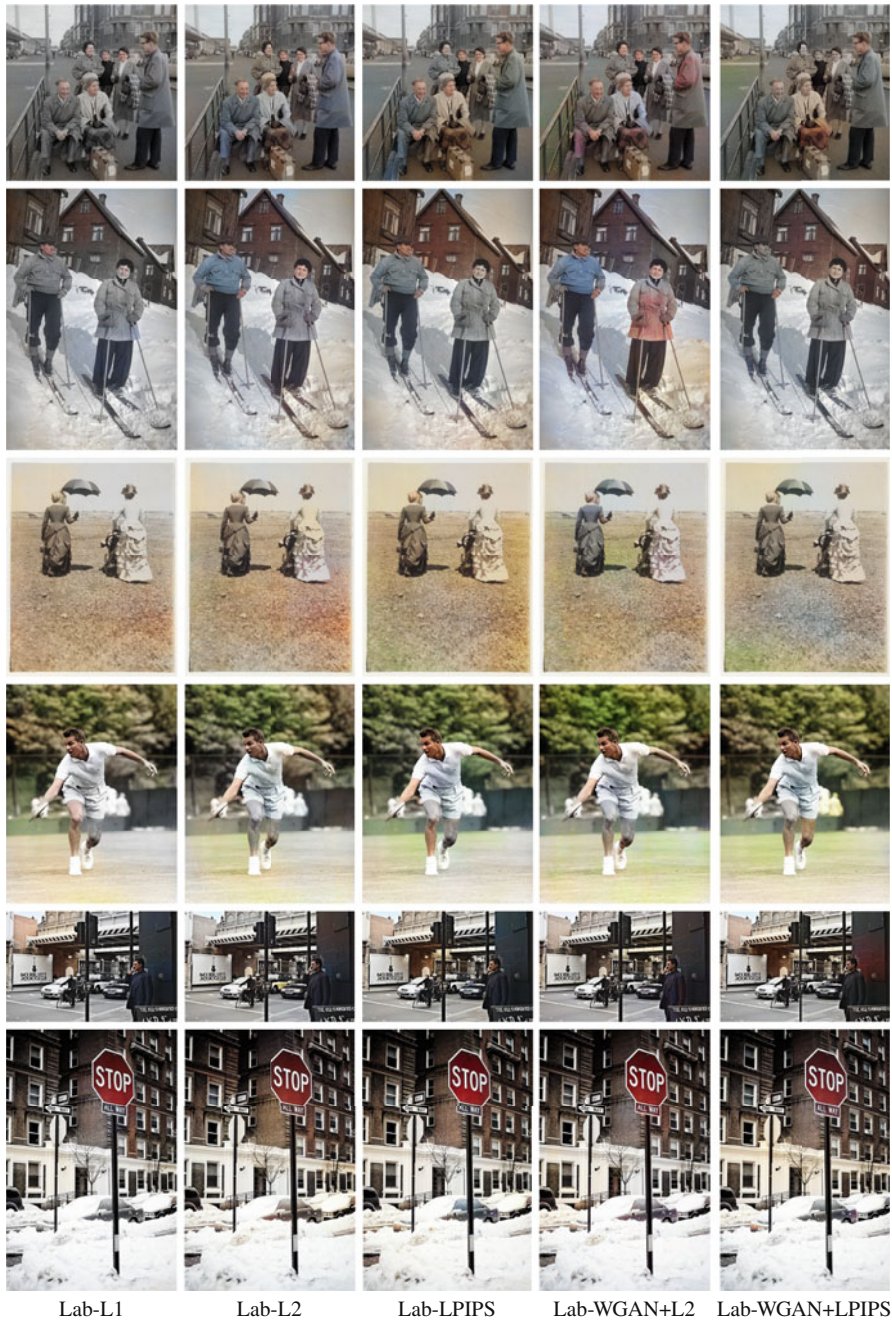


Fig. 8 Examples in original black and white Images. These colorization results have been obtained using the five networks trained, respectively, with L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS losses, and learning the two ab chrominance channels

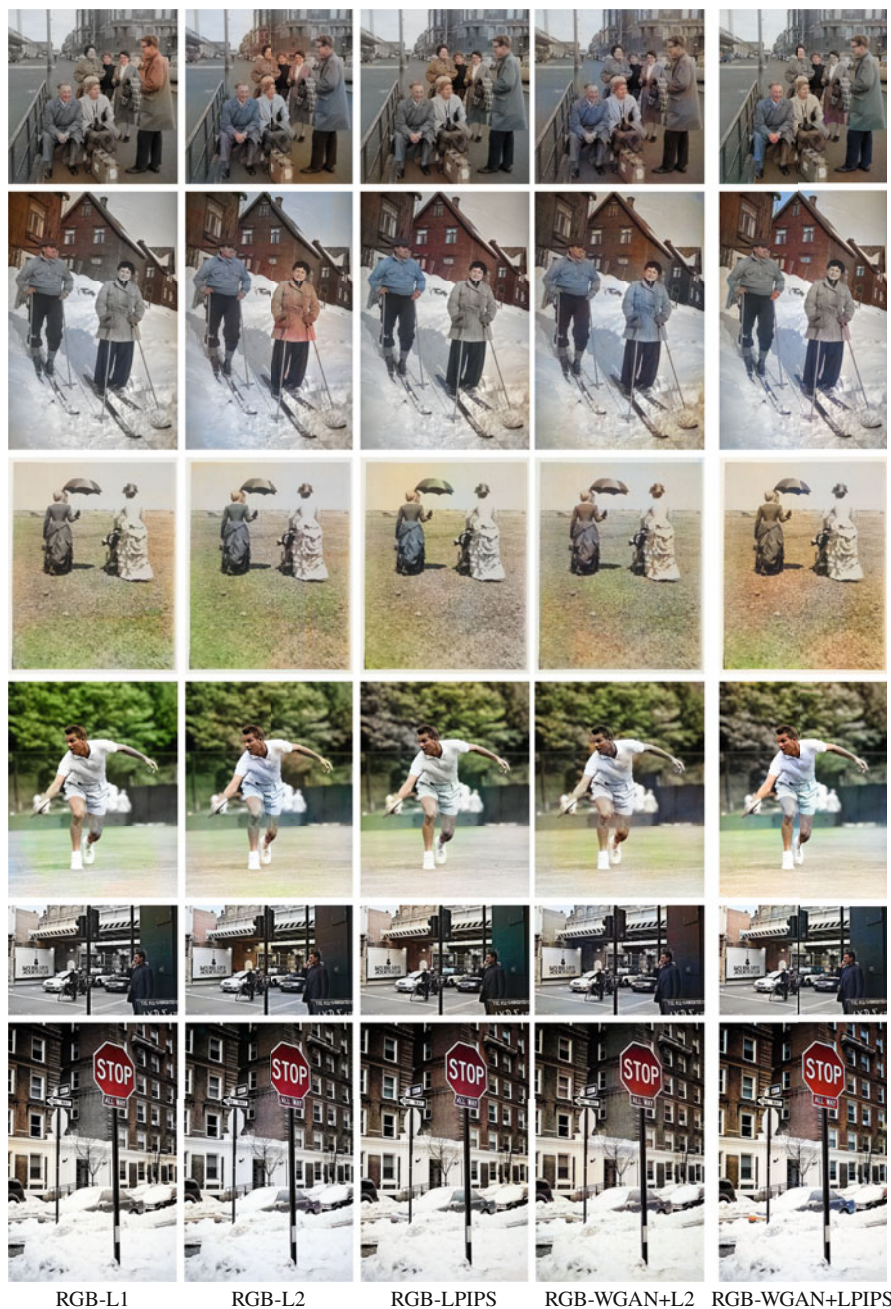


Fig. 9 Examples in original black and white Images. These colorization results have been obtained using the five networks trained, respectively, with L1, L2, LPIPS, WGAN+L2, and WGAN+LPIPS losses, and learning the three RGB color channels

Generalization to Archive Images

Finally, in Figs. 8 and 9, we can see additional colorization results on real black and white images from the Pascal VOC dataset. Those results have been obtained using the network trained with the five different losses, respectively, with L1, L2, VGG-based LPIPS, WGAN+L2, and WGAN+VGG-based LPIPS. For Fig. 8, only the two ab chrominance channels are learned, while in Fig. 9, the three RGB color channels are learned. Again, none of the losses manage to consistently colorize the skin of all the people of the image at the first, second, and fourth rows of Fig. 8, although possibly it is slightly better when using perceptual and GAN losses. Notice that also in these cases, the colors are slightly more vivid, specially visible in the first two rows of Fig. 8. However, color inconsistency and failures in spatial localization appear, more visible in the first four rows. As mentioned, this effect can be reduced by introducing semantic information (e.g., Vitoria et al. 2020) or spatial localization (e.g., Su et al. 2020).

Conclusion

In this chapter, we have studied the role of loss functions on automatic colorization with deep learning methods. Using a fixed standard network, we have shown that the choice of the right loss does not seem to play a crucial role in the colorization results. We therefore argue that most efforts should be made on the influence of the architecture design, as it is related to the type of colorization operator one can expect to obtain. Indeed, in our analysis, we used a U-Net-based architecture which has shown to have a strong impact on the experimental results. For the employed architecture, the models including the VGG-based LPIPS loss function provide overall slightly better results, especially for the perceptual metrics LPIPS and FID. Likewise, the role of both architectures and losses for obtaining a real diversity of colorization results could be explored in future works.

Acknowledgments This study has been carried out with financial support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01) and from the EU Horizon 2020 research and innovation programme NoMADS (Marie Skłodowska-Curie grant agreement No 777826). The first and fourth authors acknowledge partial support by MICINN/FEDER UE project, ref. PGC2018-098625-B-I00, and RED2018-102511-T. This chapter was written together with another chapter of the current handbook, called ► [Chap. 22, “Influence of Color Spaces for Deep Learning Image Colorization”](#). All authors have contributed to both chapters.

References

- Antic, J.: Deoldify. <https://github.com/jantic/DeOldify> (2019)
- Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein Generative Adversarial Networks. In: International Conference on Machine Learning, vol 70, pp. 214–223 (2017)

- Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via Generative Adversarial Networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 151–166 (2017)
- Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: Pixelsnail: an improved autoregressive generative model. In: International Conference on Machine Learning, pp. 864–872 (2018)
- Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: IEEE International Conference on Computer Vision, pp. 415–423 (2015)
- Deshpande, A., Lu, J., Yeh, M.-C., Jin Chong, M., Forsyth, D.: Learning diverse image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6837–6845 (2017)
- Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Comparison of full-reference image quality models for optimization of image processing systems. *Int. J. Comput. Vis.* **129**(4), 1258–1281 (2021)
- Dowson, D., Landau, B.: The Fréchet distance between multivariate normal distributions. *J. Multivar. Anal.* **12**(3), 450–455 (1982)
- Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. *J. Vis.* **16**(12), 326 (2016)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
- Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., Murphy, K.: Pixcolor: pixel recursive colorization. In: British Machine Vision Conference (2017)
- Gu, S., Timofte, R., Zhang, R.: Ntire 2019 challenge on image colorization: report. In: Conference on Computer Vision and Pattern Recognition Workshops (2019)
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems, pp. 5769–5779 (2017)
- He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. *ACM Trans. Graph.* **37**(4), 1–16 (2018)
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
- Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers (2019). arXiv preprint arXiv:1912.12180
- Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.* **35**(4), 1–11 (2016)
- Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
- Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711 (2016)
- Kong, G., Tian, H., Duan, X., Long, H.: Adversarial edge-aware image colorization with semantic segmentation. *IEEE Access* **9**, 28194–28203 (2021)
- Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer (2021). arXiv preprint arXiv:2102.04432
- Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European Conference on Computer Vision, pp. 577–593 (2016)
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014)
- Lübbe, E.: Colours in the Mind-Colour Systems in Reality: A Formula for Colour Saturation. BoD-Books on Demand, Norderstedt (2010)
- Mouzon, T., Pierre, F., Berger, M.-O.: Joint CNN and variational model for fully-automatic image colorization. In: Scale Space and Variational Methods in Computer Vision, pp. 535–546 (2019)

- Nazeri, K., Ng, E., Ebrahimi, M.: Image colorization using Generative Adversarial Networks. In: International Conference on Articulated Motion and Deformable Objects, pp. 85–94 (2018)
- Oord, A.V.D., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with PixelCNN decoders. In: Advances in Neural Information Processing Systems (2016)
- Pierre, F., Aujol, J.-F.: Recent approaches for image colorization. In: Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging (2020)
- Pierre, F., Aujol, J.-F., Bugeau, A., Papadakis, N., Ta, V.-T.: Luminance-chrominance model for image colorization. *SIAM J. Imag. Sci.* **8**(1), 536–563 (2015)
- Pucci, R., Micheloni, C., Martinel, N.: Collaborative image and object level features for image colourisation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2160–2169 (2021)
- Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for PyTorch. In: Winter Conference on Applications of Computer Vision, pp. 3674–3683 (2020)
- Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization. In: British Machine Vision Conference (2017)
- Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
- Su, J.-W., Chu, H.-K., Huang, J.-B.: Instance-aware image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7968–7977 (2020)
- Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning, pp. 1747–1756 (2016)
- Vitoria, P., Raad, L., Ballester, C.: ChromaGAN: adversarial picture colorization with semantic class distribution. In: Winter Conference on Applications of Computer Vision, pp. 2445–2454 (2020)
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
- Yoo, S., Bahng, H., Chung, S., Lee, J., Chang, J., Choo, J.: Coloring with limited data: few-shot colorization via memory augmented networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
- Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision, pp. 649–666 (2016)
- Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.* **36**, 1–11 (2017)
- Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)



Influence of Color Spaces for Deep Learning Image Colorization 22

Aurélie Bugeau, Rémi Giraud, and Lara Raad

Contents

Introduction	848
Related Work	849
On Color Spaces	849
Review of Colorization Methods	851
Datasets Used in Literature	859
Proposed Colorization Framework	861
Detailed Architecture	861
Training and Testing Images	862
Learning Strategy for Different Color Spaces	863
Analysis of the Influence of Color Spaces	863
Quantitative Evaluation	866
Qualitative Evaluation	868
Generalization to Archive Images	872
Conclusion	874
References	875

A. Bugeau (✉)

LaBRI, CNRS, UMR5800, Univ. Bordeaux, F-33400 Talence, France

Institut universitaire de France (IUF), Paris, France

e-mail: aurelie.bugeau@labri.fr

R. Giraud

Univ. Bordeaux, CNRS, IMS UMR5251, Bordeaux INP, F-33400 Talence, France

e-mail: remi.giraud@ims-bordeaux.fr

L. Raad

LIGM, CNRS, Univ Gustave Eiffel, F-77454 Marne-la-Vallée, France

e-mail: lara.raadcisa@esiee.fr

Abstract

Colorization is a process that converts a grayscale image into a colored one that looks as natural as possible. Over the years this task has received a lot of attention. Existing colorization methods rely on different color spaces: RGB, YUV, Lab, etc. In this chapter, we aim to study their influence on the results obtained by training a deep neural network, to answer the following question: “Is it crucial to correctly choose the right color space in deep learning-based colorization?” First, we briefly summarize the literature and, in particular, deep learning-based methods. We then compare the results obtained with the same deep neural network architecture with RGB, YUV, and Lab color spaces. Qualitative and quantitative analysis do not conclude similarly on which color space is better. We then show the importance of carefully designing the architecture and evaluation protocols depending on the types of images that are being processed and their specificities: strong/small contours, few/many objects, recent/archive images.

Keywords

Image colorization · Deep learning · Color spaces

Introduction

Image colorization consists in recovering a colored image from a grayscale one. This process attracts a lot of attention in the image-editing community in order to restore or colorize old grayscale movies or pictures. While turning a colored image into a grayscale one is only a matter of standard, the reverse operation is a strongly ill-posed problem as no information on which color has to be added is known. Therefore priors must be considered. In the literature, there exist three kinds of priors leading to different types of colorization methods. In the first category, initiated by Levin et al. (2004), the user manually adds initial colors through scribbles to the grayscale image. The colorization process is then performed by propagating the input color data to the whole image. The second category, called automatic or patch-based colorization, initiated by Welsh et al. (2002), consists in transferring color from one (or many) initial colored image considered as an example. The last category, which attracts most research nowadays, concerns deep learning approaches. The necessary color prior here is learned from large datasets.

Generally, in colorization methods, the initial grayscale image is considered as the luminance channel which is not modified during the colorization. The objective is then to reconstruct the two chrominance channels, before turning back to the RGB color space. Different luminance-chrominance spaces exist and have been used for image colorization. One common problem with all image colorization methods that aim at reconstructing the chrominances of the target image is that the recovered chrominances combined with the input luminance may not fall into the RGB cube when converting back to the RGB color space. Therefore, some works have decided

to work directly on RGB to cope with this limitation by constraining the luminance channel (Pierre et al. 2014).

The objective of this chapter is to analyze the influence of color spaces on the results of automatic deep learning methods for image colorization. This chapter comes together with another chapter of this handbook. This other chapter, ► [Chap. 21, “Analysis of Different Losses for Deep Learning Image Colorization”](#), focuses on the influence of losses. We refer the reader to it for a review of the traditionally used different losses and evaluation metrics. Here, after reviewing existing works in image colorization and, in particular, works based on deep learning, we will focus on the influence of color spaces. Based on our analysis of the literature, a baseline architecture is defined and later used in all comparisons. Additionally, again based on the literature review, we set a uniform training procedure to ensure fair comparisons. Experiments encompass qualitative and quantitative analysis.

The chapter is organized as follows. Section “[Related Work](#)” first recalls some basics on color spaces and then provides a detailed survey of the literature on colorization methods and finally lists the datasets traditionally used. Next, in section “[Proposed Colorization Framework](#)”, we present the chosen architecture and in section “[Learning Strategy for Different Color Spaces](#)” the learning strategy. Section “[Analysis of the Influence of Color Spaces](#)” presents the results of the different experiments. A discussion on the generalization of this work to archive images is later provided in section “[Generalization to Archive Images](#)” before a conclusion is drawn.

Related Work

On Color Spaces

This section presents the different color spaces that have been used for colorization in the literature. For more information about color theory and color constancy (i.e., the underlying ability of human vision to perceive colors very robustly with respect to changes of illumination), see, for instance, Ebner (2007) and Fairchild (2013).

Colored images are traditionally saved in the RGB color space. A grayscale image contains only one channel that encodes the luminosity (perceived brightness of that object by a human observer) or the luminance (absolute amount of light emitted by an object per unit area). A way to model this luminance Y which is close to the human perception of luminance is:

$$Y = 0.299R + 0.587G + 0.114B, \quad (1)$$

where R , G , and B are, respectively, the amount of light emitted by an object per unit area in the low, medium, and high frequency bands that are visible by a human eye. Colorization aims to retrieve color information from a grayscale image. To do so, and to easily constrain the luminance channel, most methods propose to work in a luminance-chrominance space. The problem becomes the

retrieval of two chrominance channels given the luminance Y . There exist several luminance-chrominance spaces. Two of them are mostly used for colorization. The first one, YUV, historically used for a specific analog encoding of color information in television systems, is the result of the linear transformation:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51498 & -0.10001 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

The reverse conversion from YUV and RGB is simply obtained by inverting the matrix. The other linear space that has been used for colorization is YCbCr.

The CIELAB color space, also referred to as Lab or La*b*, defined by the International Commission on Illumination (CIE) in 1976, is also frequently used for colorization. It has been designed such that the distances between colors in this space correspond to the perceptual distances of colors for a human observer. The three channels become uncorrelated. The transformation from RGB to Lab (and the reverse) is nonlinear. First, it is necessary to convert the RGB values to the CIEXYZ color space:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.769 & 1.7518 & 0.13 \\ 1 & 4.5907 & 0.0601 \\ 0 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Then, the transformation to Lab is given by:

$$\begin{aligned} L &= 116f(Y/Y_n) - 16, \\ a &= 500[f(X/X_n) - f(Y/Y_n)], \\ b &= 200[f(Y/Y_n) - f(Z/Z_n)], \end{aligned}$$

with

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3, \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{otherwise,} \end{cases}$$

where X_n , Y_n , and Z_n describe a specified white achromatic reference illuminant. Obviously, the reverse operation from Lab to RGB is also nonlinear.

Despite RGB or luminance-chrominance color spaces, few methods relying on hue-based spaces have been proposed for colorization. For instance, Larsson et al. (2016) rely on a hue-chroma-luminance space.

Table 1 lists the color spaces used in deep learning colorization methods described in the next subsection. It distinctly appears that the Lab color space is the most widely used. We will further discuss this choice in section “[Analysis of the Influence of Color Spaces](#)”.

Table 1 Color spaces used in deep learning methods for image colorization

		Using GANs					Histogram prediction			User guided		Diverse			Object aware			Survey			
	Cheng et al. (2015)	Iizuka et al. (2016)	Vitoria et al. (2020)	Nazeri et al. (2018)	Cao et al. (2017)	Yoo et al. (2019)	Antic (2019)	Larsson et al. (2016)	Zhang et al. (2016)	Mouzon et al. (2019)	Zhang et al. (2017)	He et al. (2018)	Deshpande et al. (2017)	Guadarrama et al. (2017)	Royer et al. (2017)	Kumar et al. (2021)	Su et al. (2020)	Pucci et al. (2021)	Kong et al. (2021)	Gu et al. (2019) (winner)	
RGB					•	•	•									•				•	
YUV	•					•															
YCbCr														•							
Lab		•	•	•		•		•	•	•	•	•	•		•		•	•	•		
hue/ chroma								•													
Comparison	•				•			•													

In general terms, as can be seen in Table 1, most methods work in a luminance-chrominance space, and the cost functions to optimize are in general defined in the same space. Hence, converting from and to RGB to one of these luminance/chrominance spaces is not involved in the backpropagation step. Once the training is performed, at inference time the chrominance values given by the network together with the luminance component are converted back to the RGB color space. As mentioned earlier, this operation tends to perform an abrupt value clipping to fit in the RGB cube hence modifying both the original luminance values and the predicted chrominance values. Two libraries are most commonly used for the conversion step: the color module of scikit-image (Zhang et al. 2016, 2017; Larsson et al. 2016; Royer et al. 2017) and the color space conversion functions of OpenCV (Iizuka et al. 2016; Vitoria et al. 2020).

Review of Colorization Methods

This section presents an overview of the colorization methods in the three categories: scribble-based, exemplar-based, and deep learning. For a more detailed review with the same classification, we refer the reader to the recent review Li et al. (2020). Another survey focused on deep learning approaches proposes a taxonomy to separate these methods into seven categories (Anwar et al. 2020). The authors of this review have redrawn all networks architectures, thus allowing to easily compare architecture specificity. Comparisons of methods are made on a new Natural-Color Dataset made of objects with white background.

The NTIRE challenge is a competition for different computer vision tasks related to image enhancement and restoration. One of the tasks in 2019 was image colorization (Gu et al. 2019), with two tracks: colorization without or with guidance given by a second input that provides several color guiding points.

Scribble-Based Image Colorization

The first category of colorization methods relies on color priors coming from scribbles drawn by the user (see Fig. 1). These colors are propagated to all pixels by diffusion schemes.

The first manual colorization method based on scribbles was proposed by Levin et al. (2004). It solves an optimization problem to diffuse the chrominances of scribbles with the assumption that chrominances should have small variations where the luminance has small variations. To reduce the number of needed scribbles, Luan et al. (2007) first use scribbles to segment the image before diffusing the colors. Yatziv and Sapiro (2006) propose a simple yet fast method by using geodesic distances to blend the chrominances given by the scribbles. In Huang et al. (2005), edge information is extracted to reduce color bleeding. Heu et al. (2009) use pixel priorities to ensure that important areas end up with the right colors. Other propagation schemes include probabilistic distance transform (Lagodzinski and Smolka 2008), discriminative textural features (Kawulok et al. 2012), structure tensors (Drew and Finlayson 2011), nonlocal graph regularization (Lézoray et al. 2008), matrix completion (Wang and Zhang 2012; Yao and James 2015) or rank minimization (Ling et al. 2015). As often described in the literature, with these manual approaches, the contours are not well preserved. To cope with this issue, in Ding et al. (2012), scribbles are automatically generated after segmenting the image and the user only needs to provide one color per scribble. However, all manual methods suffer from the following drawback: if the target represents a complex scene, the user interaction becomes very important. On the other hand,

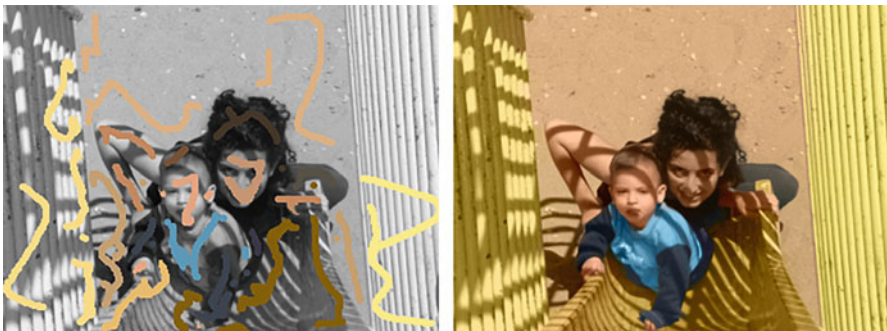


Fig. 1 Example of scribble-based image colorization taken from Levin et al. (2004). The user draws color that are successively diffused to neighbor pixels according under some constraints that depend on the different methods

these approaches propose a global optimization over the image, thus leading to spatial consistency in the result.

Exemplar-Based Image Colorization

The second category of colorization methods concerns exemplar-based methods which rely on a color reference image as prior. The first exemplar-based colorization method was proposed by Welsh et al. (2002). It makes the assumption that pixels with similar intensities or similar neighborhood should have similar colors. It extends the texture synthesis approach by Efros and Leung (1999): the final color of one pixel is copied from the most similar pixel in a reference input colored image. The similarity between pixels relies on patch-based metrics (see Fig. 2). This approach has given rise to many extension in the literature (Di Blasi and Reforgiato 2003; Liu and Zhang 2012). In particular, many works have focused on choosing or designing appropriate features for matching pixels (Chia et al. 2011; Gupta et al. 2012; Bugeau and Ta 2012; Cheng et al. 2015; Arbelot et al. 2016, 2017).

To overcome the spatial consistency and coupling problems in automatic methods, several works rely on image segmentation. For instance, Irony et al. (2005) propose to determine the best matches between the target pixels and regions in a pre-segmented source image. With these correspondences, micro-scribbles from the source are initialized on the target image and colors are propagated as in Levin et al. (2004). Tai et al. (2005) build a probabilistic segmentation of both images where one pixel can belong to many regions. They use it to transfer color between any two regions having similar statistics with an expectation-maximization scheme.

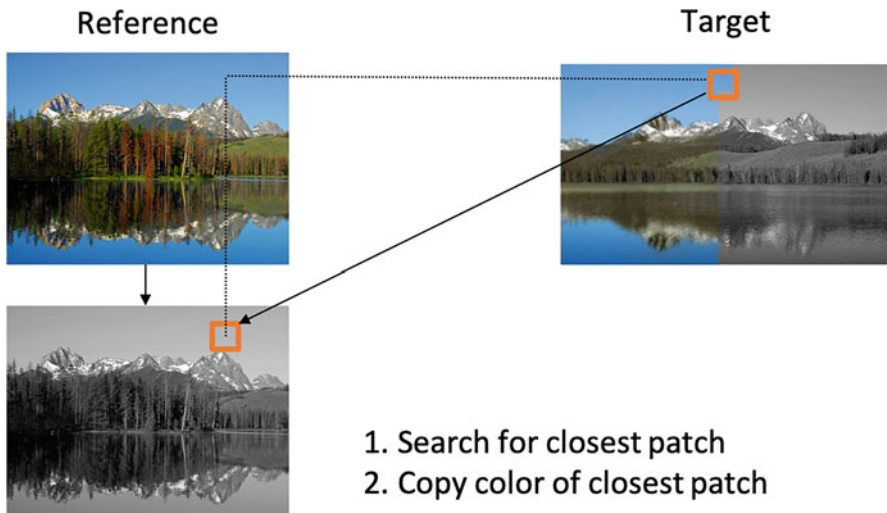


Fig. 2 Principle of exemplar-based image colorization. Methods in this category have proposed different similar patch search strategies and techniques to add spatial consistency when copying patch colors

Gupta et al. (2012) extract different features from the superpixels (Ren and Malik 2003) of the target image and match them with the source ones. The final colors are computed by imposing spatial consistency as in Levin et al. (2004). Li et al. (2017b) extract low- and high-level features on superpixels of the reference to form a dictionary then used as a dictionary-based sparse reconstruction problem. Sparse representation was previously used for colorization in Pang et al. (2013) where images are segmented from scribbles. These approaches incorporate local consistency into automatic methods via segmentation. In Charpiat et al. (2008), spatial consistency is solved with graph cuts after estimating for each pixel the conditional probability of colors. In Bugeau et al. (2014) and Pierre et al. (2014) each pixel can only take its chrominance (or RGB color) among a reduced set of possible candidates chosen from the reference image. The final color is chosen using a variational formulation. In the same trend, Fang et al. (2019) propose a superpixel-based variational model. In Li et al. (2017a), the distribution of intensity deviation for uniform and nonuniform regions is learned and used in a Markov random field (MRF) model for improved consistency. Finally, Li et al. (2019) propose cross-scale local texture matching, which are then fused using global graph-cut optimization.

A major problem of this family of methods is the high dependency on the reference image. Chia et al. (2011) therefore propose to rely on several reference images obtained from an Internet search based on semantic information.

Deep Learning Methods for Image Colorization

Since 2012, deep learning approaches, in particular convolutional neural networks (CNNs), have become very popular in the community of computer vision and computer graphics.

The first deep learning-based colorization methods were proposed in Cheng et al. (2015) and Deshpande et al. (2015). In Cheng et al. (2015), a fully automated system extracts handcrafted low and high features and feeds them as input to a three-layer fully connected neural network trained with a L2 loss. The network predicts the U and V channels of the YUV luminance-chrominance space. The authors also add an optional clustering stage where the images are divided in different types of scenes, according to the previously extracted semantic features. Then, a different neural network is trained for each of the clusters.

End-to-end approaches: Later on, papers focused more on *end-to-end approaches* (see Fig. 3).

For instance, the paper that won both tracks of the Gu et al. (2019) NTIRE 2019 Challenge on Image Colorization was the end-to-end method proposed by IPCV_IIMT. It implements an encoder-decoder structure that resembles to a U-Net with the encoder built using deep dense-residual blocks. Wan et al. (2020a) proposed to combine neural networks with color propagation. It first trains a neural network in order to colorize interest points of extracted superpixels. Then those colors are propagated by optimizing an objective function. In an older work, Iizuka et al. (2016) presented an end-to-end colorization framework based on CNNs to

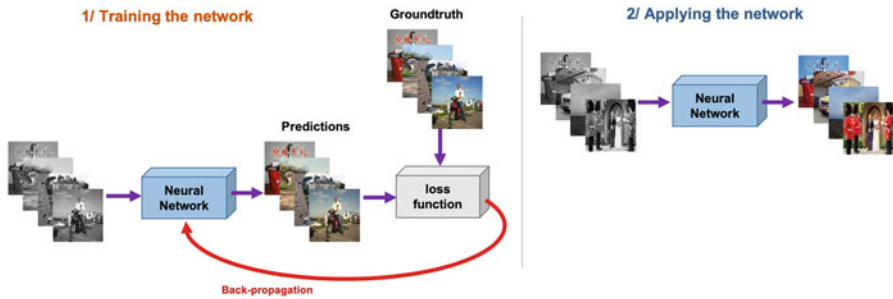


Fig. 3 Principle of basic end-to-end colorization networks

infer the ab channels of the CIE Lab color space. This work is built on the basis that a classification of the images can help to provide global priors that will improve the colorization performance. The network extracts global and local features and is jointly trained for classification and colorization in a labeled dataset.

Using GANs: Still being end-to-end, other methods use *generative adversarial networks* (GANs) (Goodfellow et al. 2014). Isola et al. (2017) propose the so-called image-to-image method pix2pix. It maps an input image to an output image using a U-Net generator and a patch GANs discriminator. The method is used in many applications including colorization. This method was extended in Nazeri et al. (2018) using deep convolutional GAN (DCGAN) (Radford et al. 2016). In Cao et al. (2017), a fully convolutional generator with a conditional GANs is considered. This architecture does not use downsampling to avoid extracting global features which are not suitable to recover accurate boundaries. To avoid noise attenuation and make the colorization results more diversified, they concatenate a noise channel onto the first half of the generator layers. GANs have also been used in chromaGAN (Vitoria et al. 2020) which extends Iizuka et al. (2016) by proposing to learn the semantic image distribution without any need of a labeled dataset. This method combines three losses: a color error loss by computing MSE on ab channels, a class distribution loss by computing the Kullback-Leibler divergence on VGG-16 class distribution vectors, and an adversarial Wasserstein GAN (WGAN) loss (Arjovsky et al. 2017). To prevent the need for training on a huge amount of data, Yoo et al. (2019) introduce MemoPainter, a few-shot colorization framework. MemoPainter is able to colorize an image with limited data by using an external memory network in addition to a colorization network. The memory network learns to retrieve a color feature that best matches the ground-truth color feature of the query image, while the generator-discriminator colorization network learns to effectively inject the color feature to the target grayscale image.

DeOldify (Antic 2019) is another end-to-end image and video colorization method mapping the missing chrominance values to the grayscale input image. A ResNet (ResNet101 or ResNet34) is used as the backbone of the generator

of a U-Net architecture trained as follows: the generator is first trained with the perceptual loss (Johnson et al. 2016), followed by training the critic as a binary classifier distinguishing between real images and those generated by the generator, and finally the generator and critic are trained together in an adversarial manner on 1–3% of the ImageNet (Deng et al. 2009) data. The latter is the so-called NoGAN strategy which is enough to add color realism to the results and which also allows to avoid flickering across video frames while the colorization is applied individually frame per frame.

Predicting distributions instead of images: Regression does not handle multi-modal color distributions well (Larsson et al. 2016). Larsson et al. (2016) and Zhang et al. (2016) address this issue by *predicting distributions* over a set of bins, as it was initially done in the exemplar-based method (Charpiat et al. 2008). They therefore rely on a discretization of color spaces. In Larsson et al. (2016), the color space is binned with evenly spaced Gaussian quantiles. Experiments are run for hue/chroma and Lab color spaces with either separated or joint distributions. Inference of the colored image from the distribution uses expectation (sum over the color bin centroids weighted by the histogram). In Zhang et al. (2016), the *ab* output space is quantized into bins with grid size 10 and the 313 values which are in gamut are kept. The inference is the annealed mean of the distribution. In Mouzon et al. (2019) and Pierre and Aujol (2020), the resulting distributions from Zhang et al. (2016) are later used in a variational approach (Pierre et al. 2015a).

Considering user priors: Few methods give the possibility to add user inputs as additional priors. The architecture in Zhang et al. (2017) learns to propagate color hints by fusing low-level cues and high-level semantic information. He et al. (2018) uses a reference colored image to guide the output of their deep exemplar-based colorization method.

Generating diverse image colorizations: Some methods have been designed to generate diverse colorizations as there is not one unique solution to the colorization problem. Deshpande et al. (2017) relied on a variational auto-encoder (VAE) to learn a low-dimensional embedding of color spaces. The mapping from a grayscale input image to color distribution of the latent space is done by learning a mixture density network (MDN). At test time, it is possible to sample the conditional model and use the VAE decoder to generate diverse colored images. In their PixColor model, Guadarrama et al. (2017) first train a conditional PixelCNN (Oord et al. 2016) to generate multiple latent low-resolution colored images, and then train a second CNN to generate the final high-resolution images. Another method, called PIC, that uses PixelCNN++ (Salimans et al. 2017) (an extension to the original PixelCNN), was proposed in Royer et al. (2017). A feed-forward CNN first maps grayscale image to an embedding that encodes color information. This embedding is then fed to the autoregressive PixelCNN++ model which predicts a distribution of image chromacity. The colTran model proposed by Kumar et al. (2021) is based on an axial transformer (Ho et al. 2019) autoregressive model. ColTran includes three networks, all

relying on column/row self-attention blocks: the autoregressive model that estimates low-resolution coarse colorization, a color upsampler, and a spatial upsampler.

Restoring and colorizing: Luo et al. (2020) propose to specifically restore and colorize old black and white portrait photos in a unified framework. It uses an additional high-quality color reference image (the sibling) automatically generated by first training a network that projects images into the StyleGAN2 (Karras et al. 2020) latent space and then uses the pretrained StyleGAN2 generator to create the sibling. Fine details and colors are extracted from the sibling. A latent code is then optimized through a three-term cost function and decoded by a StyleGAN2 generator yielding a high-quality color version of the antique input. The cost function is composed of a color term inspired by the style loss in Gatys et al. (2016a) between the features of the sibling and those of the generated high-quality colored image, a perceptual term (Johnson et al. 2016) between a degraded version of the generative model's output and the antique input, and a contextual term between the VGG features of the sibling and those of the generated high-quality colored image.

Decomposing the scene into objects: Recently, some methods try to explicitly deal with the decomposition of the scene into objects in order to tackle one of the main drawbacks of most deep learning-based colorization methods which is color bleeding across different objects. Su et al. (2020) proposed to colorize a grayscale image in an instance-aware fashion. They train three separate networks: a first one that performs global colorization, a second one that achieves instant colorization, and a third one that fuses both colorization networks. These networks are trained by minimizing the Huber loss (also called Smooth L1 loss). In general, after fusing both results the global colorization will be enhanced. The instances per image are obtained by using a standard pretrained object detection network, Mask R-CNN (He et al. 2017). Pucci et al. (2021) propose to improve Zhang et al. (2016) by using a network which is more aware of image instances, in the spirit of Su et al. (2020), by combining convolutional and capsule networks. They train from end to end a single network which first generates a per-pixel color distribution followed by a final convolutional layer that recovers the missing chrominance channels as opposed to Zhang et al. (2016) that computes the annealed mean on the per-pixel color distribution network's output. They train the network by minimizing the cross-entropy between per pixel color distributions and L2 loss on the chrominance channels. Kong et al. (2021) propose to colorize a grayscale image by training a multitask network for colorization and semantic segmentation in an adversarial manner. They train a U-Net-type network with a three-term cost function: a color regression loss in terms of hue, saturation, and lightness; the cross-entropy on the ground-truth and generated semantic labels; and a GANs term. The main objective of the proposal is to reduce color bleeding across edges.

Table 2 summarizes all these deep learning methods providing details on their particular inputs (other than the obvious grayscale image), their outputs, their

Table 2 Short description of deep networks for image colorization, their input, other than grayscale image, output. Here FCONV stands for fully convolutional, FC for fully connected, and U-Net for a U-Net-like network and not the vanilla U-Net

	Additional inputs	Network	Network's output	Post-processing
Cheng et al. (2015)	Handcrafted features	3 layers FC	UV	Joint bilateral filtering
Iizuka et al. (2016)	–	CNNs (local/global)	<i>ab</i>	Upsampling
Wan et al. (2020a)	Superpixels' handcrafted features	FC net	Interest points' color	Propagation and refinement
Using GANs				
Vitoria et al. (2020)	–	CNNs (local/global) + PatchGAN	<i>ab</i>	Upsampling
Nazeri et al. (2018)	–	U-Net (Isola et al. 2017) + DCGAN	Lab	–
Cao et al. (2017)	–	FCONV generator with multi-layer noise + PatchGAN	UV/RGB (diverse)	–
Yoo et al. (2019)	Color thief features	Colorization U-Net + memory nets noise	–	–
Antic (2019)	–	U-Net + self-attention + GAN	RGB	YUV conversion + cat(original Y/UV) + RGB conversion
Histogram prediction				
Larsson et al. (2016)	–	VGG-16 + FC layers	Distributions	Expectation
Zhang et al. (2016)	–	VGG-styled net	Distributions	Annealed mean
Mouzon et al. (2019)	–	Zhang et al. (2016)	Distributions	Variational model
User guided				
Zhang et al. (2017)	User point, global histograms, and average saturation	U-Net	Distributions + <i>ab</i>	–
He et al. (2018)	Color reference	Similarity sub-net + U-Net (gray VGG-19)	Bidirectional similarity maps + <i>ab</i>	–

(continued)

Table 2 (continued)

	Additional inputs	Network	Network's output	Post-processing
Diverse colorization and autoregressive models				
Deshpande et al. (2017)	—	cVAE + MDN	Diverse colorization	—
Guadarrama et al. (2017)	—	PixelCNN + CNN	Diverse colorization	—
Royer et al. (2017)	—	CNN + PixelCNN++	Diverse colorization	—
Kumar et al. (2021)	—	Axial transformer + color/spatial upsamplers (self-attention blocks)	Diverse colorization	—
Object aware				
Su et al. (2020)	Object bounding boxes	U-Net (global/instance) + CNN (fusion)	<i>ab</i>	—
Pucci et al. (2021)	—	CNN + capsule net	<i>ab</i>	—
Kong et al. (2021)	—	U-Net + PatchGAN	<i>ab</i> + semantic segmentation	—
Survey				
Gu et al. (2019)	—	U-Net	RGB	—

architectures, and pre- and post-processing steps. This summary table is only provided for deep learning-based methods since we focus on deep learning-based strategies in the remaining of the chapter.

Datasets Used in Literature

To train and test the deep learning methods presented in previous subsection, different datasets have been used. Table 3 summarizes the use of these datasets in colorization methods. They contain from one thousand (DIV2K (Agustsson and Timofte 2017)) to million of images (ImageNet (Deng et al. 2009)). Image dimensions also vary a lot, from 32×32 in CIFAR-10 (Krizhevsky et al. 2009) to 2K resolution in DIV2K.

Other differences concern the content of the images itself. Some datasets are very specific to a type of image: faces (LFW (Huang et al. 2007)) and bedrooms (LSUN (Yu et al. 2015)). Other present various scenes as Places (Zhou et al. 2017) with 205 scene categories, COCO (Lin et al. 2014) with 80 object categories and 91 stuff categories, and SUN (Xiao et al. 2010) with 899 scene categories.

Table 3 Datasets used in the literature for training or testing

	SUN	ImageNet/ILSVRC-2015	COCO	CIFAR-10	DIV2K	Pascal VOC	Places	LSUN bedroom or church	testing on historic BW photo	Remark/Other
Cheng et al. (2015)	•									
Iizuka et al. (2016)							•	•		
Using GANs										
Vitoria et al. (2020)		•							•	
Nazeri et al. (2018)				•			•			
Cao et al. (2017)								•		
Yoo et al. (2019)										Yumi, Monster, etc.
Antic (2019)		•								training on 1–3% of ImageNet images
Histograms prediction										
Larsson et al. (2016)	•	•								
Zhang et al. (2016)		•				•				training on 1.3M ImageNet images
User guided										
Zhang et al. (2017)		•								
He et al. (2018)		•								training on 700k ImageNet image/7 categories
Diverse										
Deshpande et al. (2017)		•						•		LFW
Guadarrama et al. (2017)		•								
Royer et al. (2017)		•		•						
Kumar et al. (2021)		•								
Object aware										
Su et al. (2020)		•	•				•			
Pucci et al. (2021)		•	•				•			
Kong et al. (2021)						•				
Survey										
Gu et al. (2019)					•					
Anwar et al. (2020)										Own Natural-Color Dataset

Proposed Colorization Framework

In this section, we present the framework that we will use for evaluating the influence of color spaces on image colorization results. First, we detail the architecture and, second, the dataset used for training and testing.

Note that the same architecture and training procedure are used in the ► [Chap. 21, “Analysis of Different Losses for Deep Learning Image Colorization”](#) of this handbook.

Detailed Architecture

The architecture used in our experiments is an encoder-decoder U-Net deep network composed of five stages (see Fig. 4). All convolutional blocks are composed of two 2D convolutional layers with 3×3 kernels, each one followed by 2D batch normalization (BN) and a ReLU activation. For the encoder, downsampling is done with max pooling layers after each convolutional block. After each downsampling, the number of filters is doubled in the following block. For the decoder, upsampling is done with 2D transpose convolutions (4×4 kernels with stride 2). At a given stage, the corresponding encoder and decoder blocks are linked with skip connections: feature maps from the encoder are concatenated with the ones from the corresponding upsampling path and fused using 1×1 convolutions. More details can be found in Table 4. The encoder architecture is identical to the CNN part of a VGG-19 network (Simonyan and Zisserman 2015). It allows us to start from pretrained weights initially used for ImageNet classification. Moreover, the encoder architecture choice was motivated by the fact that most deep learning-based approaches use a VGG-type architecture to generate the missing chrominances.

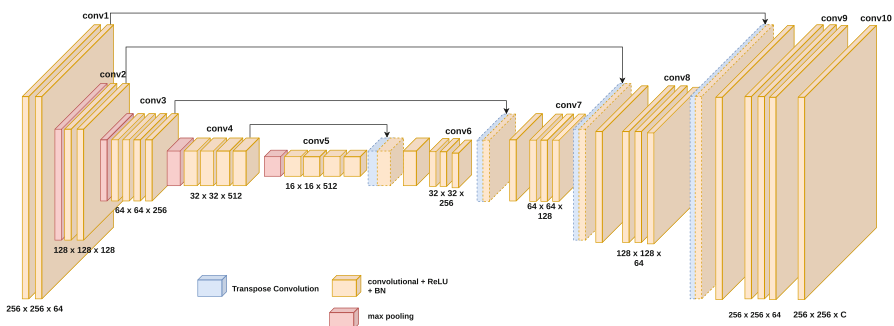


Fig. 4 Summary of the baseline U-Net architecture used in our experiments. It outputs a $256 \times 256 \times C$ image, where C stands for the number of channels, being equal to 2 when estimating the missing chrominance channels and to 3 when estimating the RGB components

Table 4 Detailed architecture and output resolution for each block

Layer type	Output resolution
Input	$3 \times H \times W$
Conv1 + Max-pooling	$64 \times H/2 \times W/2$
Conv2 + Max-pooling	$128 \times H/4 \times W/4$
Conv3 + Max-pooling	$256 \times H/8 \times W/8$
Conv4 + Max-pooling	$512 \times H/16 \times W/16$
Conv5 + Conv. Transpose (I)	$512 \times H/8 \times W/8$
Conv6 + Conv. Transpose (II)	$256 \times H/4 \times W/4$
Conv7 + Conv. Transpose (III)	$128 \times H/2 \times W/2$
Conv8 + Conv. Transpose (IV)	$64 \times H \times W$
Conv9	$64 \times H \times W$
Conv10	$C \times H \times W$

The training settings are described as follows:

- Optimizer: Adam
- Learning rate: $2e-5$ as in ChromaGAN (Vitoria et al. 2020).
- Batch size: 16 images (approx. 11 GB RAM usage on Nvidia Titan V).
- All images are resized to 256×256 for training which enable using batches. In practice, to keep the aspect ratio, the image is resized such that the smallest dimension matches 256. If the other dimension remains larger than 256, we then apply a random crop to obtain a square image. Note that the random crop is performed using the same seed for all trainings.

When generating images, it is crucial to remain in the range of acceptable values of color spaces. In particular, we must ensure that the final image takes values between 0 and 255. In our implementation, we use simple clipping on final RGB values. Other strategies are sometimes considered as in Iizuka et al. (2016) where the $a*b*$ components are globally normalized so they lie in the $[0,1]$ range of the Sigmoid transfer function.

Training and Testing Images

Throughout our experiments we use the COCO dataset (Lin et al. 2014), containing various natural images of different sizes. COCO is divided into three sets that approximately contain 118k, 5k, and 40k images that, respectively, correspond to the training, validation, and test sets. Note that we carefully remove all grayscale images, which represent around 3% of the overall amount of each set. Although larger datasets such as ImageNet have been regularly used in the literature, COCO offers a sufficient number and a good variety of images so we can efficiently train and compare numerous models.

Learning Strategy for Different Color Spaces

The goal of the whole colorization process is to generate RGB images that look visually natural. When training on different color spaces, one must decide which color space is used to compute losses and when is the conversion back to RGB performed. In this chapter, we propose to experiment with three learning strategies to compare RGB, YUV, and Lab color spaces (see Fig. 5):

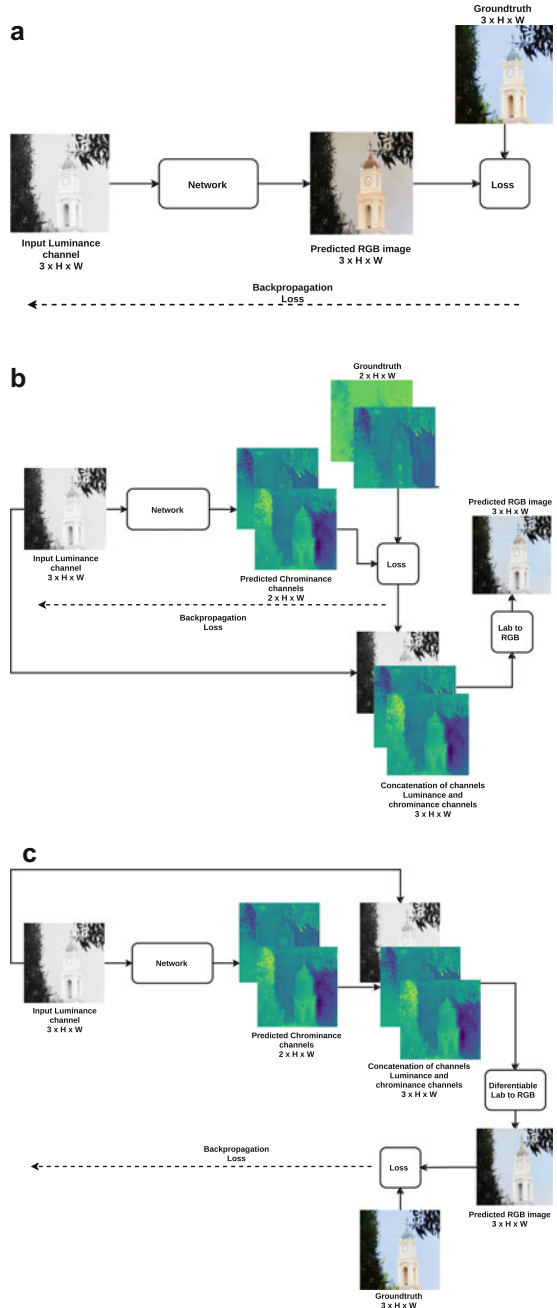
- *RGB*: in this case, the network takes as input a grayscale image L and directly estimates a three-channel RGB image of size $256 \times 256 \times 3$. The loss is done directly in the RGB color space. This strategy is illustrated in Fig. 5a.
- *YUV and Lab Luminance/chrominance*: in this case, the network takes as input a grayscale image considered as the luminance (L for Lab, Y for YUV) and outputs two chrominance channels (a, b or U, V). The loss compares the output with the corresponding chrominance channels of the ground-truth image converted to the luminance/chrominance space. After concatenating the initial luminance channel to the inferred chrominances, the image is converted back to RGB for visualization purposes. This strategy is illustrated in Fig. 5b.
- *LabRGB*: as in the previous case, the network takes as input the luminance and estimates the corresponding two chrominance channels. After concatenating with the corresponding luminance channel, they are converted to the RGB color space and the loss is computed directly there. Notice that in this last case, as the loss is computed on RGB color space, the conversion must be done in a way that is differentiable to be able to compute the gradient and allow the backpropagation step. We perform the color conversion using the color module in the Kornia library. Kornia (Riba et al. 2020) is a differentiable library that consists of a set of routines and differentiable modules to solve generic computer vision problems. It allows classical computer vision tasks to be integrated into deep learning models. Computing the loss on RGB images instead of chrominance ones enables to ensure images are similar to ground truth after the clipping operation needed to fit into the RGB cube. This strategy is illustrated in Fig. 5c.

Remark. During training, all images are resized to 256×256 . One advantage of using luminance/chrominance spaces is that only chrominance channels are resized. It is therefore possible to keep the original content of the luminance channels without manipulating it with the resizing steps.

Analysis of the Influence of Color Spaces

This section presents quantitative and qualitative results obtained with the three strategies discussed above. For this analysis, we have considered, as loss function, the L2 loss and the VGG-based LPIPS which was introduced in Ding et al. (2021) as a generalization of the feature loss (Johnson et al. 2016). These loss functions are defined hereafter.

Fig. 5 Illustration of the different learning strategies for our proposed framework. **(a)** Learning strategy directly predicting the RGB colors. **(b)** Learning strategy predicting the two chrominance channels. **(c)** Learning strategy predicting the two chrominance channels and then converting to RGB



MSE or squared L2 loss. The L2 loss, between two functions u and v defined on Ω and with values in \mathbb{R}^C , $C \in \mathbb{N}$, is defined as the squared L2 loss of their difference. That is,

$$\text{MSE}(u, v) = \|u - v\|_{L^2(\Omega; \mathbb{R}^C)}^2 = \int_{\Omega} \|u(x) - v(x)\|_2^2 dx, \quad (2)$$

where $\|\cdot\|_2$ denotes the Euclidean norm in \mathbb{R}^C . In the discrete setting, it is equal to the sum of the square differences between the image values, that is,

$$\text{MSE}(u, v) = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^C (u_{i,j,k} - v_{i,j,k})^2. \quad (3)$$

Feature Loss. The feature reconstruction loss (Gatys et al. 2016b; Johnson et al. 2016) is a perceptual loss that encourages images to have similar feature representations as the ones computed by a pretrained network, denoted here by Φ . Let $\Phi_l(u)$ be the activation of the l -th layer of the network Φ when processing the image u ; if l is a convolutional layer, then $\Phi_l(u)$ will be a feature map of size $C_l \times W_l \times H_l$. The *feature reconstruction* loss is the normalized squared Euclidean distance between feature representations, that is,

$$\mathcal{L}_{\text{feat}}^l(u, v) = \frac{1}{C_l W_l H_l} \|\Phi_l(u) - \Phi_l(v)\|_2^2. \quad (4)$$

It penalizes the output reconstructed image when it deviates in feature content from the target.

LPIPS. LPIPS (Zhang et al. 2018) computes a weighted L2 distance between deep features of a pair of images u and v :

$$\text{LPIPS}(u, v) = \sum_l \frac{1}{H_l W_l} \sum_{i=1}^{H_l} \sum_{j=1}^{W_l} \|\omega_l \odot (\Phi_l(u)_{i,j} - \Phi_l(v)_{i,j})\|_2^2, \quad (5)$$

where H_l (resp. W_l) is the height (resp. the width) of feature map Φ_l at layer l and ω_l is the weight for each feature. Note that features are unit-normalized in the channel dimension. We will denote VGG-based LPIPS when feature maps Φ_l are taken from a VGG network.

Note that to compute the VGG-based LPIPS loss, the output colorization always has to be converted to RGB, even for YUV and Lab color spaces (as in Fig. 5c), because this loss is computed with a pretrained VGG expecting RGB images as input. Since VGG-based LPIPS is computed on RGB images, the two strategies *Lab* and *LabRGB* are the same. For more details on the various losses usually used in colorization, we refer the reader to the ► [Chap. 21, “Analysis of Different](#)

Losses for Deep Learning Image Colorization. Our experiments have shown that same conclusions can be drawn with other losses.

For testing, we apply the network to images at their original resolution, while training is done on batches of square 256×256 images.

Quantitative Evaluation

There is no standard protocol for quantitative evaluation of automatic colorization methods. We refer the reader to the ► [Chap. 21, “Analysis of Different Losses for Deep Learning Image Colorization”](#) for a detailed survey of quantitative evaluation methods used in image colorization literature and analysis of correlation between losses and type of evaluation metrics used. We choose to rely on the more generally used and more recent ones: L1 (MAE), L2 (MSE), PSNR, SSIM (Wang et al. 2004), LPIPS (Zhang et al. 2018), and FID (Fréchet Inception Distance) (Dowson and Landau 1982), which are defined hereafter.

MAE or L1 loss with l^1 -coupling. The mean absolute error is defined as the L1 loss with l^1 -coupling, that is,

$$\text{MAE}(u, v) = \int_{\Omega} \|u(x) - v(x)\|_{l^1} dx = \int_{\Omega} \sum_{k=1}^C |u_k(x) - v_k(x)| dx. \quad (6)$$

In the discrete setting, it coincides with the sum of the absolute differences $|u_{i,j,k} - v_{i,j,k}|$. Some authors use a l^2 -coupled version of it:

$$\text{MAE}^c(u, v) = \sum_{i=1}^M \sum_{j=1}^N \sqrt{\sum_{k=1}^C (u_{i,j,k} - v_{i,j,k})^2}. \quad (7)$$

Both MAE and MAE^c losses are robust to outliers.

PSNR. The PSNR measures the ratio between the maximum value of a color target image $u : \Omega \rightarrow \mathbb{R}^C$ and the mean square error (MSE) between u and a colorized image $v : \Omega \rightarrow \mathbb{R}^C$ with $\Omega \in \mathbb{Z}^2$ a discrete grid of size $M \times N$. That is,

$$\text{PSNR}(u, v) = 20 \log_{10}(\max u) - 10 \log_{10} \left(\frac{1}{CMN} \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^N (u(i, j, k) - v(i, j, k))^2 \right), \quad (8)$$

where $C = 3$ when working in the RGB color space and $C = 2$ in any luminance-chrominance color space as YUV, Lab, and YCbCr. The PSNR score is considered as a reconstruction measure tending to favor methods that will output results as close as possible to the ground-truth image in terms of the MSE.

SSIM. SSIM intends to measure the perceived change in structural information between two images. It combines three measures to compare image color (l), contrast (c), and structure (s):

$$\text{SSIM}(u, v) = l(u, v)c(u, v)s(u, v) = \frac{(2\mu_u\mu_v) + c_1 (2\sigma_u\sigma_v + c_2) (\sigma_{uv} + c_3)}{\mu_u^2 + \mu_v^2 + c_1 \sigma_u^2 + \sigma_v^2 + c_2 \sigma_u\sigma_v + c_3}, \quad (9)$$

where μ_u (resp. σ_u) is the mean value (resp. the variance) of image u values and σ_{uv} the covariance of u and v . c_1, c_2, c_3 are regularization constants that are used to stabilize the division for images with mean or standard deviation close to zero.

FID. FID (Heusel et al. 2017) is a quantitative measure used to evaluate the quality of the outputs' generative model and which aims at approximating human perceptual evaluation. It is based on the Fréchet distance (Dowson and Landau 1982) which measures the distance between two multivariate Gaussian distributions. FID is computed between the feature-wise mean and covariance matrices of the features extracted from an Inception v3 neural network applied to the input images (μ_r, Σ_r) and those of the generated images (μ_g, Σ_g):

$$\text{FID}((\mu_r, \Sigma_r), (\mu_g, \Sigma_g)) = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\Sigma_r \Sigma_g)^{1/2}. \quad (10)$$

The results are presented in Table 5. In terms of these metrics, the best results are obtained with YUV color space except for L1 and Fréchet Inception Distance, even if not by much. The results in Table 5 also indicate that Lab does not outperform other color spaces when using a classic reconstruction loss (L2), while better results are obtained when using the VGG-based LPIPS. Thus, using a feature-based reconstruction loss is better suited as was already the case in exemplar-based image colorization methods where different features for patch-based metrics were proposed for matching pixels. LabRGB strategy gets the worst quantitative results based on Table 5. One would expect to get the “best of both” color spaces while recovering from the loss of information in the conversion process. However, this is not reflected with these particular evaluation metrics. The LabRGB line for VGG-based LPIPS is not included, as it would be identical to the Lab one. Also, note that the quantitative evaluation is performed on RGB images as opposed to training which is done for specific color spaces (RGB, YUV, Lab, and LabRGB).

Table 5 Quantitative evaluation of colorization results for different color spaces. Metrics are used to compare ground truth to every image in the 40k test set. Best and second best results by column are in bold and italicized respectively

Color space	Loss function	L1 ↓	L2 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
RGB	L2	0.04458	0.00587	22.3136	<i>0.9255</i>	<i>0.1606</i>	7.4223
YUV	L2	<i>0.04469</i>	0.00562	22.5052	0.9278	0.1593	<i>7.6642</i>
Lab	L2	0.04488	<i>0.00585</i>	<i>22.3283</i>	0.9250	0.1613	8.1517
LabRGB	L2	0.04608	0.00589	22.2989	0.9209	0.1698	8.3413
RGB	LPIPS	0.04573	0.00577	22.3892	<i>0.9197</i>	0.1429	3.0576
YUV	LPIPS	<i>0.04460</i>	0.00557	22.5438	0.9097	0.1400	3.3260
Lab	LPIPS	0.04374	<i>0.00566</i>	<i>22.4699</i>	0.9228	<i>0.1403</i>	<i>3.2221</i>

Qualitative Evaluation

In this section, we qualitatively analyze the results obtained by training the network with different color spaces as explained in section “[Learning Strategy for Different Color Spaces](#)”.

Figure 6 shows results on images and objects (here person skiing, stop sign and zebra) with strong contours that were highly present in the training set. The colorization of these images is really impressive for any color space. Nevertheless, YUV has the tendency to sometimes create artifacts that are not predictable. This is visible with the blue stain in the YUV-L2 zebra and the yellow spot in the YUV-LPIPS zebra. One can also notice that the overall colorization tends to be more homogeneous with LabRGB-L2 than with Lab-L2 as can be seen, for instance, on the wall behind the stop signs, the grass, and tree leaves in the zebra image which suggest that it might be better to compute losses over RGB images. A similar remark is valid for the VGG-based LPIPS results as can be seen, for instance, in the homogeneous colorization of the sky in the person skiing image where the loss is again computed over the RGB image. This indicates that there could be an additional influence on the results when using VGG-based LPIPS given that the predicted colored image is converted back to RGB before backpropagation.

Figure 7 presents results on images where the final colorization is not consistent over the whole image. On the first row, the color of the water is stopped by the chair legs. On the second row, the colors of the grass and the sky are not always similar on both sides of the hydrant. LabRGB seems to reduce this effect. This happens when strong contours seem to stop the colorization and are independent on the color space. Global coherency can only be obtained if the receptive field is large enough and that self-similarities present in natural images are preserved. These results highlight that efforts must be put on the design of architectures that would impose these constraints.

One major problem in automatic colorization results comes from color bleedings that occur as soon as contours are not strong enough. Figure 8 illustrates this problem in different contexts. On the first row, the color from the flowers bleeds

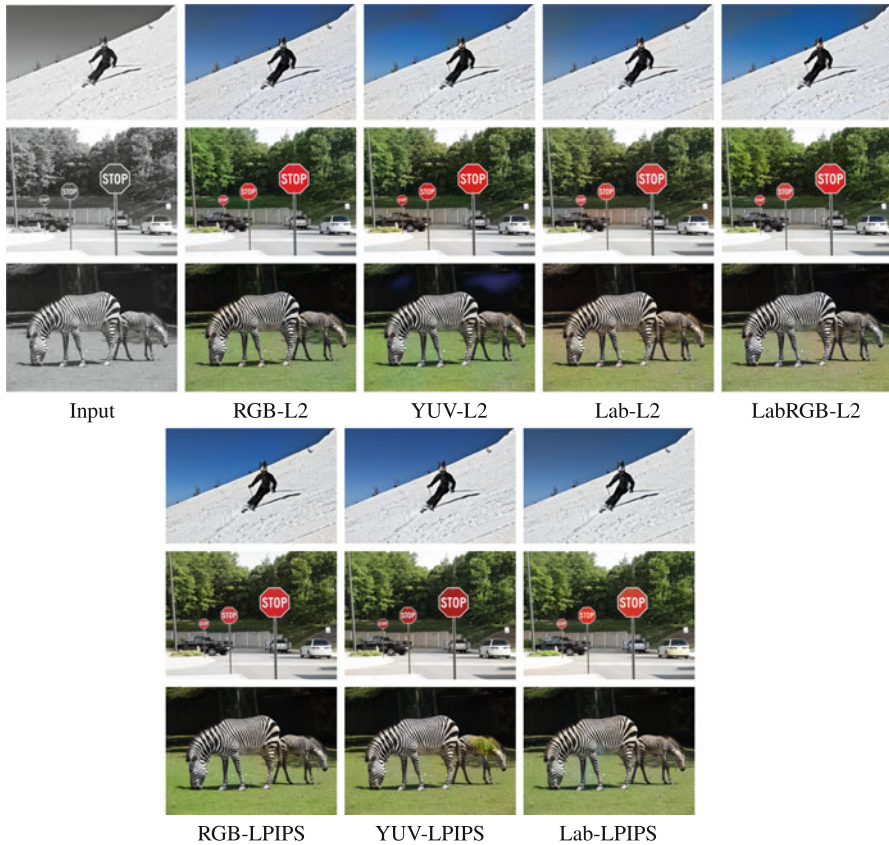


Fig. 6 Colorization results with different color spaces on images that contain objects, have strong structures, and have been seen many times in the training set. The three first rows are with L2 loss and the three last ones with VGG-based LPIPS

to the wall. On the second row, the green of the grass bleeds to the shorts. Finally, on the last row, the green of the grass bleeds to the neck of the background cow. These effects are independent from the color space or the loss. Some methods reduce this effect by introducing semantic information (e.g., Vitoria et al. 2020) or spatial localization (e.g., Su et al. 2020), while others achieve to reduce it by considering segmentation as an additional task (e.g., Kong et al. 2021). Note that with the VGG-based LPIPS, Lab color space provides more realistic result on the tennis man image.

Finally, Fig. 9 presents colorization of images containing many different objects. We see that final colors might be dependent on the color spaces and are more diverse and colorful with Lab color space. LabRGB strategy with L2 loss is probably the more realistic statement that holds with the VGG-based LPIPS.



Fig. 7 Colorization results with different color spaces on images that exhibit strong structures that may lead to inconsistent spatial colors. The two first rows are with L2 loss and the two last ones with VGG-based LPIPS

The qualitative evaluation does not point to the same conclusion as the quantitative one. According to Table 5, the best colorization is obtained for YUV color space. However, the qualitative analysis shows that even if in some cases colors are brighter and more saturated in other ones, it creates unpredictable color stains (yellowish and blueish). This raises the question on the necessity to design specific metrics for the colorization task, which should be combined with user studies. Also, in the qualitative evaluation, one can observe that when working with LabRGB instead of Lab, the overall colorization result looks more stable and homogeneous as opposed to what is concluded in the quantitative evaluation.

Summary of qualitative analysis: Our analysis leads us to the following conclusions:

- There is no major difference in the results regarding the color space that is used.
- YUV color space sometimes generates color artifacts that are hardly predictable. This is probably due to clipping that is necessary to remain in the color space range of values.
- More realistic and consistent results are obtained when losses are computed in the RGB color space.
- There is no evidence justifying why most colorization methods in the literature choose to work with Lab. One can assume that this is mainly done to ease the colorization problem by working in a perceptual luminance-chrominance color space. In addition, differentiable color conversion libraries were not available up



Fig. 8 Colorization results with different color spaces on images that contain small contours which lead to color bleeding. The two first rows are with L2 loss and the two last ones with VGG-based LPIPS

to 2020 to apply a strategy as in Fig. 5c. In fact, the qualitative results show that when training on RGB, the luminance reconstruction is satisfying in all examples. Hence, there is no obvious reason why not to work directly in RGB color space.

- Same conclusions hold with different losses.

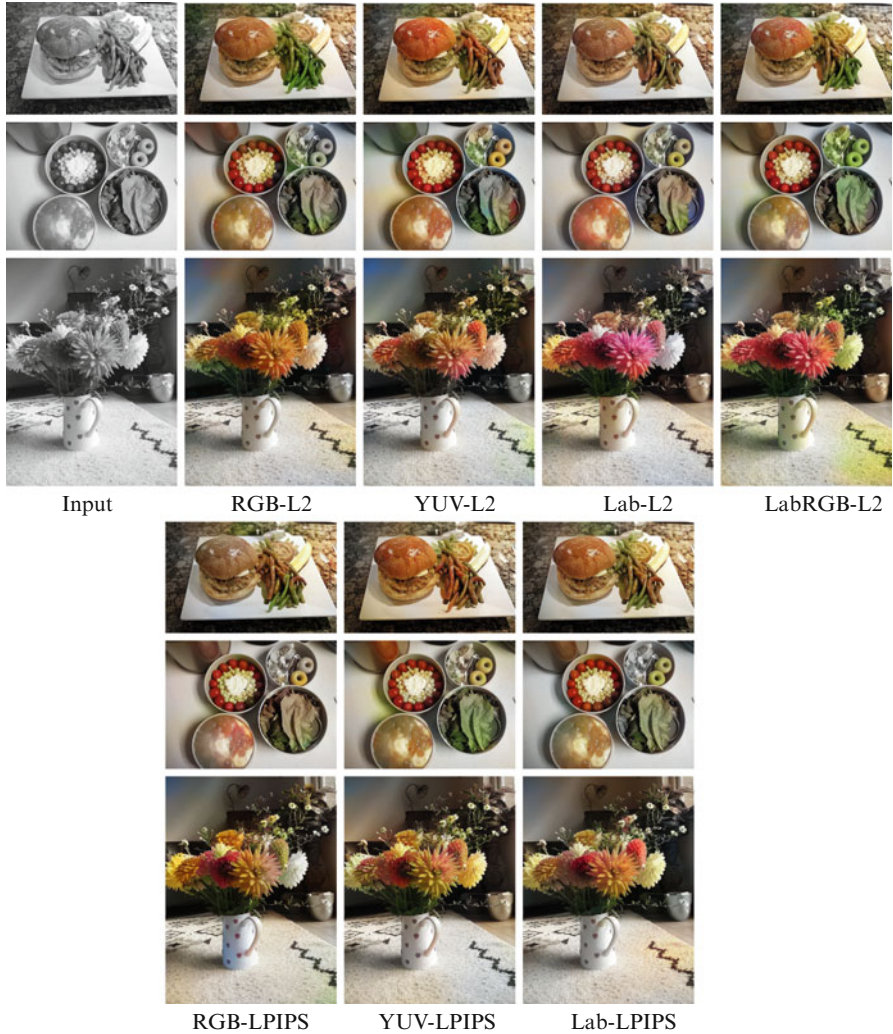


Fig. 9 Colorization results with different color spaces on images that contain several small objects which end up with different colors depending on the color spaces used. The three first rows are with L2 loss and the three last ones with VGG-based LPIPS

Generalization to Archive Images

Archive images present many artifacts due to acquisition methods (analog or numeric with different material qualities and manufacturing processes) and preservation conditions. They lead to images with different resolutions, film grains, scratches and holes, flickering, etc. Tools available for professional colorization



Fig. 10 Colorization results with different color spaces and L2 or VGG-based LPIPS on archive black and white images

enable artists to reach high-level quality images but require long human intervention. The current pipeline for professional colorization usually starts with restoration: denoising, deblurring, completion, super-resolution with off-the-shelf tools (e.g., Diamant) and manual correction. Next, images are segmented into objects and manually colorized by specialists with color spectrum that must be historically and artistically correct.

Automatic colorization methods could at least help professionals in the last step. Very few papers in the literature tackle old black and white images' colorization. In deep learning-based approaches, Vitoria et al. (2020) and Antic (2019) present some results on Legacy Black and White Photographs, while Luo et al. (2020) restore and colorize old black and white portraits. Wan et al. (2020b) focus on the restoration of old photos by training two variational autoencoders (VAE) to project clean and old photos to two latent spaces and to learn the translation between these latent spaces on synthetic paired data. Old photos are synthesized using Pascal VOC dataset's images.

Figure 10 presents some results obtained by applying the networks trained in this chapter on archive images. As we can observe on the second, third, and fourth rows, while on clean images sky and grass are often well colorized, it is not the case on archive images. This is probably due to the grain and noise in these images. Similarly the skin of persons is not as well colorized as in clean images. Color bleeding is here again a real issue. On the other hand, for objects with strong contours that were present in the database (e.g., stop sign), the colorization works very well. This indicates the importance on training or fine tuning on images that are related to the purpose of the network (many of the objects present in old black and white photos are not well represented with the most often used datasets).

Conclusion

This chapter has presented the role of the color spaces on automatic colorization with deep learning. Using a fixed standard network, we have shown, qualitatively and quantitatively, that the choice of the right color space is not straightforward and might depend on several factors such as the architecture or the type of images. With our architecture, the best quantitative results are obtained in YUV, while qualitative results rather teach us to compute losses in the RGB color space. We therefore argue that most efforts should be made on the architecture design. Furthermore, for all methods the final step consists in clipping final values to fit in the RGB color cube. This abrupt operation sometimes leads to artifacts with saturated pixels. An interesting topic for future research would be to learn a model that learns a projection into the color cube while preserving good image quality, similar to the geometric model from Pierre et al. (2015b). Future works should also include the development of methods that would give the possibility to produce several outputs in the same trend as HistoGAN (Afifi et al. 2021). Finally, if the purpose of colorization is often to enhance old black and white images, research papers rarely focus on this application. Strategies for better training or transfer learning must be developed in the future along with complete architectures that perform colorization together with other quality improvement methods such as super resolution, denoising, or deblurring.

Acknowledgments This study has been carried out with financial support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01) and from the EU Horizon 2020 research and innovation program NoMADS (Marie Skłodowska-Curie grant agreement No 777826). This chapter was written together with another chapter of the current handbook, ► [Chap. 21, “Analysis of Different Losses for Deep Learning Image Colorization.”](#) All authors have contributed to both chapters.

References

- Afifi, M., Brubaker, M.A., Brown, M.S.: HistoGAN: controlling colors of gan-generated and real images via color histograms. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7941–7950 (2021)
- Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: dataset and study. In: Conference on Computer Vision and Pattern Recognition Workshops, pp. 126–135 (2017)
- Antic, J.: Deoldify (2019). <https://github.com/jantic/DeOldify>
- Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F.S., Muzaffar, A.W.: Image colorization: a survey and dataset (2020). arXiv preprint arXiv:2008.10774
- Arbelot, B., Vergne, R., Hurtut, T., Thollot, J.: Automatic texture guided color transfer and colorization. In: Expressive, Elsevier, pp. 21–32 (2016)
- Arbelot, B., Vergne, R., Hurtut, T., Thollot, J.: Local texture-based color transfer and colorization. *Comput. Graph.* **62**, 15–27 (2017)
- Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, vol. 70, pp. 214–223 (2017)
- Bugeau, A., Ta, V.-T.: Patch-based image colorization. In: International Conference on Pattern Recognition, pp. 3058–3061 (2012)
- Bugeau, A., Ta, V.-T., Papadakis, N.: Variational exemplar-based image colorization. *IEEE Trans. Image Process.* **23**(1), 298–307 (2014)
- Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 151–166 (2017)
- Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: European Conference on Computer Vision, pp. 126–139 (2008)
- Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: IEEE International Conference on Computer Vision, pp. 415–423 (2015)
- Chia, A.Y.-S., Zhuo, S., Gupta, R.K., Tai, Y.-W., Cho, S.-Y., Tan, P., Lin, S.: Semantic colorization with internet images. In: ACM SIGGRAPH ASIA (2011)
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
- Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: IEEE International Conference on Computer Vision (2015)
- Deshpande, A., Lu, J., Yeh, M.-C., Jin Chong, M., Forsyth, D.: Learning diverse image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6837–6845 (2017)
- Di Blasi, G., Reforgiato, D.: Fast colorization of gray images. In: Eurographics Italian, Eurographics Association (2003)
- Ding, X., Xu, Y., Deng, L., Yang, X.: Colorization using quaternion algebra with automatic scribble generation. In: Advances in Multimedia Modeling, pp. 103–114 (2012)
- Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Comparison of full-reference image quality models for optimization of image processing systems. *Int. J. Comput. Vis.* **129**(4), 1258–1281 (2021)
- Dowson, D., Landau, B.: The Fréchet distance between multivariate normal distributions. *J. Multivar. Anal.* **12**(3), 450–455 (1982)
- Drew, M.S., Finlayson, G.D.: Improvement of colorization realism via the structure tensor. *Int. J. Image Graph.* **11**(4), 589–609 (2011)
- Ebner, M.: Color Constancy, vol. 7. Wiley, Hoboken (2007)

- Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision, pp. 1033–1038 (1999)
- Fairchild, M.D.: *Color Appearance Models*. Wiley, Hoboken (2013)
- Fang, F., Wang, T., Zeng, T., Zhang, G.: A superpixel-based variational model for image colorization. *IEEE Trans. Vis. Comput. Graph.* **26**(10), 2931–2943 (2019)
- Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423 (2016a)
- Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. *J. Vis.* **16**(12), 326 (2016b)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems* (2014)
- Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., Murphy, K.: Pixcolor: pixel recursive colorization. In: *British Machine Vision Conference* (2017)
- Gu, S., Timofte, R., Zhang, R.: Ntire 2019 challenge on image colorization: report. In: *Conference on Computer Vision and Pattern Recognition Workshops* (2019)
- Gupta, R.K., Chia, A.Y.-S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: *ACM International Conference on Multimedia*, pp. 369–378 (2012)
- He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)
- He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. *ACM Trans. Graph.* **37**(4), 1–16 (2018)
- Heu, J., Hyun, D.-Y., Kim, C.-S., Lee, S.-U.: Image and video colorization based on prioritized source propagation. In: *IEEE International Conference on Image Processing*, pp. 465–468 (2009)
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
- Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180* (2019)
- Huang, Y.-C., Tung, Y.-S., Chen, J.-C., Wang, S.-W., Wu, J.-L.: An adaptive edge detection based colorization algorithm and its applications. In: *ACM International Conference on Multimedia*, pp. 351–354 (2005)
- Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst (2007)
- Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.* **35**(4), 1–11 (2016)
- Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: *Eurographics Conference on Rendering Techniques* (2005)
- Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134 (2017)
- Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: *European Conference on Computer Vision*, pp. 694–711 (2016)
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119 (2020)
- Kawulok, M., Kawulok, J., Smolka, B.: Discriminative textural features for image and video colorization. *IEICE Trans. Inf. Syst.* **95-D**(7), 1722–1730 (2012)
- Kong, G., Tian, H., Duan, X., Long, H.: Adversarial edge-aware image colorization with semantic segmentation. *IEEE Access* **9**, 28194–28203 (2021)

- Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
- Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. arXiv preprint arXiv:2102.04432 (2021)
- Lagodzinski, P., Smolka, B.: Digital image colorization based on probabilistic distance transformation. In: 50th International Symposium ELMAR, vol. 2, pp. 495–498 (2008)
- Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European Conference on Computer Vision, pp. 577–593 (2016)
- Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. *ACM Trans. Graph.* **23**(3), 689–694 (2004)
- Lézoray, O., Ta, V.-T., Elmoataz, A.: Nonlocal graph regularization for image colorization. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
- Li, B., Lai, Y.-K., Rosin, P.L.: Example-based image colorization via automatic feature selection and fusion. *Neurocomputing* **266**, 687–698 (2017a)
- Li, B., Zhao, F., Su, Z., Liang, X., Lai, Y.-K., Rosin, P.L.: Example-based image colorization using locality consistent sparse representation. *IEEE Trans. Image Process.* **26**(11), 5188–5202 (2017b)
- Li, B., Lai, Y.-K., John, M., Rosin, P.L.: Automatic example-based image colorization using location-aware cross-scale matching. *IEEE Trans. Image Process.* **28**(9), 4606–4619 (2019)
- Li, B., Lai, Y.-K., Rosin, P.L.: A review of image colourisation. In: *Handbook of Pattern Recognition and Computer Vision*, p. 139. World Scientific, Singapore (2020)
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014)
- Ling, Y., Au, O.C., Pang, J., Zeng, J., Yuan, Y., Zheng, A.: Image colorization via color propagation and rank minimization. In: IEEE International Conference on Image Processing, pp. 4228–4232 (2015)
- Liu, S., Zhang, X.: Automatic grayscale image colorization using histogram regression. *Pattern Recogn. Lett.* **33**(13), 1673–1681 (2012)
- Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.-Q., Shum, H.-Y.: Natural image colorization. In: Eurographics Conference on Rendering Techniques, pp. 309–320 (2007)
- Luo, X., Zhang, X., Yoo, P., Martin-Brualla, R., Lawrence, J., Seitz, S.M.: Time-travel photography. arXiv preprint arXiv:2012.12261 (2020)
- Mouzon, T., Pierre, F., Berger, M.-O.: Joint CNN and variational model for fully-automatic image colorization. In: Scale Space and Variational Methods in Computer Vision, pp. 535–546 (2019)
- Nazeri, K., Ng, E., Ebrahimi, M.: Image colorization using generative adversarial networks. In: International Conference on Articulated Motion and Deformable Objects, pp. 85–94 (2018)
- Oord, A.V.D., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with PixelCNN decoders. In: *Advances in Neural Information Processing Systems* (2016)
- Pang, J., Au, O.C., Tang, K., Guo, Y.: Image colorization using sparse representation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1578–1582 (2013)
- Pierre, F., Aujol, J.-F.: Recent approaches for image colorization. In: *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, Springer (2020)
- Pierre, F., Aujol, J.-F., Bugeau, A., Ta, V.-T.: A unified model for image colorization. In: European Conference on Computer Vision Workshops, pp. 297–308 (2014)
- Pierre, F., Aujol, J.-F., Bugeau, A., Papadakis, N., Ta, V.-T.: Luminance-chrominance model for image colorization. *SIAM J. Imaging Sci.* **8**(1), 536–563 (2015a)
- Pierre, F., Aujol, J.-F., Bugeau, A., Ta, V.-T.: Luminance-Hue Specification in the RGB Space. In: Scale Space and Variational Methods in Computer Vision, pp. 413–424. Springer, Cham (2015b)
- Pucci, R., Micheloni, C., Martinel, N.: Collaborative image and object level features for image colourisation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2160–2169 (2021)

- Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (2016)
- Ren, X., Malik, J.: Learning a classification model for segmentation. In: IEEE International Conference on Computer Vision, pp. 10–17 (2003)
- Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for PyTorch. In: Winter Conference on Applications of Computer Vision, pp. 3674–3683 (2020)
- Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization. In: British Machine Vision Conference (2017)
- Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: PixelCNN++: improving the PixelCNN with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517 (2017)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
- Su, J.-W., Chu, H.-K., Huang, J.-B.: Instance-aware image colorization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7968–7977 (2020)
- Tai, Y.-W., Jia, J., Tang, C.-K.: Local color transfer via probabilistic segmentation by expectation-maximization. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 747–754 (2005)
- Vitoria, P., Raad, L., Ballester, C.: ChromaGAN: adversarial picture colorization with semantic class distribution. In: Winter Conference on Applications of Computer Vision, pp. 2445–2454 (2020)
- Wan, S., Xia, Y., Qi, L., Yang, Y.-H., Atiquzzaman, M.: Automated colorization of a grayscale image with seed points propagation. *IEEE Trans. Multimedia* **22**(7), 1756–1768 (2020a)
- Wan, Z., Zhang, B., Chen, D., Zhang, P., Chen, D., Liao, J., Wen, F.: Bringing old photos back to life. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2747–2757 (2020b)
- Wang, S., Zhang, Z.: Colorization by matrix completion. In: AAAI Conference on Artificial Intelligence (2012)
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
- Welsh, T., Ashikhmin, M., Mueller, K. Transferring color to greyscale images. *ACM Trans. Graph.* **21**(3), 277–280 (2002)
- Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3485–3492 (2010)
- Yao, Q., James, T.K.: Colorization by patch-based local low-rank matrix completion. In: AAAI Conference on Artificial Intelligence (2015)
- Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. *IEEE Trans. Image Process.* **15**(5), 1120–1129 (2006)
- Yoo, S., Bahng, H., Chung, S., Lee, J., Chang, J., Choo, J.: Coloring with limited data: few-shot colorization via memory augmented networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
- Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
- Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision, pp. 649–666 (2016)
- Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.* **36**, 1–11 (2017)
- Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(6), 1452–1464 (2017)



Variational Model-Based Deep Neural Networks for Image Reconstruction

23

Yunmei Chen, Xiaojing Ye, and Qingchao Zhang

Contents

Introduction	880
Learned Algorithm for Specified Optimization Problem	882
Structured Image Reconstruction Networks	885
Proximal Point Network	886
ISTA-Net	888
ADMM-Net	891
Variational Network	894
Primal-Dual Network	896
Learnable Descent Algorithm	898
Concluding Remarks	901
References	905

Abstract

In recent years, we have witnessed unprecedented growth of research interests in deep learning approaches to image reconstruction. A majority of these approaches are inspired by the well-developed variational method and associated optimization algorithms for the inverse problem of image reconstruction. These approaches mimic the iterative schemes of the standard optimization algorithms but integrate learnable components to form structured deep neural networks and employ large amount of observation data to train the networks for the specific reconstruction tasks. They have demonstrated significantly improved

Y. Chen (✉) · Q. Zhang

Department of Mathematics, University of Florida, Gainesville, FL, USA

e-mail: yun@math.ufl.edu; qingchaozhang@ufl.edu

X. Ye

Department of Mathematics and Statistics, Georgia State University, Atlanta, GA, USA

e-mail: xye@gsu.edu

empirical performance and require much lower computational cost compared to the classical methods in a variety of applications. We provide the details of the derivations, the network architectures, and the training procedures for several typical networks in this field.

Keywords

Image reconstruction · Variational method · Deep neural network · Optimization

Introduction

Variational method has been one of the most mature and effective approaches for solving inverse problems in imaging Aubert and Vese (1997), Dal Maso et al. (1992), Koepfler et al. (1994), and Scherzer et al. (2009). In the context of image reconstruction, the inverse problem can be formulated as an optimization in a general form as follows:

$$\min_u g(u) + h(u), \quad (1)$$

where u is the image to be reconstructed, $h(u)$ is the data fidelity that measures the discrepancy between u and the acquired data (often in the transformed domain), and $g(u)$ is a regularization term which imposes the prior knowledge or our preference on the solution u .

To instantiate the variational method (1), we may consider the image reconstruction problem with total-variation (TV) regularization for compressive sensing magnetic resonance imaging (CS-MRI) in the discretized form: Suppose that the gray-scale image u to be reconstructed is defined on the two-dimensional $\sqrt{n} \times \sqrt{n}$ mesh grid (thus a total of n pixels) representing its square domain $[0, 1]^2$. Then u can be interpreted as a vector in \mathbb{R}^n where its i th component $u_i \in \mathbb{R}$ is the integral (or average) of the image intensity value over the i th pixel for $i = 1, \dots, n$. MRI scanners can acquire the Fourier coefficients of u , from which one can recover u simply by applying inverse Fourier transform. For fast imaging in CS-MRI, we only acquire a fraction of Fourier coefficients $b \in \mathbb{C}^m$ with $m < n$, which relates to u by $b = P\mathcal{F}u + e$ where $\mathcal{F} \in \mathbb{C}^{n \times n}$ is the discrete Fourier transform matrix, $P \in \mathbb{R}^{m \times n}$ is a binary selection matrix (one entry as 1 and the rest as 0 in each row) indicating the indices of the sampled Fourier coefficients, and $e \in \mathbb{C}^m$ represents the unknown noise in data acquisition. Then the data fidelity term $h(u)$ in (1) can be set to $(1/2) \cdot \|P\mathcal{F}u - b\|_2^2$. For fast imaging, m is often much smaller than n and hence we need additional regularization $g(u)$ in (1) to ensure robust and stable recovery of u . TV is one of the most commonly used regularization in image reconstruction—the simplified version of TV in the discrete setting is $TV(u) = \sum_{i=1}^n \|D_i u\|_2$ where $D_i \in \mathbb{R}^{2 \times n}$ is binary and has only two nonzero entries (1 and -1) corresponding to the forward finite difference approximations to partial derivatives along the coordinate axes at pixel i . Hence the regularization can be set to $g(u) = \mu TV(u)$

for some user-chosen weight parameter $\mu > 0$ in (1). The motivation of using TV as regularization is that images with small TV tend to have distinct constant intensity values in different regions and sharp intensity change on the boundary between two regions, hence displaying the included objects with clear intensity contrasts. The minimization in (1) thus reflects the principle of the variational method for image recovery—we want to find the minimizer u such that it is consistent to the observed data (small value of $h(u)$) and meanwhile has desired regularity (small value of $g(u)$). To this point, (1) becomes an optimization problem of $u \in \mathbb{R}^n$, for which we can apply a proper numerical optimization algorithm and solve for u from (1).

The variational method yields a concise and elegant formulation of image reconstruction as in (1). It has achieved great success in image reconstruction thanks to the fast developments of numerical optimization techniques in the past decades. However, there are several main issues associated with this approach.

The first issue with (1) is the choice of regularization $g(u)$. There are numerous regularization terms proposed in the literature. Although many of them have proven robust in practice, they are often overly simplified and cannot capture the fine details in medical images which are critical in diagnosis and treatment. For example, TV regularization is known for its “staircase” effect due to its promotion of sparse gradients, such that the reconstructed images tend to be piecewise constant which are not ideal approximations to the real-world images. For example, important fine structures and minor contrast changes can be smeared in the reconstructed image using TV regularization, which is unacceptable for applications that require high image quality.

The second issue is the parameter tuning. To achieve desired balance between noise reduction and faithful structural reconstruction, the parameters of a reconstruction model (e.g., $\mu > 0$ mentioned above) and its associated optimization algorithm (such as step sizes) need to be carefully tuned. Unfortunately, the image quality is often very sensitive to these parameters; and the optimal parameters are also shown to be highly dependent on the specific acquisition settings and imaging datasets.

Last but not least, the reconstruction time of iterative optimization algorithms is also a major concern on their applications in real-world problems. Despite that the efficiency of optimization algorithms is continuously being improved, these algorithms, even for convex problems, often require hundreds of iterations or more to converge, which result in long computational time.

The issues with the classical variational methods and optimization algorithms mentioned above inspired a new class of deep learning-based approaches. Deep learning Goodfellow et al. (2016) with deep neural networks (DNNs) as the core component has achieved great success in a variety of real-world applications, including computer vision (He et al. 2016; Krizhevsky et al. 2012; Zeiler and Fergus 2014), natural language processing (Devlin et al. 1810; Hinton et al. 2012; Sarikaya et al. 2014; Socher et al. 2012; Vaswani et al. 2017), medical imaging (Hammernik et al. 2018; Schlemper et al. 2018; Sun et al. 2016), etc. DNNs have provable representation power and can be trained with little or no knowledge about the underlying functions. However, there are several major issues of such standard

deep learning approaches: (i) Generic DNNs may fail to approximate the desired functions if the training data is scarce; (ii) the training of these DNNs is prone to overfitting, noises, and outliers; and (iii) the trained DNNs are mostly “blackboxes” without rigorous mathematical justification and can be very difficult to interpret.

To mitigate the aforementioned issues of DNNs, a class of *learnable optimization algorithms* (LOAs) has been proposed recently. In brief, the architectures of the neural networks in LOAs mimic the iterative scheme of the optimization algorithms, also known of “unrolling” the optimization algorithms. More specifically, these reconstruction networks are composed of a small number of phases, where each phase mimics one iteration of a classical, optimization-based reconstruction algorithm. In most cases, the terms corresponding to the manually designed regularization in the classical methods are parameterized by multilayer perceptrons whose parameters are to be learned adaptively in the offline training process with lots of imaging data. After training, these networks work as fast feedforward mappings with extremely low computational cost, so that the reconstruction of new images can be performed on the fly. These methods combine the best parts of variational methods and deep learning for fast and adaptive image reconstruction. In the next section, we first consider the algorithms that are designed to solve a prescribed model in the form of (1). Section “[Structured Image Reconstruction Networks](#)” is dedicated to the class of deep reconstruction networks that can learn the variational model or algorithm such that the outputs are high-quality reconstructions of the images.

Learned Algorithm for Specified Optimization Problem

Learned optimization algorithms are modifications of traditional optimization algorithms by including trainable components, such as deep neural networks or the layers, for fast and adaptive numerical solution. This approach is motivated by the viewing the iterative scheme in traditional optimization algorithm (e.g., gradient descent) as a feedforward neural network with repeated, predesigned layers. The main structures of these algorithms largely adopt those of the original optimization algorithms. To make these algorithms more adaptive to the given problem, learnable components are introduced so they can improve over the original algorithms using the available data.

In this section, we showcase several learned optimization algorithms for the well-known l_1 minimization problem as follows:

$$\min_u \mu \|u\|_1 + \frac{1}{2} \|Au - b\|^2, \quad (2)$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and the parameter $\mu > 0$ are given. The solution of (2) is also known as the least absolute shrinkage and selection operator (lasso) or sparse recovery since the solution u fits the observed data b in the data fidelity term $h(u) := (1/2) \cdot \|Au - b\|^2$ and meanwhile tends to have only a small amount of nonzero

components (hence sparse) due to the l_1 regularization $g(u) := \mu \|u\|_1$. A basic method for solving (2) is called the iterative shrinkage-threshold algorithm (ISTA). To solve (2), ISTA first approximates $h(u)$ by its first-order Taylor expansion at the previous iterate $u^{(k)}$ plus a quadratic penalty term with weight $1/(2\alpha)$ in each iteration k as follows:

$$\begin{aligned} h(u) &\approx h(u^{(k)}) + \langle \nabla h(u^{(k)}), u - u^{(k)} \rangle + \frac{1}{2\alpha} \|u - u^{(k)}\|^2 \\ &= \frac{1}{2\alpha} \|u - (u^{(k)} - \alpha \nabla h(u^{(k)}))\|^2 + \text{const}, \end{aligned} \quad (3)$$

where we completed the square to obtain the equality above, and the term “const” represents a constant independent of u . As a result, ISTA generates the next iterate $u^{(k+1)}$ by

$$u^{(k+1)} = \arg \min_u \left\{ g(u) + \frac{1}{2\alpha} \|u - (u^{(k)} - \alpha \nabla h(u^{(k)}))\|^2 \right\}, \quad (4)$$

where the constant term is omitted since it does not affect the result $u^{(k+1)}$ in (4). To obtain $u^{(k+1)}$ in (4), it is essential to find the solution of the proximity operator prox_g defined below for any given $z \in \mathbb{R}^n$:

$$\text{prox}_g(z) := \arg \min_x \left\{ g(x) + \frac{1}{2} \|x - z\|^2 \right\}. \quad (5)$$

With $g(x) := \mu \|x\|_1$, the proximity operator prox_g has a closed form solution, called the shrinkage operator S_μ . That is, the i th component of $S_\mu(z) = \text{prox}_g(z) \in \mathbb{R}^n$ is

$$[S_\mu(z)]_i = [\text{prox}_g(z)]_i = \text{sign}(z_i) \cdot \max\{|z_i| - \mu, 0\}. \quad (6)$$

Therefore, $S_\mu(z)$ “shrinks” the magnitude of each component of its argument z by μ ; if the magnitude is smaller than μ , then it becomes 0 after the shrinkage. Combining (4), (5), and (6) yields the scheme of ISTA:

$$u^{(k+1)} = S_{\mu/L} \left(u^{(k)} - \frac{1}{L} A^\top (A u^{(k)} - b) \right), \quad (7)$$

where α is set to the optimal value $1/L$ in (7) and L is the largest eigenvalue of $A^\top A$ (i.e., the Lipschitz constant of $\nabla h(u) = A^\top (A u - b)$). It can be shown that, starting from any initial guess $u^{(0)}$, ISTA (7) generates a sequence $\{u^{(k)}\}$ that converges to a solution of (2) at a sublinear rate of $O(1/k)$ in function value.

However, the practical performance of ISTA is not satisfactory as it often requires hundreds to thousands of iterations to obtain an acceptable approximation to the solution. Although there are a variety of optimization techniques to improve the

convergence of ISTA, the traditional variational formulation and optimization still fall short in real-world applications due to the relatively slow convergence and the issues mentioned in section “Introduction”. Inspired by the great success of deep learning, for a fixed A , we may ask whether it is possible to learn the terms, such as μ , L , and even A^\top , in (7) adaptively if we have many instances of b and their corresponding solutions to (2). In Gregor and LeCun (2010), this approach is examined and results in the learned ISTA (LISTA) formed as a K -layer feedforward neural network:

$$u^{(k+1)} = \sigma_k(W_1^{(k)}b + W_2^{(k)}u^{(k)}) \quad (8)$$

for $k = 0, \dots, K - 1$. In LISTA (8), the linear mappings $W_1^{(k)}$, $W_2^{(k)}$ and the nonlinear mapping (can also be a preselected nonlinear activation function) σ_k can be learned, such that the final output $u^{(K)}$, as a function of these parameters $\Theta := (\dots, W_1^{(k)}, W_2^{(k)}, \sigma_k, \dots)$, is close to a solution u^* of (2) for a given b . More specifically, given N pairs of training data $\{(b_j, u_j^*) : 1 \leq j \leq N\}$, where $b_j \in \mathbb{R}^m$ is the input data of the optimization problem (2) and $u_j^* \in \mathbb{R}^n$ is the corresponding ground truth (e.g., solution obtained by solving the minimization problem (2) with b_j using some classical optimization algorithm to high accuracy), then one can learn the optimal network parameter Θ^* by solving the minimization problem

$$\min_{\Theta} \frac{1}{N} \sum_{j=1}^N \|u^{(K)}(b_j; \Theta) - u_j^*\|^2$$

where $u^{(K)}(b; \Theta)$ denotes the output of the K -phase network with parameter Θ and input data b . By training the parameter Θ with various of b and the corresponding u^* , LISTA can find an effective path from $u^{(0)}$ to $u^{(K)}$ using the learned Θ^* . If training result is satisfactory with a small K (e.g., $K = 10$), then LISTA, as a feedforward neural network, is expected to compute good approximation of u^* given new input b on the fly. Note that LISTA (8) reduces to ISTA (7) if the parameters are not learned but pre-defined as $W_1^{(k)} = A^\top/L$, $W_2^{(k)} = I - A^\top A/L$, and $\sigma_k(\cdot) = S_{\mu/L}(\cdot)$ for all k . It is shown that LISTA can achieve similar solution accuracy with iteration number K 18 to 35 times fewer than that required in ISTA or FISTA for problems with dimension 100 to 400 (Gregor and LeCun 2010).

In recent years, there have been a number of follow-up research works that exploit the properties and variations of LISTA. In Chen et al. (2018), a simplified version of LISTA is proposed:

$$u^{(k+1)} = S_{\mu/L} \left(u^{(k)} - \frac{1}{L} W^\top (A u^{(k)} - b) \right), \quad (9)$$

with learnable W , and the convergence of (9) for solving (2) is also established in Chen et al. (2018) and Liu et al. (2019). In Sprechmann et al. (2015), LISTA is extended to learnable pursuit process architectures for structured sparse and robust

low rank models derived from proximal gradient algorithm. It is shown that such network architecture can approximate the exact sparse or low rank representation at a fraction of the complexity of the standard optimization methods. In Xin et al. (2016), a learned iterative hard thresholding (IHT) algorithm where σ_k is replaced by a hard thresholding operator H_k is developed, and its potential to recover minimal l_0 norm solution is shown both theoretically and empirically. The work Borgerding et al. (2017) developed a learned approximate message passing (LAMP) algorithm for the lasso problem (2):

$$v^{(k+1)} = \beta_k v^{(k)} - Au^{(k)} + b, \quad (10a)$$

$$u^{(k+1)} = S_{\mu_k}(u^{(k)} + A^\top v^{(k+1)}). \quad (10b)$$

In contrast to LISTA, LAMP (10) includes a residual $v^{(k)}$ in each layer k , which performs shrinkage dependent on k . By the inclusion of the ‘‘Onsager correction’’ term $\beta_k v^{(k)}$ to decouple errors across layers, LAMP appears to outperform LISTA in accuracy empirically. For example, on synthetic data with Gaussian matrix A , LAMP takes 7 iteration numbers to obtain the normalized mean square error (NMSE) -34dB , whereas LISTA takes 15 iterations (Borgerding et al. 2017).

The aforementioned learned optimization algorithms are for unconstrained minimizations. Recently, the work in Xie et al. (1905) developed an algorithm, called the differentiable linearized alternating direction method of multipliers (D-LADMM), can be used to solve problems with linear equality constraints. D-LADMM is a K -layer linearized ADMM-inspired deep neural network, which is obtained by using learnable weights in the classical linearized ADMM and generalizing the proximal operator to learnable activation functions. It is proved that there exist a set of learnable parameters for D-LADMM to generate globally converged solutions.

To this point, we have seen several instances of modifying the ISTA (7) to obtain deep neural networks with trainable components to solve (2). Each iteration of ISTA is transformed into one layer of a neural network, the parameters of which are then trained using available imaging data. Once properly trained, these networks can often achieve more accurate approximations of the solution in much less time than the traditional approaches. Global convergence results, sometimes even better than the original optimization algorithms, have been established for several of these methods. However, most of these methods are restricted to the variational model (1) with l_1 or l_0 regularization, so that the proximity operators can yield closed-form shrinkage as the nonlinear activation function. It remains as an open problem on extending this type of methods to handle more general or learnable regularization.

Structured Image Reconstruction Networks

In this section, we introduce several deep neural networks inspired by classical optimization algorithms for image reconstruction. Unlike the learned algorithms discussed in section ‘‘[Learned Algorithm for Specified Optimization Problem](#)’’,

these networks aim at solving the given reconstruction problem demonstrated by training dataset (often includes ground truth images), rather than any prescribed optimization problem such as the lasso (2). As a result, they do not require manually designed regularization and specified objective function but can implicitly learn an adaptive regularization using the training data. This class of methods has become the mainstream for deep learning-based image reconstruction research in recent years.

The optimization-inspired reconstruction networks in this section also share the same main feature: each phase of these networks corresponds to one iteration of the classical optimization. More specifically, the data fidelity term h in (1) that describes the relation between image and acquired data is largely preserved as in optimization algorithms. However, unlike the methods in section “[Learned Algorithm for Specified Optimization Problem](#)”, the regularization term g is unknown but can be replaced by neural networks whose parameters are learned adaptively from data.

In the remainder of this section, we introduce several reconstruction neural networks developed along this line. Most of these networks can be applied to a wide range of image reconstruction problems as they are customized to learn from the training data directly rather than for any specific imaging application or modality. The training process can be time-consuming but is performed offline. Once trained properly, however, they serve as fast feedforward mappings that reconstruct high-quality images of the same type as those in the training dataset.

Proximal Point Network

A group of deep neural networks inspired by variational methods and optimization algorithms directly leverage the popular deep neural network structures into the optimization schemes. Considering the variational model (1) with general g and h , we can rewrite its proximal point algorithm (4) as an equivalent two-step scheme by introducing an auxiliary variable $r^{(k)} = u^{(k-1)} - \alpha \nabla h(u^{(k-1)})$ and using the definition of the proximity operator in (5):

$$r^{(k)} = u^{(k-1)} - \alpha \nabla h(u^{(k-1)}), \quad (11a)$$

$$u^{(k)} = \text{prox}_{\alpha g}(r^{(k)}). \quad (11b)$$

As the data fidelity h is formulated based on the definitive relation between image and acquired data, such as $h(u) = (1/2) \cdot \|P\mathcal{F}u - b\|^2$ in CS-MRI as shown in section “[Introduction](#)”, it is often kept unmodified in (11a). Moreover, the step size α can be set to α_k which is not manually chosen but learned during the training process. On the other hand, the proximal term in (11b) is due to the regularization g and performs as an image “denoiser” that modifies inputs $r^{(k)}$ to obtain an improved image $u^{(k)}$. Instead of choosing regularization g manually and solving (11b) in each iteration, we can directly parametrize its proximity operator $\text{prox}_{\alpha g}$ as a learnable denoiser parametrized as convolutional neural network (CNN) (Goodfellow et al. 2016). Moreover, we can use the residual network (ResNet) structure proposed in

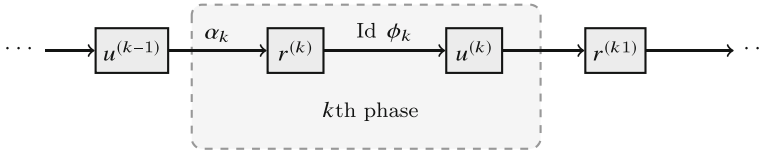


Fig. 1 Architecture of the proximal point network (11a) and (12). The k th phase updates $r^{(k)}$ and $u^{(k)}$. The dependencies of each variable on other variables are shown as incoming arrows, and the network parameters used for update are labeled next to the corresponding arrows

He et al. (2016) for the CNN which proves to be more effective for reducing training error in imaging applications. Namely, we replace the proximity operator $\text{prox}_{\alpha g}$ in (11b) by a denoising network (Zhang et al. 2017):

$$u^{(k)} = r^{(k)} + \phi_k(r^{(k)}) \quad (12)$$

where ϕ_k is a standard multiplayer CNN that maps $r^{(k)}$ to the residual between $u^{(k)}$ and $r^{(k)}$. The architecture of the proximal point network given by (11a) and (12) is illustrated in Fig. 1, where each arrow indicates a mapping from its input to the output with the required network parameters labeled next to it.

Let Θ denote the collection of learnable parameters in ϕ_k (e.g., the convolutional kernels and the biases) and algorithm parameters (e.g., $\alpha_k > 0$) for all $k = 1, \dots, K$, and then the output after K cycles (phases) of (11a) and (12) is a function of Θ for any given imaging data b . Denote this output by $u^{(K)}(b; \Theta)$, which is the output of any given image data b passing through this network with parameter Θ ; we can form the loss function of Θ by regression as:

$$L(\Theta; b, u^*) = \frac{1}{2} \|u^{(K)}(b; \Theta) - u^*\|^2, \quad (13)$$

where u^* is the ground truth image corresponding to the (possibly noisy and incomplete) imaging data b , both given in the training data. By feeding in a large amount of instances of form (b, u^*) , we can solve for the minimizer Θ^* of the sum of L as in (13) over all of these instances. Then the deep reconstruction network with K phases, each consisting of (11a) and (12), is a feedforward neural network with parameters Θ^* for fast image reconstruction given any new coming data b .

The proximal point network can be applied to a variety of imaging applications, including image denoising, image deblurring, and image super-resolution by replacing the proximal operator by a denoiser network in regularization subproblem of half-quadratic splitting algorithm (Zhang et al. 2017). In Zhang et al. (2017), ϕ_k is designed to contain 7 dilated convolutions with 64 feature maps in each middle layer, where ReLU activation function is used after the first convolution, and both batch normalization (BN) and ReLU are used in every convolution thereafter. The training data is composed of 256×4000 image patches of size 35×35 cropped from the BSD400 (Martin et al. 2001), 400 images from ImageNet validation set

(Deng et al. 2009), and 4,744 Waterloo Exploration images (Ma et al. 2016). They evaluate their results on BSD68 (Roth and Black 2009), Set5, and Set14 (Timofte et al. 2014), respectively. In Zhang and Ghanem (2018), IRCNN is compared with several other methods on Set11 (Kulkarni et al. 2016) with various sampling ratios, and the results will be presented later in this section.

The work developed in Cheng et al. (2019), Chun et al. (2019), Meinhardt et al. (2017), Rick Chang et al. (2017), Wang et al. (2016), and Zhang et al. (2017) can all be considered as variations of the method described above. For instance, CNN denoiser has been placed in the proximal gradient descent algorithm in Meinhardt et al. (2017), subproblem in half-quadratic splitting in Zhang et al. (2017), subproblem in ADMM in Meinhardt et al. (2017) and Rick Chang et al. (2017), and subproblems in primal-dual algorithm in Cheng et al. (2019), Meinhardt et al. (2017), and Wang et al. (2016).

ISTA-Net

ISTA-Net Zhang and Ghanem (2018) is a deep neural network architecture for image reconstruction inspired by ISTA as given in (7). Recall that ISTA is originally derived to solve the l_1 minimization problem (2), i.e., (1) with $g(u) = \mu\|u\|_1$ and $h(u) = (1/2) \cdot \|Au - b\|^2$, as we showed in section “[Learned Algorithm for Specified Optimization Problem](#)”. For image reconstruction, the sole l_1 norm is not a suitable regularization since almost all natural images are not sparse themselves. Instead, they are often sparse in certain transform domains. Let $\Psi \in \mathbb{R}^{n \times n}$ be a *sparsifying* operator (e.g., wavelet transform) that transforms u into a sparse vector Ψu . Then, we can modify lasso (2) and obtain a similar form as:

$$\min_u g(\Psi u) + h(u) . \quad (14)$$

Although (14) does not exactly match the ISTA (2) due to the presence of Ψ , this can be easily resolved by using an orthogonal sparsifying operator Ψ and setting $x = \Psi u$ as the unknown for (2). For example, if we set Ψ to an orthogonal 2D wavelet transform. In this case, we just need to solve x from the exact form of (2) with $g(x) = \mu\|x\|_1$ and $\tilde{h}(x) := h(\Psi^\top x)$ as the data fidelity, and recover $u = \Psi^\top x$ using the output x of ISTA. Integrating this change of variables into the scheme (11), we obtain a slightly modified version of ISTA as follows:

$$r^{(k)} = u^{(k-1)} - \alpha \nabla h(u^{(k-1)}), \quad (15a)$$

$$u^{(k)} = \Psi^\top \text{prox}_{\alpha g}(\Psi r^{(k)}) = \Psi^\top S_\theta(\Psi r^{(k)}), \quad (15b)$$

where $\theta = \alpha\mu$ combines the two parameters, and (15b) involves shrinkage due to the choice of $g(x) = \mu\|x\|_1$. The gradient ∇h in (15a) is due to the data fidelity h in (14). Therefore, we do not need to “learn” this part in the reconstruction. On the other hand, the use of the sparsifying transform Ψ and ℓ_1 regularization is rather

heuristic. If there is sufficient amount of training data, it is likely that we can learn a better representation of this regularization using a deep learning technique.

Bearing this idea, ISTA-Net is proposed to replace the transform Ψ and Ψ^\top in (15) by multilayer convolutional neural networks (CNN), while keeping the $\text{prox}_{\alpha g}$, i.e., the shrinkage due to the ℓ_1 norm, as it seems robust in suppressing noises. To this end, ISTA-Net follows the scheme of ISTA (15) and constructs a deep neural network of a prescribed K phases as in section “[Proximal Point Network](#)”.

Unlike LISTA and its variations in section “[Learned Algorithm for Specified Optimization Problem](#)”, the k th phase of ISTA-Net is to mimic the two steps in the k th iteration of ISTA in (15). Given the output $u^{(k-1)}$ of the previous phase, the update of $r^{(k)}$ follows (15a) directly since h is known to accurately describe the data formation. Therefore, only the parameter α in (15a), which behaves as the step size in ISTA, is set to α_k and is to be learned during the training process in ISTA-Net. After $r^{(k)}$ is updated, it is passed to (15b) with Ψ and Ψ^\top replaced by two multilayer CNNs $H^{(k)}$ and $\tilde{H}^{(k)}$, respectively, and the shrinkage parameter θ is replaced by θ_k , which is to be learned as well. Namely, $u^{(k)}$ is updated by

$$u^{(k)} = \tilde{H}^{(k)}(S_{\theta_k}(H^{(k)}(r^{(k)}))). \quad (16)$$

In ISTA-Net Zhang and Ghanem (2018), $H^{(k)}$ and $\tilde{H}^{(k)}$ are set to simple two-layer CNNs as follows:

$$H^{(k)}(r) = w_2^{(k)} * \sigma(w_1^{(k)} * r^{(k)}) \quad \text{and} \quad \tilde{H}^{(k)}(\tilde{r}) = \tilde{w}_2^{(k)} * \sigma(\tilde{w}_1^{(k)} * \tilde{r}^{(k)}) \quad (17)$$

where $w_1^{(k)}$, $w_2^{(k)}$, $\tilde{w}_1^{(k)}$, and $\tilde{w}_2^{(k)}$ are convolutional kernels in the k th phase to be learned, and σ is a component-wise activation function such as ReLU, i.e., $\sigma(x) = \max(x, 0)$ component wisely. In the numerical implementation of ISTA-Net Zhang and Ghanem (2018), w_1 and \tilde{w}_2 are convolutions with d kernels of size 3×3 ; w_2 and \tilde{w}_1 are convolutions with d kernels of size $3 \times 3 \times d$ with d set to 32.

To this point, we can see that ISTA-Net is a deep neural network with a prescribed number of K phases. Each phase of ISTA-Net mimics one iteration (15) of ISTA and is formed as: $r^{(k)}$ and $u^{(k)}$ by

$$r^{(k)} = u^{(k-1)} - \alpha_k \nabla h(u^{(k-1)}), \quad (18a)$$

$$u^{(k)} = \tilde{H}^{(k)} S_{\theta_k}(H^{(k)} r^{(k)}), \quad (18b)$$

where we have omitted excessive parentheses for notation simplicity, i.e., $H^{(k)} r^{(k)}$ stands for $H^{(k)}(r^{(k)})$, etc. The K phases are concatenated in order, where the k th phase accepts the output $u^{(k-1)}$ of the previous phase, updates $r^{(k)}$ using (18a) with α_k , and finally outputs $u^{(k)}$ using (18b). Hence, the parameters to be learned are α_k , θ_k , and $w_1^{(k)}$, $w_2^{(k)}$ in $H^{(k)}$ and $\tilde{w}_1^{(k)}$ and $\tilde{w}_2^{(k)}$ in $\tilde{H}^{(k)}$ for $k = 1, 2, \dots, K$. In the first phase, the input is the initial guess $u^{(0)}$, which can be set to $A^\top b$. The output of the last phase, $u^{(K)}$, is used in the loss function that measures its squared discrepancy

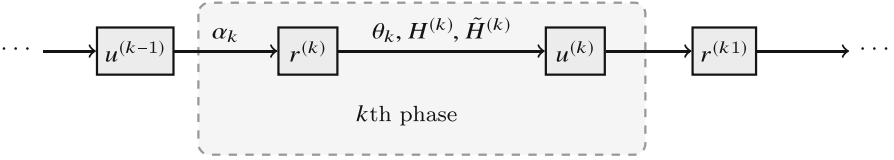


Fig. 2 Architecture of ISTA-Net (18). The k th phase updates $r^{(k)}$ and $u^{(k)}$. The dependencies of each variable on other variables are shown as incoming arrows, and the network parameters used for update are labeled next to the corresponding arrows

to the corresponding ground truth, high-quality image u^* :

$$L_{\text{dis}}(\Theta; b, u^*) = \frac{1}{2} \|u^{(K)}(b; \Theta) - u^*\|^2 \tag{19}$$

where (b, u^*) is a training pair as in the proximal point network in section “Proximal Point Network”, and $\Theta := \{\alpha_k, \theta_k, w_1^{(k)}, w_2^{(k)}, \tilde{w}_1^{(k)}, \tilde{w}_2^{(k)} \mid k = 1, \dots, K\}$. The structure of the ISTA-Net can be visualized in Fig. 2. For more details of the network structure and its relation to the back-propagation procedure, we refer to Wang et al. (2019).

In addition, since $H^{(k)}$ and $\tilde{H}^{(k)}$ in (17) are replacing Ψ and Ψ^\top , respectively, they are expected to satisfy $\tilde{H}^{(k)} H^{(k)} = I$, the identity mapping. To make this constraint approximately satisfied, the mismatch between $\tilde{H}^{(k)}(H^{(k)}(u^*))$ and u^* can be integrated into the following loss function, despite that it is much weaker than $\tilde{H}^{(k)} H^{(k)} = I$:

$$L_{\text{id}}(\Theta; u^*) = \frac{1}{2} \sum_{k=1}^K \|\tilde{H}^{(k)}(H^{(k)}(u^*)) - u^*\|^2. \tag{20}$$

The loss function for a particular training pair (b, u^*) is thus the sum of the losses in (19) and (20) with a balancing parameter $\gamma > 0$:

$$L(\Theta; b, u^*) = L_{\text{dis}}(\Theta; b, u^*) + \gamma L_{\text{id}}(\Theta; u^*), \tag{21}$$

and the total loss function during training is the sum of $L(\Theta; b, u^*)$ in (21) over all training pairs of form (b, u^*) in the training dataset.

The optimal parameter Θ^* can be obtained by minimizing the loss function (21), which can be accomplished using the stochastic gradient descent (SGD) method. The key in the implementation of SGD is the computation of the gradient of (21) with respect to each network parameter, i.e., $\alpha_k, \theta_k, w_1^{(k)}, w_2^{(k)}, \tilde{w}_1^{(k)}, \tilde{w}_2^{(k)}$ for $k = 1, \dots, K$. More specifically, we first need to compute the gradient of L defined in (21) with respect to the main variables $u^{(k)}$ and $r^{(k)}$. Then we compute the gradients of $u^{(k)}$ with respect to its parameters, i.e., $\theta_k, w_1^{(k)}, w_2^{(k)}, \tilde{w}_1^{(k)}, \tilde{w}_2^{(k)}$, and the gradient of $r^{(k)}$ with respect to $\alpha^{(k)}$. Finally, the gradients of L with respect to these network

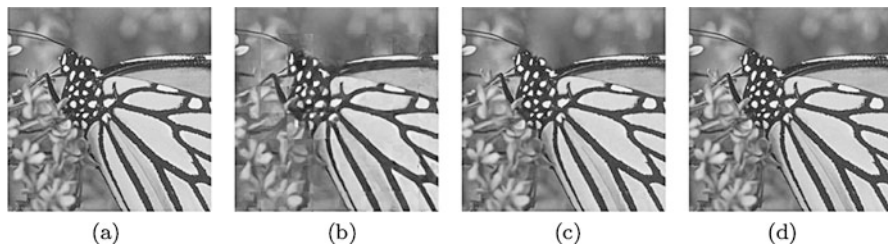


Fig. 3 Qualitative reconstruction results of ISTA-Net⁺ (Zhang and Ghanem 2018) applied to the Butterfly image in Set11 (Kulkarni et al. 2016) with various sampling ratios. The numbers in the captions of (b)–(d) are the corresponding sampling ratios, and PSNR are shown in the parentheses. Results are generated by the code available at <https://github.com/jianzhangcs/ISTA-Net>. (a) True (b) 10% (25.91) (c) 25% (33.52) (d) 50% (40.18)

parameters can be built by multiplying the involved partial derivatives according to the chain rule. The derivations are fairly straightforward. For completeness, we provided the details of this back-propagation in the Appendix.

ISTA-Net (Zhang and Ghanem 2018) evaluated the reconstruction results on datasets BSD68 (Martin et al. 2001) and Set11 (Kulkarni et al. 2016), respectively. The training set contains $N = 88,912$ pairs (b, u^*) , where u^* is 33×33 image patch randomly cropped from the images in 91Images dataset (Kulkarni et al. 2016) and b is the corresponding CS measurement. In Table 1, the reconstructed results are shown and compared with a traditional variational method TVAL3 (Li et al. 2013) and a non-iterative network IRCNN (Zhang et al. 2017), where the ISTA-Net⁺ is the residual shortcut enhanced version ISTA-Net; for the detailed implementation of ISTA-Net⁺, please refer to Zhang and Ghanem (2018). Some reconstructed images of Butterfly in Set11 (Kulkarni et al. 2016) by ISTA-Net⁺ with various sampling ratios are displayed in Fig. 3.

ADMM-Net

ADMM-Net (Sun et al. 2016) is one of the earliest attempts to unroll a known optimization algorithm into a deep neural network. ADMM-Net is originated from the alternating minimization method of multipliers, or ADMM for short, which is a numerical algorithm particularly effective for convex optimization problems with linear equality constraints. Combined with the variable splitting technique, ADMM has been very popular and successful in solving variety of nonsmooth and/or constrained problems.

In its standard form, ADMM can solve constrained convex problems where the primal variable (i.e., the variable to be solved in the optimization problem) consists of two blocks related by a linear equality constraint. In addition, there is a dual variable, i.e., the Lagrangian multiplier, associated with the equality constraint. In each iteration, ADMM updates the two blocks of the primal variables in order, one

at each time with the other one fixed and then the dual variable using the updated primal variable. ADMM yields more complex iterations due to the multiple-variable structure than ISTA.

We first recall the variable splitting and the original ADMM for image reconstruction problem, which is formulated as the one in ISTA as (14):

$$\min_u g(\Psi u) + \frac{1}{2} \|Au - b\|^2, \quad (22)$$

but with more specific data fidelity $h(u) = (1/2) \cdot \|Au - b\|^2$. Here, we write the regularization in (22) as a composite function where g is simple (i.e., the proximity operator prox_g has closed form or is easy to compute) and Ψ as a linear operator. A typical example is the total variation regularization we mentioned in section “Introduction”: $g(\Psi u) := \mu \sum_{i=1}^n \|D_i u\|_2$ with weight parameter $\mu > 0$. That is, Ψ is the discrete gradient operator (finite forward differences) D , and g is a slight variation of l_1 norm which takes sum of the l_2 norms of the gradients at all pixels. For ADMM to work efficiently, there is also requirement on the matrices Ψ and A , which we will specify later. To apply ADMM, we first use variable splitting by introducing an auxiliary variable w such that $w = Du$ and rewrite (22) as the following equivalent problem:

$$\min_{w,u} \left\{ g(w) + \frac{1}{2} \|Au - b\|^2 \right\}, \text{ subject to } w = Du. \quad (23)$$

Then, we formulate its associated augmented Lagrangian:

$$L(u, w; \lambda) = g(w) + \frac{1}{2} \|Au - b\|^2 + \langle \lambda, w - Du \rangle + \frac{\rho}{2} \|w - Du\|^2, \quad (24)$$

with Lagrangian multiplier λ . ADMM is then applied to solve (23) with the augmented Lagrangian (24). In each iteration of ADMM, the primal variables w and u are updated in order, and then the dual variable λ is updated. In the case of CS-MRI with $A = P\mathcal{F}$ mentioned in section “Introduction”, the subproblems are given as follows:

$$w^{(k)} = S_\theta(Du^{(k-1)} - \lambda^{(k-1)}), \quad (25a)$$

$$u^{(k)} = (\rho D^\top D + A^\top A)^{-1}(A^\top b + \rho D^\top w^{(k)} - D^\top \lambda^{(k-1)}), \quad (25b)$$

$$\lambda^{(k)} = \lambda^{(k-1)} + \rho(w^{(k)} - Du^{(k)}), \quad (25c)$$

where $\theta = \mu/\rho$. Given an initial guess $(w^{(0)}, u^{(0)}, \lambda^{(0)})$, ADMM repeats the cycle of the three steps (25) for iteration $k = 1, 2, \dots$, until a stopping criterion is satisfied. As we can see, for ADMM to work efficiently, the inverse of $D^\top D + \rho A^\top A$ in (25b) must be easy to compute. In certain imaging applications, this is

possible since both $D^\top D$ and $A^\top A$ can be diagonalized by fast transforms (such as Fourier), with which the update $u^{(k)}$ (25b) requires very low computational cost.

ADMM-Net (Sun et al. 2016) is a deep reconstruction network architecture that mimics the ADMM scheme (25). Similar to the case of ISTA-Net, each phase of ADMM-Net mimics one iteration of ADMM (25). More specifically, ADMM-Net sets a fixed iteration number K . The k th phase of ADMM-Net mimics the k th iteration of ADMM (25), but ADMM-Net replaces the gradient operator D by a parameterized filter (convolution) $H^{(k)}$ and the fixed parameters θ and ρ by θ_k and ρ_k to be learned through training. The original ADMM-Net (Sun et al. 2016) is designed to solve the single-coil CS-MRI problem with $A = P\mathcal{F}$, for which the k th phase of ADMM-Net reduces to:

$$w^{(k)} = S_{\theta_k}(H^{(k)}u^{(k-1)} - \lambda^{(k-1)}), \quad (26a)$$

$$u^{(k)} = \mathcal{F}^\top (P^\top P + \rho_k \mathcal{F}H^{(k)\top} H^{(k)} \mathcal{F}^\top)^{-1} (P^\top b + \rho_k \mathcal{F}H^{(k)\top} (w^{(k)} + \lambda^{(k-1)})), \quad (26b)$$

$$\lambda^{(k)} = \lambda^{(k-1)} + (w^{(k)} - H^{(k)}u^{(k)}), \quad (26c)$$

where S_θ is the shrinkage by $\theta > 0$ as in (18b).

In ADMM-Net (Sun et al. 2016), $H^{(k)}$ is set to a linear combination of a set of given filters $\{B_l\}$ with coefficients $\gamma^{(k)} = (\dots, \gamma_l^{(k)}, \dots) \in \mathbb{R}^{|B_l|}$, i.e., $H^{(k)} = \sum_l \gamma_l^{(k)} B_l$. Therefore, $H^{(k)}$ is completely determined by the coefficients $\gamma^{(k)}$ in the k th phase. Moreover, the shrinkage in (25a) is replaced by a piecewise linear function (PLF) determined by a set of control points and the associated function values. More specifically, let $\{p_0, \dots, p_{N_c}\}$ be a set of $N_c + 1$ control points on \mathbb{R} . In Sun et al. (2016), these control points are simply chosen as uniform mesh grid points on the interval $[-1, 1]$, i.e., $p_0 = -1$ and $p_{N_c} = 1$, and $p_l - p_{l-1} = 2/N_c$ for $l = 1, \dots, N_c$. Then, the PLF $S(h; \{p_l, q_l^{(k)}\})$ in $[-1, 1]$ is completely determined by the values $\{q_l^{(k)}\}$ at the corresponding control points $\{p_l\}$. Outside the interval $[-1, 1]$, the PLF $S(h; \{p_l, q_l^{(k)}\})$ is set to have slope 1 and concatenates with its part in $[-1, 1]$ at p_0 and p_{N_c} to form a continuous function. Then, instead of learning θ_k in the shrinkage operation S_{θ_k} in (25a), the original ADMM-Net learns the values $\{q_l^{(k)}\}$ as a part of the network parameters. The output $u^{(K)}$ is a function of the input b and network parameters $\Theta = \{\theta_k, \rho_k, \gamma^{(k)} \mid k = 1, \dots, K\}$. The architecture of ADMM-Net is shown in Fig. 4. As usual, the loss function can be set to the squared error of $u^{(K)}$ from the ground truth, reference image u^* corresponding to data b :

$$L(\Theta; b, u^*) = \frac{1}{2} \|u^{(K)}(b; \Theta) - u^*\|^2. \quad (27)$$

The total loss function is the sum of the loss in (27) above over all training pairs (b, u^*) in the given training dataset. Then, the total loss function is minimized using the (stochastic) gradient descent method, and the minimizer Θ^* is the learned

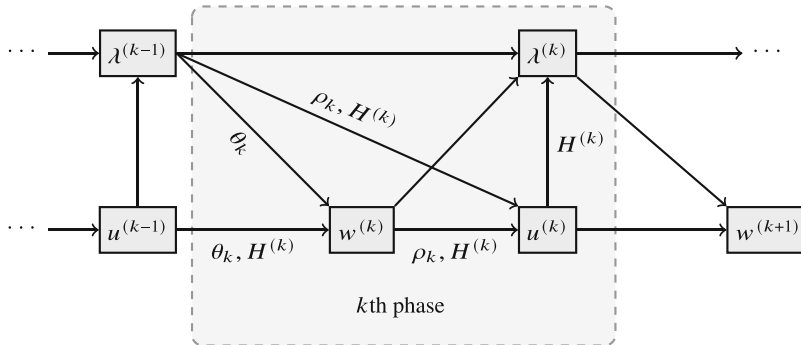


Fig. 4 Architecture of ADMM-Net (26). The k th phase updates $w^{(k)}$, $u^{(k)}$, and $\lambda^{(k)}$. The dependencies of each variable on other variables are shown as incoming arrows, and the network parameters used for update are labeled next to the corresponding arrows

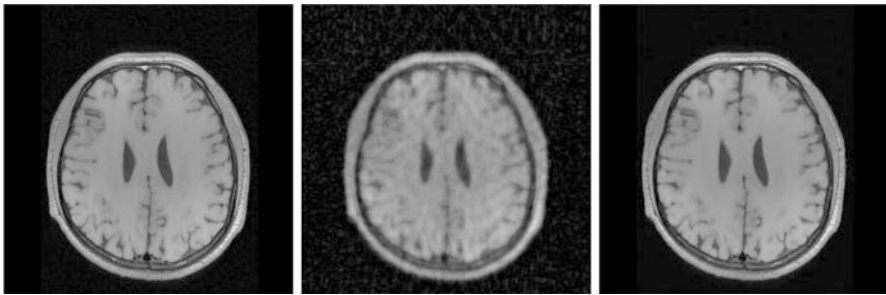


Fig. 5 Brain MR image reconstruction by ADMM-Net (Sun et al. 2016) with sampling ratio 20%. Left: ground truth. Middle: image reconstructed by zero filling. Right: reconstructed image by ADMM-Net. Results are generated by the code available at <https://github.com/yangyan92/Deep-ADMM-Net>

network parameters. More details about the derivation of the back-propagation and its relation to the network structure in Fig. 4 are provided in Wang et al. (2019). In Sun et al. (2016), ADMM-Net is applied to brain and chest MR image reconstruction, where the training and testing datasets are 100 and 50 images, respectively, randomly picked from MRI dataset (Bennett 2013). The qualitative results of a selected brain MR images reconstructed by ADMM-Net with CS ratio 20% are presented in Fig. 5.

Variational Network

As we have seen above, the proximal point network, ISTA-Net, and ADMM-Net all aim to solve the variational model of form:

$$\min_u f(u), \quad \text{where } f(u) := g(Du) + \lambda h(u), \quad (28)$$

where g , D , and even h can be learned from the training data adaptively. If we apply the well-known gradient descent method in numerical optimization to (28), we obtain:

$$u^{(k)} = u^{(k-1)} - \alpha_k (D^\top \nabla g(Du^{(k-1)}) + \lambda \nabla h(u^{(k-1)})) \quad (29)$$

where α_k is the step size in iteration k . Note that above we adopted a slight abuse of notation ∇g , since in image reconstruction g often represents the ℓ_1 norm or alike which is not differentiable. Hence, it is more rigorous to interpret ∇g as a subgradient of g , and the updating rule (29) is the subgradient descent. Nevertheless, this term will be replaced by a parameterized function to be learned in training, and thus its differentiability is not an important issue in the following derivation of the variational reconstruction network.

The variational network (Hammernik et al. 2018) was inspired by this concise updating rule (29). In Hammernik et al. (2018), the variational network is a fixed number of K phases, and each phase mimics one iteration of (29). The k th phase of variational network is built as

$$u^{(k)} = u^{(k-1)} - H^{(k)\top} \phi_k(H^{(k)}u^{(k-1)}) - \lambda_k \nabla h(u^{(k-1)}), \quad (30)$$

Here λ_k , $H^{(k)}$, and ϕ_k are all to be learned from data. The step size α_k is omitted since it is absorbed by the learnable terms. In particular, $H^{(k)}$ is a convolution to replace the manually chosen linear operator D (e.g., gradient in traditional image reconstruction) in (29), and ϕ_k is a parameterized function to replace ∇g .

In Hammernik et al. (2018), ϕ_k in (30) is represented as a linear combination of Gaussian functions. First of all, ϕ_k is to be applied to $H^{(k)}u^{(k-1)} \in \mathbb{R}^n$ component wisely, and hence it is sufficient to describe the component-wise operation of ϕ_k using a univariate function. To this end, we first determine a set of $N_c + 1$ control points $\{p_l : l = 0, \dots, N_c\}$ uniformly spaced on a prescribed interval $[-I, I]$ such that $-I = p_0 < p_1 < \dots < p_{N_c} = I$ and $p_l - p_{l-1} = 2I/N_c$ for $l = 1, \dots, N_c$. For each point p_l , the Gaussian function with a prescribed standard deviation σ is given by

$$B_l(x) = e^{-(x-p_l)^2/(2\sigma^2)}. \quad (31)$$

Then, ϕ_k is set to a linear combination of $B_l(x)$ with coefficients $\gamma_l^{(k)}$ to be determined:

$$\phi_k(x) = \sum_{l=0}^{N_c} \gamma_l^{(k)} B_l(x). \quad (32)$$

One can also design other basis functions, instead of (31) or even parametrize ϕ_k as a generic neural network. For $H^{(k)}$, it is a convolution operation applied to $u^{(k-1)}$,

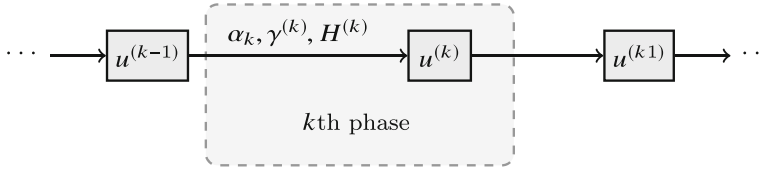


Fig. 6 Architecture of the variational network (30). The k th phase updates $u^{(k)}$. The dependencies of each variable on other variables are shown as incoming arrows, and the network parameters used for update are labeled next to the corresponding arrows

and hence it suffices to determine the convolution kernel. This is a very simplified case of convolution layers of CNNs, and we omit the details here.

Now we can see that the variational network consists of K phases, where each phase operates as (30). In particular, the first phase accepts $u^{(0)}$ as the input such as $A^\top b$. The last K th phase outputs $u^{(K)}$, which is used in the loss function to compare with the reference image u^* :

$$L(\Theta; b, u^*) = \frac{1}{2} \|u^{(K)}(b; \Theta) - u^*\|^2. \tag{33}$$

where the network parameter $\Theta := \{\alpha_k, \gamma^{(k)}, H^{(k)} \mid k = 1, \dots, K\}$. The total loss function is then the sum of (33) over all training pairs of form (b, u^*) . The architecture of variational network is presented in Fig. 6. More details about the derivation of the back-propagation and its relation to the network structure in Fig. 6 are provided in Wang et al. (2019). Similar to the proximal point network and ISTA-Net introduced above, the variational network can be applied to problems where the data fidelity term h is differentiable with Lipschitz continuous gradient.

In Hammernik et al. (2018), the variational network considered above is applied to parallel imaging MR image reconstruction. In their experiment, $H^{(k)}$ is implemented as 48 real/imaginary filter pairs and N_c is prescribed to be 31. The network is trained on the dataset which contains 20 image slices from 10 patients and tested on reconstructing the whole image volume for 10 clinical patients that is non-overlapping with training set. The qualitative illustration of a reconstructed scan of variational network is visualized in Fig. 7.

Primal-Dual Network

Primal-dual network (PD-Net) is a deep neural network architecture for image reconstruction inspired by the primal-dual hybrid gradient algorithm (Chambolle and Pock 2011). There have been a number of work that developed PD-Nets and applied to image reconstruction (Adler and Öktem 2018; Cheng et al. 2019; Heide et al. 2014; Meinhardt et al. 2017).



Fig. 7 The reconstruction result of an exemplified MR image by variational network (Hammernik et al. 2018) with sampling ratio 31.60. Results are generated by the code available at <https://github.com/VLOGroup/mri-variationalnetwork>. (a) Mask (b) Reference (c) VN (d) Error

As we discussed above, in the image reconstruction context, the variational models (1) are often represented with $g(u)$ as a regularization function and $\tilde{h}(u) = h(Au) := (1/2) \cdot \|Au - b\|^2$. In this case, we can rewrite (1) as an equivalent min-max problem by Fenchel transformation:

$$\min_u \max_{z,y} \langle Au, z \rangle - h^*(z) + \langle u, y \rangle - g^*(y) \quad (34)$$

where $h^*(z)$ and $g^*(y)$ are the conjugates (Fenchel dual) of $h(Au)$ and $g(u)$, respectively. Due to the Moreau's decomposition theorem:

$$\text{prox}_{\tau f^*}(b) = b - \tau \text{prox}_{\tau^{-1} f}(b/\tau) \quad (35)$$

for any $b \in \mathbb{R}^n$, $\tau > 0$, and convex function f , one can obtain the following iterative scheme by applying the primal-dual gradient algorithm to (34):

$$\begin{aligned} z^{(k+1)} &= \arg \min_z \left\{ -\langle A\bar{u}^k, z \rangle + h^*(z) + \frac{1}{2\gamma} \|z - z^k\|^2 \right\} \\ &= \text{prox}_{\gamma h^*}(z^k + \gamma A\bar{u}^k) = z^k + \gamma A\bar{u}^k - \gamma \text{prox}_{\gamma^{-1} h}\left(\frac{1}{\gamma} z^k + A\bar{u}^k\right) \end{aligned} \quad (36a)$$

$$\begin{aligned} y^{(k+1)} &= \arg \min_y \left\{ -\langle \bar{u}^k, y \rangle + g^*(y) + \frac{1}{2\gamma} \|y - y^k\|^2 \right\} \\ &= \text{prox}_{\gamma g^*}(y^k + \gamma \bar{u}^k) = y^k + \gamma \bar{u}^k - \gamma \text{prox}_{\gamma^{-1} g}\left(\frac{1}{\gamma} y^k + \bar{u}^k\right) \end{aligned} \quad (36b)$$

$$\begin{aligned} u^{(k+1)} &= \arg \min_u \left\{ \langle Au, z^{(k+1)} \rangle + \langle u, y^{(k+1)} \rangle + \frac{1}{2\tau} \|u - u^{(k)}\|^2 \right\} \\ &= u^k - \tau A^\top z^{(k+1)} - \tau y^{(k+1)} \end{aligned} \quad (36c)$$

$$\bar{u}^{(k+1)} = u^{(k+1)} + \theta(u^{(k+1)} - u^{(k)}) \quad (36d)$$

Similar to the deep reconstruction networks introduced above, PD-Net also mimics the primal-dual algorithm above to construct K phases such that the k th phase in PD-Net corresponds to the k th iteration in (36). Then the proximity operator $\text{prox}_{\gamma^{-1}h}$ and $\text{prox}_{\gamma^{-1}g}$ in the updates (36a) and (36b) are replaced by CNN denoisers as in section “[Proximal Point Network](#)”. The PD-Nets have been applied to natural image reconstruction in Meinhardt et al. (2017) and MRI compressive sensing in Adler and Öktem (2018); Cheng et al. (2019), which demonstrate promising performance in these applications.

Depending on which terms are designed to be learnable, three variants of the PD-Net architecture are provided in Cheng et al. (2019), which are PDHG-CSNet, CP-Net and PD-Net as follows. (i) The primal-dual hybrid gradient CS network (PDHG-CSNet) substitutes $\text{prox}_{\tau g}$ with a learned CNN denoiser in Chambolle-Pock algorithm (Chambolle and Pock 2011) which solves the (1) with $\tilde{h}(u) = h(Au) := (1/2) \cdot \|Au - b\|^2$ by iterating

$$z^{(k+1)} = \frac{z^{(k)} + \sigma(A\bar{u}^{(k)} - b)}{1 + \sigma}, \quad (37a)$$

$$u^{(k+1)} = \text{prox}_{\tau g}(u^{(k)} - \tau A^* z^{(k+1)}), \quad (37b)$$

$$\bar{u}^{(k+1)} = u^{(k+1)} + \theta(u^{(k+1)} - u^{(k)}), \quad (37c)$$

where σ , τ , and θ are algorithm parameters. (ii) The Chambolle-Pock network (CP-Net) learns a generalized Chambolle-Pock algorithm with the data fidelity term $(1/2) \cdot \|Au - b\|^2$ relaxed to $h(Au)$. Then the updating scheme of $z^{(k+1)}$ becomes $z^{(k+1)} = \text{prox}_{\sigma h^*}(z^{(k)} + \sigma A\bar{u}^{(k)})$ and CP-Net learns both $\text{prox}_{\tau g}$ and $\text{prox}_{\sigma h^*}$ with CNN denoisers. (iii) By breaking the linear combination parts in above iterates for $z^{(k+1)}$, $u^{(k+1)}$, and $\bar{u}^{(k+1)}$ in CP-Net, primal-dual net (PD-Net) further increases the network flexibility by freely learning those combinations in addition to the learnable proximal operators. In Cheng et al. (2019), the primal or dual proximal operators are substituted by learned CNN denoisers with 3 convolutional layers and 32 channels in each hidden layer. All these networks are trained and tested on 1400 and 200 images of size 256×256 and the corresponding k-space data undersampled by Poisson disk sampling mask. The qualitative reconstruction results of these three variations of the network on MR images are shown in Fig. 8, which are obtained from (Cheng et al. 2019).

Learnable Descent Algorithm

The LOAs conducted in the supervised learning framework are motivated by a disciplined bilevel optimization problem as follows:

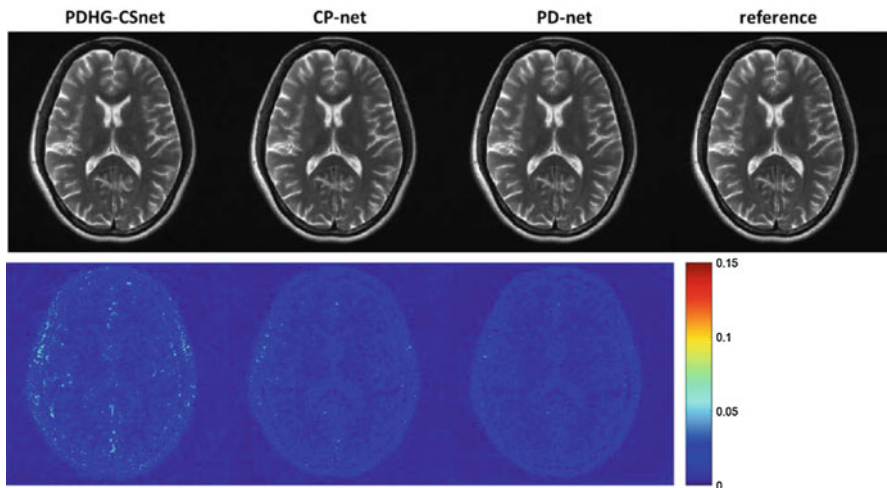


Fig. 8 Images reconstructed by primal-dual hybrid gradient CS network (PDHG-CSNet), Chambolle-Pock algorithm-inspired network (CP-Net), and primal dual net (PD-Net). The data was undersampled with a 6X Poisson disk mask

$$\min_{\Theta} \frac{1}{N} \sum_{j=1}^N \mathcal{L}(u(b_j; \Theta), u_j^*) + R(\Theta), \quad (38a)$$

$$\text{s.t. } u(b_j; \Theta) = \arg \min_{u \in \mathcal{U}} \{f(u; b_j, \Theta) := g(u; \Theta) + h(u; b_j, \Theta)\} \quad (38b)$$

where h is the data fidelity term to ensure that the reconstructed image u is faithful to the given data b , and g is the regularization that may incorporate proper prior information of u . The regularization $g(\cdot; \Theta)$ (and possibly h also) is realized as a DNN with parameter Θ to be learned. The loss function $\mathcal{L}(u, u^*)$ is to measure the difference between a reconstruction u and the corresponding ground truth image u^* from the training data. The optimal parameter Θ of g (and h) is then obtained by solving the upper-level optimization (38a).

If the actual minimizer $u(b; \Theta)$ is replaced by the direct output of an LOA-based DNN (such as ISTA-Net etc. in the previous subsection) which mimics an iterative optimization scheme for solving the lower-level minimization in the constraint of (38) and then (38) reduces to the unrolling methods introduced in the previous subsections. However, the unrolled networks do not have any convergence guarantee, and the learned components do not represent g in (38) and can be difficult to interpret.

To obtain convergence guarantee with interpretable network structures, (Chen et al. 2020) proposed a novel learnable descent algorithm (LDA). Consider the case where the data fidelity term $h(u) := (1/2) \cdot \|Au - b\|^2$ (or any smooth but possibly nonconvex function) and $g(u)$ is a nonsmooth nonconvex regularization function

which is design to be $g(u) = \|r(u)\|_{2,1} = \sum_{i=1}^m \|r_i(u)\|$. Here $r = (r_1, \dots, r_m)$ is a smooth but nonconvex mapping realized by a deep neural network whose parameters are learned from training data, and $r_i(u) \in \mathbb{R}^d$ stands for a d -dimensional feature vector for $i = 1, \dots, m$. To overcome the nondifferentiability issue of $g(u)$, a smooth approximation of g by applying Nesterov's smoothing technique (Nesterov 2005) is employed: $g_\varepsilon(u) = \sum_{i \in I_0} \frac{1}{2\varepsilon} \|r_i(u)\|^2 + \sum_{i \in I_1} \left(\|r_i(u)\| - \frac{\varepsilon}{2} \right)$, where the index set I_0 and its complement I_1 at u for the given r and ε are defined by $I_0 = \{i \in [m] \mid \|r_i(u)\| \leq \varepsilon\}$, $I_1 = [m] \setminus I_0$. Denote $f_\varepsilon(u) = h(u) + g_\varepsilon(u)$ (we omit Θ for notation simplicity). Then LDA iterates

$$z_{k+1} = u_k - \alpha_k \nabla h(u_k), \quad (39a)$$

$$w_{k+1} = z_{k+1} - \tau_k \nabla g_{\varepsilon_k}(z_{k+1}), \quad (39b)$$

$$v_{k+1} = z_{k+1} - \alpha_k \nabla g_{\varepsilon_k}(u_k), \quad (39c)$$

where in each iteration $u_{k+1} = w_{k+1}$ if $f_{\varepsilon_k}(w_{k+1}) \leq f_{\varepsilon_k}(v_{k+1})$ and v_{k+1} otherwise; and $\varepsilon_{k+1} = \lambda \varepsilon_k$ if $\|\nabla f_{\varepsilon_k}(u_{k+1})\| < \sigma \varepsilon_k$ and $\varepsilon_{k+1} = \varepsilon_k$ otherwise, where $\lambda \in (0, 1)$ is a prescribed hyperparameter. It is shown that ε_k will monotonically decrease to 0 such that f_{ε_k} approximates the original nonsmooth nonconvex function f , and any accumulation points of a particular subsequence of $\{u_k\}$ is a Clarke stationary point (analogue to the critical points of differentiable functions) of the nonsmooth nonconvex function f (Chen et al. 2020).

Since LDA follows the algorithm exactly, the convergence of the LDA network can be guaranteed. Moreover, the practical performance of LDA is very promising in a wide range of image reconstruction applications. For example, Table 1 shows the PSNR of the reconstructions obtained by LDA (with r parameterized by a simple generic 4-layer CNN and $K = 15$ total phases) on the dataset Set11 (Kulkarni et al. 2016) with a prefixed sampling matrix. Compared to the classical TV-based reconstruction method and several unrolling methods, LDA achieves the best reconstruction quality with highest PSNR. In addition, LDA uses much fewer parameters than the other networks as Θ is shared by all its phases. In Fig. 9, the qualitative reconstruction result of LDA is shown and compared with several state-

Table 1 Average PSNR (dB) of reconstructions obtained by the some methods on *Set11* dataset with various CS ratios and the number of learnable network parameters (#Param), where the PSNR data is quoted from Zhang and Ghanem (2018) and Chen et al. (2020)

Method	10%	25%	50%	#Param
TVAL3 Li et al. (2013)	22.99	27.92	33.55	NA
IRCNN Zhang et al. (2017)	24.02	30.07	36.23	185,472
ISTA-Net Zhang and Ghanem (2018)	25.80	31.53	37.43	171,090
ISTA-Net ⁺ Zhang and Ghanem (2018)	26.64	32.57	38.07	336,978
LDA Chen et al. (2020)	27.42	32.92	38.50	27,967

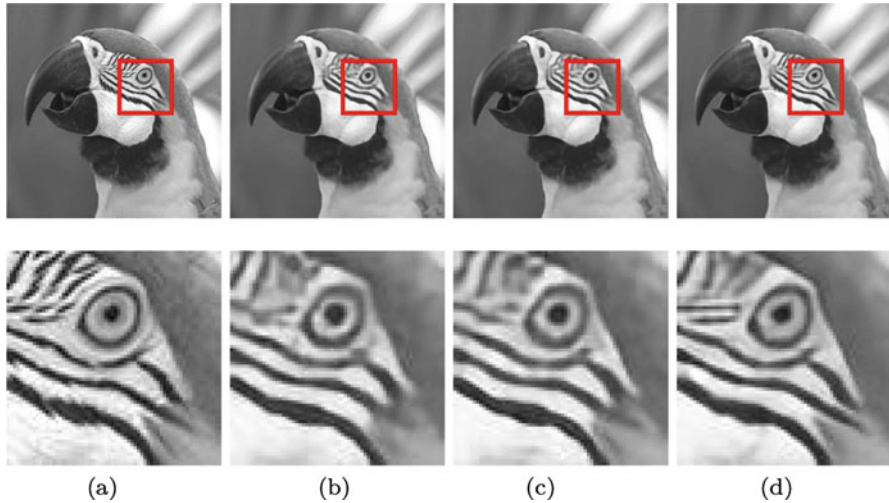


Fig. 9 Reconstruction of parrot image in Set11 (Kulkarni et al. 2016) with CS ratio 10% obtained by CS-Net (Shi et al. 2017), SCS-Net (Shi et al. 2019) and LDA (Chen et al. 2020). Images in the bottom row zoom in the corresponding ones in the top row. PSNR are shown in the parentheses. (a) Reference (b) CS-Net (28.00) (c) SCS-Net (28.10) (d) LDA (29.54)

of-the-art reconstruction networks. A more intriguing property of LDA is that the feature map r is explicitly learned and can be interpreted. In Fig. 10, the 2-norm of the learned feature map r at all pixels is shown and compared to the norm of gradient (forward differences at each pixel) used by the classical TV-based method. It can be seen that important details, such as the antennae of the butterfly, the lip of Lena, and the bill of the parrot, are faithfully recovered by LDA.

Concluding Remarks

We reviewed several typical deep neural networks inspired by the variational method and associated numerical optimization algorithms for the inverse problem of image reconstruction. These neural networks have architectures that mimic the well-known efficient optimization algorithms, such that each phase of a network corresponds to one iteration in the original numerical scheme. The algorithm parameters and other manually selected terms, such as the regularization, in the variational model and optimization algorithm are replaced by learnable components in the deep reconstruction network. The network output is thus a function of these parameters and learnable components. Given the ground truth or high-quality image data, we can form the loss function which measures the discrepancy between the network output and the ground truth and apply back-propagation and stochastic gradient descent method to optimize the parameters such that the loss function is minimized during the training procedure. After training, these networks with optimal parameters serve as fast feedforward networks that can reconstruct high-quality images on the fly.

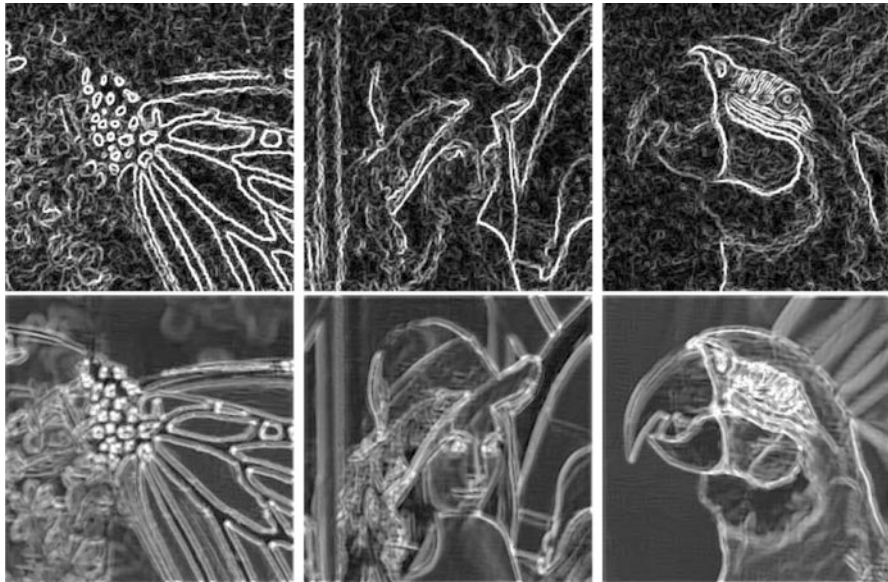


Fig. 10 The norm of the gradient at every pixel in TV based image reconstruction (top row) and the norm of the feature map r at every pixel learned in LDA (bottom row), where important details, such as the antennae of the butterfly, the lip of Lena, and the bill of the parrot, are faithfully recovered by LDA. (Images are obtained from Chen et al. 2020)

These methods have demonstrated significantly improved empirical performance and require much lower computational cost compared to the classical methods in a variety of applications.

Appendix: Back-Propagation in ISTA-Net

For completeness, we provide the details of derivations to obtain gradients of the loss function L in (21) with respect to the network parameters Θ for ISTA-Net. For more details of the network structure and its relation to the back-propagation procedure for ISTA-Net and ADMM-Net introduced in section “[Structured Image Reconstruction Networks](#)”, we refer to Wang et al. (2019).

The process of back-propagation is essentially applying chain rule repeatedly, also called the “back-propagation” in deep learning. To obtain the gradient of the loss function L with respect to the parameters, it is helpful to consult the network structure for the dependency between the parameters and the inputs and outputs of nodes.

We first check the gradients of L defined in (21) with respect to $u^{(k)}$ and $r^{(k)}$. Note that L takes $u^{(k)}$ and $r^{(k)}$, which are vectors in \mathbb{R}^n , and output scalars, we know the gradients of L with respect to $u^{(k)}$ and $r^{(k)}$ are both vectors in \mathbb{R}^n as well. We use

partial derivatives to indicate spatial dependencies and compute the gradients here. First of all, we have

$$\frac{\partial L}{\partial r^{(k)}} = \frac{\partial L}{\partial u^{(k)}} \frac{\partial u^{(k)}}{\partial r^{(k)}}, \quad (40)$$

due to that $u^{(k)}$ is a function of $r^{(k)}$ as shown in Fig. 2. The gradient $\partial u^{(k)}/\partial r^{(k)}$ in (40) is straightforward to compute due to the relation between $r^{(k)}$ and $u^{(k)}$ in (18b) and the chain rule:

$$\frac{\partial u^{(k)}}{\partial r^{(k)}} = \nabla \tilde{H}^{(k)}(s_k) \cdot S'_{\theta_k}(h_k) \cdot \nabla H^{(k)}(r^{(k)}), \quad (41)$$

where the notations are simplified using the following definitions,

$$h_k := H^{(k)} r^{(k)} \quad \text{and} \quad s_k := S_{\theta_k}(h_k). \quad (42)$$

Substituting (41) into (40), we see that $\partial L/\partial r^{(k)}$ can be obtained once we have $\partial L/\partial u^{(k)}$. The gradient $\partial L/\partial u^{(k)}$ can also be computed by the chain rule:

$$\frac{\partial L}{\partial u^{(k)}} = \frac{\partial L}{\partial r^{(k+1)}} \frac{\partial r^{(k+1)}}{\partial u^{(k)}}, \quad (43)$$

where $\partial r^{(k+1)}/\partial u^{(k)}$ is obtained by (18a) for $k \leftarrow k + 1$ as

$$\frac{\partial r^{(k+1)}}{\partial u^{(k)}} = I - \alpha_{k+1} \nabla^2 h(u^{(k)}). \quad (44)$$

Hence, we can get $\partial L/\partial u^{(k)}$ once $\partial L/\partial r^{(k+1)}$ is computed. Therefore, we can compute the gradients of L with respect to $u^{(k)}$ and $r^{(k)}$ for all k in the order from left to right using (40), (41), (43), and (44), starting from $\partial L/\partial u^{(K)} = u^{(K)} - u^*$, as follows:

$$\frac{\partial L}{\partial u^{(K)}} \rightarrow \frac{\partial L}{\partial r^{(K)}} \rightarrow \cdots \rightarrow \frac{\partial L}{\partial r^{(k+1)}} \rightarrow \frac{\partial L}{\partial u^{(k)}} \rightarrow \frac{\partial L}{\partial r^{(k)}} \rightarrow \cdots \rightarrow \frac{\partial L}{\partial u^{(0)}} \quad (45)$$

That is, we first compute $\partial L/\partial u^{(K)} = u^{(K)} - u^*$ according to the definition of L in (21), use it to compute $\partial L/\partial r^{(K)}$ according to (40) and (41), and then $\partial L/\partial u^{(K-1)}$ according to (43) and (44), and so on. This is the effect of back-propagation.

Now we compute the gradients of $r^{(k)}$ and $u^{(k)}$ with respect to the network parameters used in (18a) and (18b), respectively. The derivative of $r^{(k)}$ with respect to α_k is straightforward due to (18a):

$$\frac{\partial r^{(k)}}{\partial \alpha_k} = -\nabla h(u^{(k)}). \quad (46)$$

The gradient of $u^{(k)}$ with respect to $w_j^{(k)}$ in the j th layer of the CNN $H^{(k)}$ defined in (17) can be obtained by applying the chain rule to (18b):

$$\frac{\partial u^{(k)}}{\partial w_j^{(k)}} = \nabla \tilde{H}^{(k)}(s_k) \cdot S'_{\theta_k}(h_k) \cdot \frac{\partial h_k}{\partial w_j^{(k)}} \quad (47)$$

for $j = 1, 2$, where h_k is the output of $H^{(k)}$ given the input $r^{(k)}$ and s_k is the output of S_{θ_k} given the input h_k defined in (42). The partial derivative $\partial h_k / \partial w_j^{(k)}$ is standard as in the back-propagation of CNN, which we omit the details here. Similarly, the gradient of $u^{(k)}$ with respect to $\tilde{w}_j^{(k)}$ in the j th layer of the CNN $\tilde{H}^{(k)}$ defined in (17) can be obtained since $u^{(k)}$ and s_k are the output and input of $\tilde{H}^{(k)}$, respectively. The gradient of $u^{(k)}$ with respect to θ_k is slightly different:

$$\frac{\partial u^{(k)}}{\partial \theta_k} = \nabla \tilde{H}^{(k)}(s_k) \cdot \frac{\partial S_{\theta_k}(h_k)}{\partial \theta_k}. \quad (48)$$

In this case, we will need to treat $S_{\theta_k}(h_k) \in \mathbb{R}^n$ as a function of θ_k for given h_k , i.e., $S.(h_k) : \theta_k \mapsto S_{\theta_k}(h_k)$ defined by

$$[S_{\theta_k}(h_k)]_i = \begin{cases} -\theta_k + [h_k]_i & \text{if } 0 < \theta_k < h_k, \\ \theta_k - [h_k]_i & \text{if } 0 < \theta_k < -h_k, \\ 0 & \text{otherwise.} \end{cases} \quad (49)$$

Hence, the derivative of $S_{\theta_k}(h_k)$ with respect to θ_k is

$$\left[\frac{\partial S_{\theta_k}(h_k)}{\partial \theta_k} \right]_i = \begin{cases} -1 & \text{if } 0 < \theta_k < h_k, \\ 1 & \text{if } 0 < \theta_k < -h_k, \\ 0 & \text{otherwise.} \end{cases} \quad (50)$$

With all the partial derivatives obtained above, we can apply the chain rule to compute the gradient of L with respect to each of the network parameters. For example,

$$\frac{\partial L}{\partial \alpha_k} = \frac{\partial L}{\partial r^{(k)}} \frac{\partial r^{(k)}}{\partial \alpha_k}, \quad (51)$$

where $\partial L / \partial r^{(k)}$ is obtained by (40) and (41) following the back-propagation process and $\partial r^{(k)} / \partial \alpha_k$ is obtained by (46). The partial derivatives with respect to the other parameters can be similarly computed as follows:

$$\frac{\partial L}{\partial \theta_k} = \frac{\partial L}{\partial u^{(k)}} \frac{\partial u^{(k)}}{\partial \theta_k}, \quad \frac{\partial L}{\partial w_j^{(k)}} = \frac{\partial L}{\partial u^{(k)}} \frac{\partial u^{(k)}}{\partial w_j^{(k)}}, \quad \frac{\partial L}{\partial \tilde{w}_j^{(k)}} = \frac{\partial L}{\partial u^{(k)}} \frac{\partial u^{(k)}}{\partial \tilde{w}_j^{(k)}} \quad (52)$$

where $\partial L / \partial u^{(k)}$ is obtained by (43) and (44) and the partial derivatives of $u^{(k)}$ with respect to θ_k , $w_j^{(k)}$, and $\tilde{w}_j^{(k)}$ are obtained similarly as explained above.

With these gradients of L with respect to the network parameters, we can employ a stochastic gradient descent (SGD) method and find the optimal parameters Θ^* that minimizes (21) over the entire training dataset. With the optimal Θ^* , ISTA-Net works as a feedforward mapping, which takes imaging data b and outputs a reconstructed image $u^{(K)}$. This feedforward mapping can be computed very fast since all operations in (18) are explicit given Θ^* .

References

- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Aubert, G., Vese, L.: A variational method in image recovery. *SIAM J. Numer. Anal.* **34**(5), 1948–1979 (1997)
- Bennett Landman, S.W.E.: 2013 diencephalon free challenge (2013). <https://doi.org/10.7303/syn3270353>
- Borgerding, M., Schniter, P., Rangan, S.: Amp-inspired deep networks for sparse linear inverse problems. *IEEE Trans. Signal Process.* **65**(16), 4293–4308 (2017)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* **40**(1), 120–145 (2011)
- Chen, X., Liu, J., Wang, Z., Yin, W.: Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In: *Advances in Neural Information Processing Systems*, pp. 9061–9071 (2018)
- Chen, Y., Liu, H., Ye, X., Zhang, Q.: Learnable descent algorithm for nonsmooth nonconvex image reconstruction. *arXiv preprint arXiv:2007.11245* (2020)
- Cheng, J., Wang, H., Ying, L., Liang, D.: Model learning: Primal dual networks for fast mr imaging. *ArXiv abs/1908.02426* (2019)
- Chun, I.Y., Huang, Z., Lim, H., Fessler, J.A.: Momentum-net: Fast and convergent iterative neural network for inverse problems. *arXiv preprint arXiv:1907.11818* (2019)
- Dal Maso, G., Morel, J.M., Solimini, S.: A variational method in image segmentation: Existence and approximation results. *Acta Math.* **168**(1), 89–151 (1992)
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. *IEEE* (2009)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: *Deep Learning*, vol. 1. MIT Press, Cambridge (2016)
- Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: J. Fürnkranz, T. Joachims (eds.) *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pp. 399–406. Haifa (2010)
- Hammernik, K., Klatzer, T., Kobler, E., Recht, M.P., Sodickson, D.K., Pock, T., Knoll, F.: Learning a variational network for reconstruction of accelerated MRI data. *Magn. Reson. Med.* **79**(6), 3055–3071 (2018)

- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- Heide, F., Steinberger, M., Tsai, Y.T., Rouf, M., Pająk, D., Reddy, D., Gallo, O., Liu, J., Heidrich, W., Egiazarian, K., et al.: Flexisp: A flexible camera image processing framework. *ACM Trans. Graph.* **33**(6), 231 (2014)
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al.: Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **29**, 82–97 (2012)
- Koepfler, G., Lopez, C., Morel, J.M.: A multiscale algorithm for image segmentation by variational method. *SIAM J. Numer. Anal.* **31**(1), 282–299 (1994)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
- Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., Ashok, A.: Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 449–458 (2016)
- Li, C., Yin, W., Jiang, H., Zhang, Y.: An efficient augmented lagrangian method with applications to total variation minimization. *Comput. Optim. Appl.* **56**(3), 507–530 (2013)
- Liu, J., Chen, X., Wang, Z., Yin, W.: Alista: Analytic weights are as good as learned weights in lista. In: ICLR (2019)
- Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: New challenges for image quality assessment models. *IEEE Trans. Image Process.* **26**(2), 1004–1016 (2016)
- Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of 8th International Conference Computer Vision, vol. 2, pp. 416–423 (2001)
- Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1781–1790 (2017)
- Nesterov, Y.: Smooth minimization of non-smooth functions. *Math. Program.* **103**(1), 127–152 (2005)
- Rick Chang, J., Li, C.L., Póczos, B., Vijaya Kumar, B., Sankaranarayanan, A.C.: One network to solve them all—solving linear inverse problems using deep projection models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5888–5897 (2017)
- Roth, S., Black, M.J.: Fields of experts. *Int. J. Comput. Vis.* **82**(2), 205 (2009)
- Sarikaya, R., Hinton, G.E., Deoras, A.: Application of deep belief networks for natural language understanding. *IEEE/ACM Trans. Audio Speech Lang. Process.* **22**(4), 778–784 (2014)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Springer, New York (2009)
- Schlemper, J., Caballero, J., Hajnal, J.V., Price, A.N., Rueckert, D.: A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Trans. Med. Imaging* **37**(2), 491–503 (2018)
- Shi, W., Jiang, F., Liu, S., Zhao, D.: Scalable convolutional neural network for image compressed sensing. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
- Shi, W., Jiang, F., Zhang, S., Zhao, D.: Deep networks for compressed image sensing. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 877–882. IEEE (2017)
- Socher, R., Bengio, Y., Manning, C.D.: Deep learning for nlp (without magic). In: Tutorial Abstracts of ACL 2012, pp. 5–5. Association for Computational Linguistics (2012)
- Sprechmann, P., Bronstein, A.M., Sapiro, G.: Learning efficient sparse and low rank models. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1821–1833 (2015)
- Sun, J., Li, H., Xu, Z., et al.: Deep admm-net for compressive sensing mri. In: Advances in Neural Information Processing Systems, pp. 10–18 (2016)
- Timofte, R., De Smet, V., Van Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Asian Conference on Computer Vision, pp. 111–126. Springer (2014)

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)
- Wang, G., Zhang, Y., Ye, X., Mou, X.: *Machine Learning for Tomographic Imaging* (2019). IOP Publishing. <https://doi.org/10.1088/2053-2563/ab3cc4>
- Wang, S., Fidler, S., Urtasun, R.: Proximal deep structured models. In: *Advances in Neural Information Processing Systems*, pp. 865–873 (2016)
- Xie, X., Wu, J., Zhong, Z., Liu, G., Lin, Z.: Differentiable linearized admm. arXiv preprint arXiv:1905.06179 (2019)
- Xin, B., Wang, Y., Gao, W., Wipf, D., Wang, B.: Maximal sparsity with deep networks? In: *Advances in Neural Information Processing Systems*, pp. 4340–4348 (2016)
- Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *European Conference on Computer Vision*, pp. 818–833. Springer (2014)
- Zhang, J., Ghanem, B.: Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837 (2018)
- Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3929–3938 (2017)



Juan Carlos De los Reyes and David Villacís

Contents

Introduction	910
Variational Inverse Problems Setting	912
Image Reconstruction as an Inverse Problem	912
Regularizers	912
Restoration Models	915
Optimality and Duality	915
Solution Methods	916
Bilevel Optimization in Imaging	917
Total Variation Gaussian Denoising	919
Solution Algorithms	924
Infinite-Dimensional Case	924
Existence and Other Properties	926
Stationarity Conditions	927
Dualization	929
Nonlocal Problems	930
Neural Network Optimization	932
Deep Neural Networks as a Further Regularizer	933
Deep Unrolling Within Optimization	933
Numerical Experiments	934
Conclusions	938
References	939

Abstract

Optimization techniques have been widely used for image restoration tasks, as many imaging problems may be formulated as minimization ones with the

J. C. De los Reyes (✉) · D. Villacís
Research Center for Mathematical Modelling (MODEMAT), Escuela Politécnica Nacional,
Quito, Ecuador
e-mail: juan.delosreyes@epn.edu.ec; david.villacis01@epn.edu.ec

recovered image as the target minimizer. Recently, novel optimization ideas also entered the scene in combination with machine learning approaches, to improve the reconstruction of images by optimally choosing different parameters/functions of interest in the models. This chapter provides a review of the latest developments concerning the latter, with special emphasis on bilevel optimization techniques and their use for learning local and nonlocal image restoration models in a supervised manner. Moreover, the use of related optimization ideas within the development of neural networks in imaging will be briefly discussed.

Keywords

Bilevel optimization · Machine learning · Variational models

Introduction

Several classical image processing tasks such as denoising, inpainting, and deblurring, among others, may be treated as minimization problems in suitable function spaces and using properly chosen energy functionals, typically nonsmooth ones. As a consequence, the historical connection between optimization and imaging has been very fruitful, and several analytical and algorithmic developments have originated from this close relationship. We refer to Chambolle and Pock (2016) and the references therein for a thorough review on these links and current developments.

More recently, new optimization ideas entered the scene hand in hand with modern data-driven approaches. Although machine learning techniques have years of tradition on solving inverse and imaging problems, its use in combination with structural properties of the mathematical models has proven to be of relevance, leading to state-of-the-art developments and applications (see, e.g., Calatroni et al. 2017; Arridge et al. 2019; Holler et al. 2018; Hintermüller and Papafitsoros 2019; Sherry et al. 2020).

A learning approach that combines practical and theoretical advantages is *bilevel optimization*. Within this setting, the imaging problems are considered as lower-level constraints, while on the upper-level a loss function, based on a training set, is used for estimating the different parameters involved in the models. The resulting mathematical problems pose different challenges that need to be addressed using sophisticated tools from variational and nonsmooth analysis (Outrata 2000; Mordukhovich 2018; Schirotzek 2007).

A prototypical problem in this direction is the parameter learning associated with image restoration models. An initial contribution in this respect was the paper by Tappen and coauthors Tappen (2007), where the parameters of a *Markov random fields* model were learned by means of variational optimization. Thereafter, Haber and coauthors Haber et al. (2008) considered a general learning approach for inverse problems and, although no mathematical theory was developed, made a case for the

successful application of such methodology. A renewed interest took place around the year 2013, where on basis of developments on optimal control of variational inequalities, the learning of parameters for variational denoising models was carried out in function space (De los Reyes 2011) and in finite-dimensions (Kunisch and Pock 2013). Since then, the field has expanded, and several papers have been devoted to different theoretical and computational aspects: noise model learning (Calatroni et al. 2013; Calatroni and Papafitsoros 2019), higher-order regularizers (De los Reyes et al. 2017; Davoli and Liu 2018; Davoli et al. 2019; Hintermüller and Rautenberg 2017), blind deconvolution (Hintermüller and Wu 2015), inexact gradients (Ochs et al. 2016; Ehrhardt and Roberts 2020), and nonlocal models (d’Elia et al. 2019; Bartels and Weber 2020).

When confronted with variational imaging models, the bilevel optimization problem structure becomes quite involved to be analyzed, as classical nonlinear or bilevel programming results (see, e.g., Dempe 2002) cannot be directly utilized. As a remedy, tools from nonsmooth variational analysis have to be employed to cope with the difficulties related with the lack of differentiability of the solution mapping or the failure of standard constraint qualification conditions. In finite dimensions, for instance, generalized Mordukhovich tangential and normal cones (Mordukhovich 2018) have to be computed in order to obtain relatively sharp stationarity conditions. These aspects will be illustrated in section “[Bilevel Optimization in Imaging](#)” of this manuscript, targeting the parameter learning of image denoising problems.

The analysis of the infinite-dimensional counterpart becomes even harder, as topological properties of finite-dimensional spaces are in general missing and, therefore, variational analysis results on generalized normal cones are mostly inapplicable. The study of the function space setting, however, has proven to be of importance for deriving structural properties of the reconstructed images and optimal parameters (De los Reyes et al. 2016), as well as for devising mesh-independent solution algorithms. Moreover, the study of spatially dependent parameters in variational imaging problems has attracted increasing attention in recent years. Apart of the learning approach carried out in Van Chung et al. (2017), Hintermüller and coauthors have considered an alternative loss functional based on image statistics in combination with dualization of the lower-level problem (Hintermüller and Rautenberg 2017). Recently, also bilevel problems with infinite-dimensional nonlocal variational lower-level models have been investigated (d’Elia et al. 2019). A summary of these contributions will be presented in section “[Infinite-Dimensional Case](#)” of this chapter.

Although supervised bilevel learning has been usually presented as a competing approach to modern neural networks, theoretical results obtained for the variational optimization problems may be considered in the design of novel types of neural networks as well. This effort has been carried out in Lunz et al. (2018) and Kobler et al. (2020), where generative adversarial neural networks and multi-scale convolutional neural networks are considered, respectively. Moreover, the use of neural networks for improving the efficiency of intermediate steps within an optimization method has also been studied (Adler and Öktem 2018; Sun et al.

2016; Kobler et al. 2017). A short discussion on these connections is provided in section “[Neural Network Optimization](#)”.

Variational Inverse Problems Setting

Image Reconstruction as an Inverse Problem

Image reconstruction aims to restore or enhance a degraded image obtained by a given acquisition process. In general, images can be degraded due to poor imaging conditions and problems in the storage device or the communication channel, to name a few. A frequentist model used to analyze this phenomenon can be stated as

$$f = A(u) + n, \quad (1)$$

where u is the original image, f is the observed degraded image, n is the noise contained in the observed image, and A is a possibly nonlinear forward operator that models the acquisition process. In most imaging problems, the operator A is rank deficient, leading to an ill-posed inverse problem. Therefore, nonuniqueness of solutions or instability of the direct inversion of such operator motivates the use of different solution techniques.

A classical way to solve such inverse problems is to make use of a variational “energy” formulation. Using this methodology, we can state the solution of (1) as the solution of the following optimization problem:

$$\hat{u} := \arg \min_u \mathcal{E}(u, \lambda, \alpha) := \mathcal{F}(u, \lambda) + \mathcal{R}(Hu, \alpha), \quad (2)$$

where \hat{u} is the reconstructed image, H a bounded linear operator, \mathcal{F} the *data fidelity*, and \mathcal{R} a *regularization* term. The parameters λ and α affect the contribution of the fidelity and regularization terms to the final solution, respectively. The choice of these two terms has a crucial impact on the solution. Indeed, the data fidelity term models the type of noise present in the image, while the regularization term promotes certain features which are known a priori about the image.

Regularizers

A seminal idea proposed by Tikhonov and Arsenin (1977) for the solution of inverse problems is to use the following type of regularization term:

$$\mathcal{R}(\nabla u, \alpha) = \alpha \int_{\Omega} \|\nabla u\|_2^2 dx, \quad (3)$$

aiming at recovering certain smooth properties of the solution. In the context of image restoration, however, the solution obtained correspondingly is not desirable, precisely since the regularizer involved has very strong isotropic smoothing properties which leads to a loss of edge information in the reconstructed image.

In order to preserve the edge information as much as possible, Rudin et al. (1992) proposed the use of the *isotropic total variation* of the image:

$$TV_\alpha(u) := \alpha \int_\Omega \|\nabla u\|_2 dx. \tag{4}$$

This regularizer promotes solutions close to a piecewise constant image that is composed by homogeneous regions separated by sharp edges. Because one of the main characteristics of images are edges, as they define divisions between objects in a scene, their preservation seems like a good idea and a favorable feature of TV regularization. The drawback of such a regularization procedure becomes apparent as soon as it is applied to images that are not only consist of constant intensity regions and jumps but also possess more complicated structures, like smooth intensity variations or textures. A well-known artifact introduced by TV regularization in this case is called staircasing (Ring 2000).

One possibility to counteract such artifacts is the introduction of higher-order derivatives in the image regularization. Two main second-order total variation models have been introduced in the past: the infimal-convolution total variation (ICTV) model of Chambolle and Lions Chambolle and Lions (1997) and the total generalized variation (TGV) proposed by Bredies and coauthors (2010). Although higher-order models were also formally introduced, we focus on second-order ones, since these regularizers have received much more attention in recent relevant imaging applications (Knoll et al. 2011; Bredies et al. 2010). For an open and bounded image domain $\Omega \subset \mathbb{R}^2$, the ICTV regularizer reads:

$$ICTV_{\alpha,\beta}(u) := \min_{v \in W^{1,1}(\Omega), \nabla v \in BV(\Omega)} \alpha \|Du - \nabla v\|_{m(\Omega; \mathbb{R}^2)} + \beta \|D\nabla v\|_{m(\Omega; \mathbb{R}^{2 \times 2})}. \tag{5}$$

On the other hand, second-order TGV (Bredies et al. 2010) reads:

$$TGV_{\alpha,\beta}^2(u) := \min_{w \in BD(\Omega)} \alpha \|Du - w\|_{m(\Omega; \mathbb{R}^2)} + \beta \|Ew\|_{m(\Omega; \text{Sym}^2(\mathbb{R}^2))}. \tag{6}$$

Here $BD(\Omega) := \{w \in L^1(\Omega; \mathbb{R}^n) \mid \|Ew\|_{m(\Omega; \mathbb{R}^{n \times n})} < \infty\}$ is the space of vector fields of bounded deformation on Ω , and E denotes the *symmetrized gradient* and $\text{Sym}^2(\mathbb{R}^2)$ the space of symmetric tensors of order 2 with arguments in \mathbb{R}^2 . The parameters α, β are fixed positive parameters. The main difference between (5) and (6) is that we do not generally have that $w = \nabla v$ for any function v . That results in some qualitative differences of ICTV and TGV regularization; compare, for instance De los Reyes et al. (2017).

Although TV-based regularizers perform well in many instances, for images with texture structures, neighborhood information is not enough to get good

reconstructions. A remedy to this are nonlocal models, which consider similar intensity patterns between pixels or patches in a given spatial neighborhood or all over the whole image domain. Originally, the main concern within this framework was the design of direct nonlocal filters (Yaroslavsky 1986; Tomasi and Manduchi 1998; Buades et al. 2005), being the *nonlocal means filter* arguably the more popular regularizer in this context. The techniques diversified afterward with the consideration of different energy functionals to accomplish the denoising task (Gilboa and Osher 2007, 2008; Lou et al. 2010). In particular, the variational framework developed in Gilboa and Osher (2007) enabled the employment of additional modeling features that have been used already for image reconstruction tasks in local models. A modified variational nonlocal means regularizer, for instance, is given by

$$NL(u) := \int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (u(\mathbf{x}) - u(\mathbf{y}))^2 \gamma(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}d\mathbf{x}, \tag{7}$$

with the localized integrable kernel

$$\gamma(\mathbf{x}, \mathbf{y}) = \exp \left\{ - \int_{B_\rho(0)} w(\boldsymbol{\tau}) (f(\mathbf{x} + \boldsymbol{\tau}) - f(\mathbf{y} + \boldsymbol{\tau}))^2 \, d\boldsymbol{\tau} \right\} \chi(\mathbf{y} \in B_\epsilon(\mathbf{x})),$$

Here, Ω_I stands for the interaction domain of a bounded region Ω consisting of all points outside of the domain that interact with points inside of it. The function $w(t)$ controls the intensity threshold at which the nonlocal filter acts and is the target of a learning scheme. For a comparison between total variation and nonlocal means, see Fig. 1.

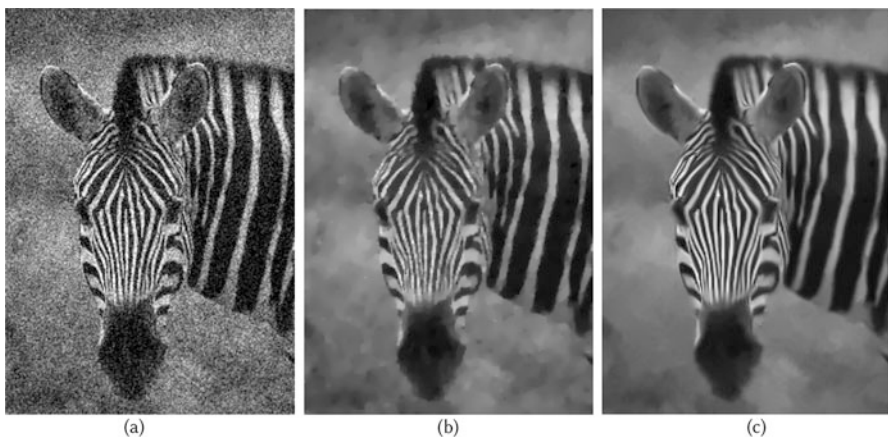


Fig. 1 Comparison of regularizers in variational image denoising. (a) Noisy (b) Total Variation (c) Nonlocal Means

Restoration Models

Three well-known image restoration tasks are denoising, deblurring, and inpainting. The goal of denoising is to recover a noise-free image u from a particular noise contaminated one f . This perturbation is usually modeled based on the statistical estimates or approximated by a proper noise model coming from the physics behind the acquisition of f . For a normally distributed f , the data term corresponds to a squared Euclidean norm (Rudin et al. 1992):

$$\mathcal{F}(u, \lambda) := \lambda \int_{\Omega} \|u - f\|^2 dx. \quad (8)$$

In the case of a Poisson noise distribution present in the damaged image, the data fidelity term was studied in Sawatzky et al. (2009) and Le et al. (2007) and has the form $\mathcal{F}(u, \lambda) := \lambda \int_{\Omega} (u - f) \log u dx$. In Nikolova (2004), the author studied impulse noise contaminated images and proposed the nonsmooth data fidelity term $\mathcal{F}(u, \lambda) := \lambda \int_{\Omega} \|u - f\|_1 dx$. Other convex and non-convex data fidelity models, as well as several combinations, have been investigated as well.

In the case of deblurring, the task consists in recovering a sharp image from its blurry observation. This blur usually comes as an *optical blur* from the deviation of the object from the focused imaging plane, *mechanical blur* from the rapid motion of either the target object or the imaging device, and of *medium-induced blur* due to the optical turbulence of the photonics media. Given a blur operator A , the image deblurring problem reads

$$\mathcal{F}(u, \lambda) := \lambda \int_{\Omega} \|A(u) - f\|^2 dx. \quad (9)$$

The remaining task, *image inpainting*, consists in recovering lost parts of a damaged image. If Ω corresponds to the original image domain, due to different problems in image acquisition, transmission, and numerous external factors, there usually exists a subdomain $\Omega_0 \subset \Omega$ where the information is missing. Moreover, the observable portion of the image $\Omega \setminus \Omega_0$ is often degraded with noise and blur. The final goal of this task, which also encompasses denoising and deblurring, is to reconstruct the image in the entire domain Ω from this degraded observation. The fidelity term takes typically the Gaussian form:

$$\mathcal{F}(u, \lambda) := \lambda \int_{\Omega \setminus \Omega_0} \|A(u) - f\|^2 dx. \quad (10)$$

Optimality and Duality

As described in the previous section, variational regularizers are typically nonsmooth, while fidelity terms are in many circumstances convex and differentiable. In both cases, however, convexity appears to be an important feature, which enables

the use of convex analysis tools for characterizing the solution of the restoration models at hand.

By restating problem (2) for fixed parameters $\lambda \in \mathcal{P}_\lambda^+$ and $\alpha \in \mathcal{P}_\alpha^+$, we obtain

$$\min_{u \in X} \mathcal{F}_\lambda(u) + \mathcal{R}_\alpha(Hu), \tag{11}$$

where X, Y are two Banach spaces and $\mathcal{P}_\lambda^+, \mathcal{P}_\alpha^+$ are suitable positive sets in the parameters spaces. Assuming that $\mathcal{R}_\alpha : Y \rightarrow \mathbb{R}$ is a proper convex, lower semicontinuous, and possibly nonsmooth function; $\mathcal{F}_\lambda : X \rightarrow \mathbb{R}$ a smooth, proper convex, and lower semicontinuous function; and $H : X \rightarrow Y$ a bounded linear operator, the optimality condition for this primal problem reads

$$0 \in \partial(\mathcal{F}_\lambda(u) + \mathcal{R}_\alpha(Hu)) = \partial(\mathcal{F}_\lambda(u)) + H^*(\partial\mathcal{R}_\alpha(Hu)), \tag{12}$$

where $\partial(\cdot)$ denotes the standard convex analysis subdifferential. Introducing the dual multiplier $q \in Y$, the dual problem of (11) is given by

$$\max_{q \in Y} -\mathcal{F}_\lambda^*(-H^*q) - \mathcal{R}_\alpha^*(q), \tag{13}$$

where \mathcal{F}_λ^* and \mathcal{R}_α^* stand for the convex conjugate of \mathcal{F}_λ and \mathcal{R}_α , respectively.

By satisfying some suitable hypotheses on \mathcal{F}_λ and \mathcal{R}_α , existence of optimal solutions for both the primal and dual problems can be guaranteed. Furthermore, it can be proven that the cost functional values coincide and that both solutions are linked through extremality conditions, i.e., the primal \hat{u} and dual \hat{q} optimal solutions satisfy

$$H^*\hat{q} \in \partial\mathcal{F}_\lambda(\hat{u}), \tag{14a}$$

$$-\hat{q} \in \partial\mathcal{R}_\alpha(H\hat{u}). \tag{14b}$$

In addition, we can formulate an equivalent primal-dual saddle point problem (Ekeland and Temam 1999) with the following structure:

$$\min_{u \in X} \max_{q \in Y} \langle H(u), q \rangle + \mathcal{F}_\lambda(u) - \mathcal{R}_\alpha^*(q). \tag{15}$$

Solution Methods

Since the nonsmoothness of the function \mathcal{R}_α prevents the direct use of standard differentiable techniques, there are several numerical strategies for finding solutions to (2). A first idea consists in solving this type of problems by making use of subgradient-based methods for dealing with the primal problem directly. Although this appears to be the most natural approach, this option has the drawback of the classical slow convergence rate of subgradient methods (Beck 2017 Chapter 8).

By exploiting the differentiability of \mathcal{F}_λ and the fact that in general the regularizer \mathcal{R}_α is a simple convex lower semicontinuous function, *forward-backward* splitting schemes were developed, where in each iteration a gradient descent step on \mathcal{F} and a proximal step on \mathcal{R}_α are performed. The resulting algorithm behaves robustly and gets faster as the smoothness properties of \mathcal{F}_λ improve. Moreover, accelerated versions of this scheme (like the FISTA algorithm) became quite popular in the last years.

Alternatively, the saddle point formulation (15) may be numerically exploited. A popular strategy considers an alternate update, where first a descent step for the primal variable u is performed and thereafter an ascent step in the dual variable p is carried out. This procedure, called *ADMM*, can further be speed up by considering a relaxation step (see, e.g., Chambolle and Pock 2011). These primal-dual update steps are well-suited for parallel computation, making these methods practical for high-resolution image denoising (Villacís 2017). Related popular primal-dual methods are the well-known Douglas-Rachford and the Chambolle-Pock algorithms. An extension to nonlinear operators H can be found in Valkonen (2014).

Another frequent numerical alternative consists in regularizing the non-differentiable term by means of a sufficiently smooth function. As a consequence, fast second-order methods, i.e., methods where both gradient and hessian information is used to define a descent direction, may be devised for the solution of the regularized problems. Indeed, Newton and semismooth Newton methods, along with globalization strategies, have been used to solve image restoration models (see, e.g., Hintermüller and Stadler 2006; De los Reyes and Schönlieb 2013).

Bilevel Optimization in Imaging

The parameters λ and α , considered as invariant in the previous section, actually play a crucial role in the quality of the reconstructed image. Instead of trying to tune them by trial-and-error, the natural question on whether it is possible to select them in an optimal way arises. Combining existing training sets with a supervised bilevel optimization framework, a rigorous learning approach has been developed for variational image restoration in recent years (De los Reyes and Schönlieb 2013; De los Reyes et al. 2017; Kunisch and Pock 2013; Hintermüller and Wu 2015).

Let us consider a training dataset of P pairs $(u_k^{\text{train}}, f_k)$, for $k = 1, \dots, P$, where each u_k^{train} corresponds to ground-truth data and f_k to the corresponding corrupted one. To obtain the optimal parameters (λ, α) , we consider the following class of *bilevel optimization* problems:

$$\min_{(\lambda, \alpha)} \quad \sum_{k=1}^P J(u_k, u_k^{\text{train}}) \quad (16)$$

$$\text{s.t.} \quad u_k \in \arg \min_{u \in \mathbb{R}^n} \mathcal{E}(u, \lambda, \alpha, f_k), \quad (17)$$

where the upper-level problem handles the optimal parameter loss function J , while the lower-level problem corresponds to the restoration model of interest.

A general family of lower-level problems that allow us to learn the noise model, as described in De los Reyes and Schönlieb (2013) and Calatroni et al. (2013), as well as the weights for a family of regularizers (De los Reyes et al. 2017; Kunisch and Pock 2013) is given by the energy

$$\arg \min_{u \in \mathbb{R}^n} \mathcal{E}(u, \lambda, \alpha, f) := \sum_{j=1}^M \sum_{i=1}^{r_j} \lambda_{j,i} \phi_j(u; f)_i + \sum_{l=1}^N \sum_{i=1}^{s_l} \alpha_{l,i} \|(\mathbb{B}_l u)_i\|, \quad (18)$$

where $\phi_j, j = 1, \dots, M$, are different convex restoration (fidelity) models and $\mathbb{B}_l, l = 1, \dots, N$, are bounded linear operators (matrices or tensors) related to different regularizers. The norm $\|\cdot\|$ corresponds to the Euclidean or the Frobenius norm, depending on the corresponding operators. The vector $u \in \mathbb{R}^n$ can be just the reconstructed image or an extended version that includes additional information (e.g., higher-order information). The abstract model (18) has indeed two sets of model parameters: λ for the different data terms available and α for the regularization terms considered. Moreover, these parameters may be considered *scale-dependent*, meaning that each parameter $\lambda_j \in \mathbb{R}_+^{r_j}, \alpha_l \in \mathbb{R}_+^{s_l}$, takes one scalar value for each component (pixel, patch, etc.) of the image model and regularizer, respectively.

In contrast, by assuming scalar parameters $\alpha_l, \lambda_j \in \mathbb{R}_+$, we will affect all components with the same intensity, yielding

$$\arg \min_{u \in \mathbb{R}^n} \mathcal{E}(u, \lambda, \alpha, f) := \sum_{j=1}^M \lambda_j \sum_{i=1}^{r_j} \phi_j(u; f)_i + \sum_{l=1}^N \alpha_l \sum_{i=1}^{s_l} \|(\mathbb{B}_l u)_i\|. \quad (19)$$

Moreover, a further generalization for patch-dependent parameters can be made. Let us consider $\lambda_j \in \mathbb{R}^{m_j}, \alpha_l \in \mathbb{R}^{m_l}$, with $m_j, m_l \ll n$, and patch operators $P_j : \mathbb{R}^{m_j} \mapsto \mathbb{R}_+^{r_j}$ and $Q_l : \mathbb{R}^{m_l} \mapsto \mathbb{R}_+^{s_l}$. The lower-level problem energy may then be written as

$$\arg \min_{u \in \mathbb{R}^n} \mathcal{E}(u, \lambda, \alpha, f) := \sum_{j=1}^M \sum_{i=1}^{r_j} P_j(\lambda_j)_i \phi_j(u; f)_i + \sum_{l=1}^N \sum_{i=1}^{s_l} Q_l(\alpha_l)_i \|(\mathbb{B}_l u)_i\|. \quad (20)$$

Most classical image denoising variational models (TV- l_2 , TV- l_1 , TGV- l_2 , ICTV- l_2 , etc.) as well as TV deblurring and inpainting are instances of the latter.

Also an essential component of The bilevel problem are equations (16) and (17) is the loss function J , which models the quality of the reconstruction when compared to the original image provided in the dataset. One classic approach is to

compute the difference between a ground truth image u^{train} and its reconstruction u using a mean squared error (MSE) criteria $J(u, u^{\text{train}}) = MSE(u, u^{\text{train}}) := \frac{1}{2} \|u - u^{\text{train}}\|_2^2$, which is closely related to the peak signal-to-noise ratio quality measure $PSNR(u, u^{\text{train}}) := 10 \log_{10}(255^2 / MSE(u, u^{\text{train}}))$. Even though this measure is widely used in the imaging community due to its low computational complexity, it depends strongly on the image intensity scaling. Furthermore, PSNR does not necessarily coincide with a human visual response to the image quality.

A more reliable quality measure proposed is the structural similarity index (SSIM) (Wang et al. 2004), which can be casted as

$$J(u, u^{\text{train}}) = SSIM(u, u^{\text{train}}) = l(u, u^{\text{train}})c(u, u^{\text{train}})s(u, u^{\text{train}}),$$

where

$$\begin{aligned} l(u, u^{\text{train}}) &= \frac{2\mu_u \mu_{u^{\text{train}}} + C_1}{\mu_u^2 + \mu_{u^{\text{train}}}^2 + C_1}, \\ c(u, u^{\text{train}}) &= \frac{2\sigma_u \sigma_{u^{\text{train}}} + C_2}{\sigma_u^2 + \sigma_{u^{\text{train}}}^2 + C_2}, \\ s(u, u^{\text{train}}) &= \frac{2\sigma_{uu^{\text{train}}} + C_3}{\sigma_u + \sigma_{u^{\text{train}}} + C_3}, \end{aligned}$$

and μ_u and σ_u correspond to the mean luminance and the standard deviation of the image u , respectively. The use of this quality measure in the bilevel optimization context is, however, restrictive due to its nonsmoothness and non-convexity.

An alternative loss function aimed at prioritizing jump preservation was proposed in De los Reyes et al. (2017), where the authors make use of a Huber regularization of a total variation cost:

$$J(u, u^{\text{train}}) := \sum_{j=1}^m \|\mathbb{K}(u - u^{\text{train}})_j\| \epsilon.$$

This loss function is differentiable, convex, and it was proven advantageous for evaluating the quality of the reconstructed image.

Total Variation Gaussian Denoising

To simplify the exposition of the methodology, let us restrict the analysis to the bilevel problem (16) in the specific case of total variation denoising and a single image dataset (u^{train}, f) . By considering a scale-dependent parameter $\lambda \in \mathbb{R}_+^n$, our bilevel problem then reads

$$\min_{\lambda \in \mathbb{R}_+^n} \quad J(u(\lambda), u^{\text{train}}) \tag{21}$$

$$\text{s.t} \quad u(\lambda) = \arg \min_{u \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^n \lambda_i \|u_i - f_i\|^2 + \sum_{i=1}^s \|(\mathbb{K}u)_i\| \tag{22}$$

where $\mathbb{K} : \mathbb{R}^n \rightarrow \mathbb{R}^{s \times 2}$ is the discrete gradient operator with respect to directions in x and y , i.e., $\mathbb{K}u = (K_x u, K_y u)$, where K_x and K_y correspond to the discrete partial derivative with respect to the horizontal and vertical direction, respectively. Thanks to the convexity of the energy function in the lower-level problem, we can replace the constraint by its necessary and sufficient optimality condition, yielding

$$\min_{\lambda \in \mathbb{R}_+^n} \quad J(u(\lambda), u^{\text{train}}) \tag{23}$$

$$\text{s.t} \quad \langle \lambda \circ (u - f), v - u \rangle + \sum_{i=1}^s \|(\mathbb{K}v)_i\| - \sum_{i=1}^s \|(\mathbb{K}u)_i\| \geq 0, \quad \forall v \in \mathbb{R}^n, \tag{24}$$

where \circ stands for the Hadamard product between vectors. This is an optimization problem constrained by a variational inequality of the second kind, along with non-negativity constraints for the parameter λ .

Moreover, using duality techniques, the variational inequality of the second kind in problem (23) can be equivalently written in primal-dual form, yielding the following reformulation of problem (21):

$$\begin{aligned} & \underset{(\lambda, u, q) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{s \times 2}}{\text{minimize}} && J(u, u^{\text{train}}) \\ & \text{subject to} && \lambda \circ (u - f) + \mathbb{K}^\top q = 0 \\ & && \langle q_j, (\mathbb{K}u)_j \rangle = \|(\mathbb{K}u)_j\|, \quad \forall j = 1, \dots, s \\ & && \|q_j\| \leq 1, \quad \forall j = 1, \dots, s \\ & && \lambda_j \geq 0, \quad \forall j = 1, \dots, n. \end{aligned} \tag{25}$$

Failure of Standard Constraint Qualification Conditions

A key goal in the study of an optimization problem is the derivation of optimality conditions that allow a proper characterization of stationary points. To do so, Lagrange multiplier’s existence theorems are usually proven on basis of the so-called constraint qualification conditions (Nocedal and Wright 2006). Next, we show that in the case of problem (23), the situation is not standard at all and classical optimization theory typically fails.

Even though the primal-dual reformulation transforms problem (23) into a constrained nonlinear optimization one, the difficulties related to the nonsmoothness remain in the constraints. One may try to circumvent this by considering a smooth reformulation of the restrictions in order to use standard nonlinear programming techniques. One possibility consists in rewriting (25) in the equivalent differentiable form:

$$\begin{aligned}
 & \min J(u, u^{\text{train}}) \\
 & \text{s.t } \lambda \circ (u - f) + \mathbb{K}^\top q = 0 \\
 & \quad \langle q_i, (\mathbb{K}u)_i \rangle^2 - \|(\mathbb{K}u)_i\|^2 = 0, \quad \forall i = 1, \dots, s \\
 & \quad -\langle q_i, (\mathbb{K}u)_i \rangle \leq 0, \quad \forall i = 1, \dots, s \\
 & \quad \|q_i\|^2 - 1 \leq 0, \quad \forall i = 1, \dots, s \\
 & \quad -\lambda_i \leq 0, \quad \forall i = 1, \dots, n,
 \end{aligned}$$

and trying to apply nonlinear programming results.

Considering a toy example where $u \in \mathbb{R}^2$, $\lambda \in \mathbb{R}$, $q \in \mathbb{R}^2$ and $\mathbb{K} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is defined by $\mathbb{K} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$, we may indeed analyze case-by-case and verify whether a standard constraint qualification has a chance to hold. To verify either the *Linear Independence Constraint Qualification Condition (LICQ)* or the *Mangasarian-Fromowitz Constraint Qualification Condition (MFCQ)* Nocedal and Wright (2006), we have to analyze the rank of the matrix formed by the gradients of the equality constraints, which is given by

$$\nabla h(u, q, \lambda) := \begin{pmatrix} \lambda & 0 & 2(u_1 - u_2)(q_1^2 - 1) & 0 \\ 0 & \lambda & -2(u_1 - u_2)(q_1^2 - 1) & 2u_2(q_2^2 - 1) \\ 1 & -1 & 2q_1(u_1 - u_2)^2 & 0 \\ 0 & 1 & 0 & 2q_2u_2^2 \\ u_1 - f_1 & u_2 - f_2 & 0 & 0 \end{pmatrix} \quad (26)$$

We then obtain the following cases:

$(\mathbb{K}u)_1 = 0, (\mathbb{K}u)_2 \neq 0$: In this case we know that $u_1 - u_2 = 0$ and the dual variable verifies $|q_2| = 1$. Consequently, $\nabla h_3(u, q, \lambda) = (0, 0, 0, 0, 0)^\top$ and, therefore, the columns of $\nabla h(u, q, \lambda)$ are not linearly independent, and neither LICQ nor MFCQ holds.

$(\mathbb{K}u)_1 \neq 0, (\mathbb{K}u)_2 = 0$: Similar than the previous case, we reach to the same violation of linear independence, with $\nabla h_4(u, q, \lambda)$ equal to zero.

$(\mathbb{K}u)_1 \neq 0, (\mathbb{K}u)_2 \neq 0$: In this case $|q_i| = 1, i = 1, 2$ and we obtain $\nabla h_3(u, q, \lambda) = (0, 0, 2q_1(u_1 - u_2)^2, 0, 0)^\top$ and $\nabla h_4(u, q, \lambda) = (0, 0, 0, 2q_2u_2^2, 0)^\top$.

The linear independence may be satisfied in this case, and existence of Lagrange multipliers may have a chance to be justified. This is, however, a case with scarce practical relevance. In the imaging setting, it would be related to completely smooth images (with no edges).

Alternative Optimality Conditions

From the discussion above, it becomes clear that standard constraint qualifications cannot be expected to hold for the type of bilevel problems at hand and, therefore, classical nonlinear programming results cannot be used for guaranteeing existence of Lagrange multipliers. As an alternative, nonsmooth analysis techniques may be

used to derive stationarity conditions, at the price of being possibly weaker than the ones originally expected.

In that sense, a first idea consists in carrying out a nonsmooth analysis of the solution operator associated to the lower-level problem. Indeed, it can be shown (De los Reyes and Meyer 2016; Hintermüller and Wu 2015) that the solution mapping $S : \mathbb{R}_+^n \rightarrow \mathbb{R}^n, \lambda \mapsto u$, for the lower-level problem is Bouligand differentiable, i.e., directionally differentiable and locally Lipschitz continuous. Using the chain rule for B-differentiable functions, the composite loss function is Bouligand differentiable as well (Dontchev and Rockafellar 2009). This implies that the problem (28) can be written in reduced form as

$$\min_{\lambda \in \mathbb{R}_+^n} \mathcal{J}(\lambda) = J(S(\lambda), \lambda),$$

and a stationarity condition for a local optimal solution λ^* is given by

$$\langle J_u(u^*, \lambda^*), \eta \rangle + \langle J_\lambda(u^*, \lambda^*), \lambda - \lambda^* \rangle \geq 0, \quad \forall \lambda \in \mathbb{R}_+^n, \tag{27}$$

where $u^* = S(\lambda^*)$ and $\eta := S'(\lambda^*; \lambda - \lambda^*)$ is the directional derivative of the solution mapping in direction $\lambda - \lambda^*$. Condition (27) is also known as *Bouligand (B-) stationarity*. Even though this stationarity condition is sharp, it is hardly usable due to the nonlinearity of the directional derivative.

A different approach is pursued in Outrata (2000), where the author reformulates problems such as (23) using a generalized equation:

$$\min_{\lambda \in \mathbb{R}_+^n} J(u(\lambda), u^{\text{train}}) \tag{28}$$

$$\text{s.t.} \quad 0 \in \lambda \circ (u - f) + Q(u), \tag{29}$$

with $Q : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ a multifunction with a closed graph defined by

$$Q(u) := \left\{ \mathbb{K}^\top q : q \in \mathbb{R}^{s \times 2} : \begin{cases} q_j = \frac{(\mathbb{K}u)_j}{\|(\mathbb{K}u)_j\|}, & \text{if } (\mathbb{K}u)_j \neq 0, \\ \|q_j\| \leq 1, & \text{if } (\mathbb{K}u)_j = 0. \end{cases} \right\}$$

This problem may be interpreted as a *Generalized Mathematical Program with Equilibrium Constraints*, and Mordukhovich variational analysis may be used to derive first-order necessary optimality conditions (Outrata 2000, Theorem 3.1). To this aim, let us introduce the computed Mordukhovich normal cone (see, e.g., Hintermüller and Wu 2015):

$$N_{GphQ}^M(u, \mathbb{K}^\top q) = \left\{ (\mathbb{K}^\top w, v) : \begin{cases} \|(\mathbb{K}u)_j \| w_j = (\mathbb{K}v)_j - \langle (\mathbb{K}v)_j, q_j \rangle q_j, & \text{if } (\mathbb{K}u)_j \neq 0, \\ (\mathbb{K}v)_j = 0, & \text{if } \|q_j\|_2 < 1, \\ (\mathbb{K}v)_j = 0, \vee \\ (\mathbb{K}v)_j = cq_j (c \in \mathbb{R}), \langle w_j, q_j \rangle = 0 \vee \\ (\mathbb{K}v)_j = cq_j (c \geq 0), \langle w_j, q_j \rangle \geq 0. \end{cases} \text{if } (\mathbb{K}u)_j = 0, \|q_j\|_2 = 1 \right\}$$

Let (λ^*, u^*, q^*) be a local solution of problem (28), and let $(\mathbb{K}^\top w, v) \in N_{GphQ}^M(u^*, \mathbb{K}^\top q^*)$ be a solution of the system

$$\begin{pmatrix} 0 & -diag(u^* - f) \\ I & -diag(\lambda^*) \end{pmatrix} \begin{pmatrix} \mathbb{K}^\top w \\ v \end{pmatrix} \in \{0\} \times N_{\mathbb{R}_+^n}^M \tag{30}$$

The vector (λ^*, u^*, q^*) is said to satisfy the *constraint qualification* if $\mathbb{K}^\top w = 0$ and $v = 0$ is the unique solution to the problem above.

Under this constraint qualification, there exist Lagrange multipliers $(\mathbb{K}^\top \varphi, p, \vartheta)$ such that the following *Mordukhovich (M-) stationary* system holds true:

$$\lambda \circ (u^* - f) + \mathbb{K}^\top q^* = 0, \tag{31a}$$

$$\langle q_j^*, (\mathbb{K}u^*)_j \rangle = \| (\mathbb{K}u^*)_j \|^2, \quad \forall i = 1, \dots, s, \tag{31b}$$

$$\|q_j^*\| \leq 1, \quad \forall j = 1, \dots, s, \tag{31c}$$

$$\lambda \circ p + \mathbb{K}^\top \varphi = \nabla_u J(u^*), \tag{31d}$$

$$(u^* - f) \circ p + \vartheta = 0, \tag{31e}$$

$$\| (\mathbb{K}u^*)_j \| \varphi_j = (\mathbb{K}p)_j - \langle (\mathbb{K}p)_j, q_j^* \rangle q_j^*, \quad \text{if } (\mathbb{K}u^*)_j \neq 0, \tag{31f}$$

$$(\mathbb{K}p)_j = 0, \quad \text{if } (\mathbb{K}u^*)_j = 0, \|q_j^*\| < 1, \tag{31g}$$

$$\left. \begin{aligned} (\mathbb{K}p)_j &= 0 \vee \\ (\mathbb{K}p)_j &= cq_j^* (c \in \mathbb{R}), \langle \varphi_j, q_j^* \rangle = 0 \vee \\ (\mathbb{K}p)_j &= cq_j^* (c \geq 0), \langle \varphi_j, q_j^* \rangle \geq 0, \end{aligned} \right\} \text{if } (\mathbb{K}u^*)_j = 0, \|q_j^*\| = 1, \tag{31h}$$

$$0 \leq \lambda \perp \vartheta \geq 0, \tag{31i}$$

The difference between M-stationarity and strong stationarity systems concerns the information about the multipliers on the so-called biactive set $\mathcal{B} = \{j \in \{1, \dots, s\} : (\mathbb{K}u)_j = 0, \|q_j\| = 1\}$. The biactive characterization of those multipliers in (31h) is actually weaker than in a strong stationarity system.

An even weaker stationarity system may be obtained by regularizing the Euclidean norm in (23) and then deriving optimality conditions for each regularized problem and afterward passing to the limit in the regularized optimality systems (De los Reyes 2011). In that case, a Clarke (*C*-) stationary system is obtained, where (31h) is replaced by

$$(\mathbb{K}p^*)_j = cq_j^*(c \in \mathbb{R}), \langle \varphi_j^*, q_j^* \rangle \geq 0, \quad \text{if } (\mathbb{K}u^*)_j = 0, \|q_j^*\| = 1. \quad (32)$$

Finally, it can be proven that if *strict complementarity holds*, i.e., if the biactive set is empty, all strong, B-, M-, and C-stationarity conditions are equivalent (see, e.g., De los Reyes 2015; De los Reyes and Meyer 2016).

Solution Algorithms

When dealing with the numerical optimization of the bilevel problem, the solution of a regularized version of (28) appears to be the more frequent approach. In this line, the nonsmoothness is regularized by means of a differentiable function, and nonlinear optimization methods are then applied. In De los Reyes and Schönlieb (2013), for instance, the authors implement a BFGS algorithm with Armijo backtracking to solve a regularized bilevel problem for image denoising. Alternatively, the authors in Hintermüller and Wu (2015) propose a projected gradient method to find stationary points in the case of blind deconvolution.

For dealing with the nonsmooth bilevel problem, we point out the works (Oustrata and Zowe 1995) and (Christof et al. 2020). In the first one, subgradients of the reduced cost function are computed by means of a generalized adjoint equation, while, in the second one, a trust-region method exploiting the nonsmooth Bouligand subdifferential properties of the solution operator is proposed. Both algorithms are precisely devised for optimization problems with variational inequality constraints, and convergence toward a C-stationary point is verified in the second one.

Infinite-Dimensional Case

The infinite-dimensional counterpart of the bilevel learning approach (16) poses additional difficulties in the analysis of the resulting nonsmooth problems, since properties like directional differentiability of the solution mapping cannot be derived in function spaces, unless very restrictive assumptions are made (De los Reyes and Meyer 2016).

The study of the infinite-dimensional problems becomes important, however, to derive properties which are resolution independent, as well as to shed light on the development of algorithms whose efficiency does not depend on the number of pixels of the image. Moreover, in the recent past, the use of parameter functions,

instead of vectors, has proven to be superior for different imaging tasks, and, in order to consider spatially dependent parameters, the function space framework appears indeed to be the natural choice in this context.

Considering as image domain the open bounded convex set $\Omega \subset \mathbb{R}^2$ and assuming that the noisy image f lies in the Hilbert $Y = L^2(\Omega)$, the bilevel problem, for a single training pair, consists in searching for parameters $\lambda = (\lambda_1, \dots, \lambda_M)$ and $\alpha = (\alpha_1, \dots, \alpha_N)$ in abstract nonnegative parameter sets \mathcal{P}_λ^+ and \mathcal{P}_α^+ that solve

$$\min_{\alpha \in \mathcal{P}_\alpha^+, \lambda \in \mathcal{P}_\lambda^+} J(u_{\alpha, \lambda}) \quad \text{s.t.} \quad u_{\alpha, \lambda} \in \arg \min_{u \in X} \mathcal{E}(u; \lambda, \alpha), \quad (\text{P})$$

with

$$\mathcal{E}(u; \lambda, \alpha) := \sum_{i=1}^M \int_{\Omega} \lambda_i(x) \phi_i(x, [Au](x)) dx + \sum_{j=1}^N \int_{\Omega} \alpha_j(x) d|B_j u|(x).$$

where the loss functional $J : X \rightarrow \mathbb{R}$ is assumed to be convex, proper, and weak* lower semicontinuous. Our solution u lies in an abstract space X , mapped by the linear operator A to Y . Depending on B , A , and the ϕ_i , different problems as well as assumptions have to be made (De los Reyes et al. 2016). In general, convexity of $\mathcal{E}(\cdot; \lambda, \alpha)$ and compactness properties in the space of functions of bounded variation turn out to be crucial for proving existence of optimal solutions.

To overcome the difficulties related to the nonsmoothness of (P) and the lack of regularity of the solutions, smoothing terms are usually added within the bilevel framework in order to carry out the analysis. For one, we require Huber regularization of the Radon norms. This is needed for the single-valued differentiability of the solution map $(\lambda, \alpha) \mapsto u_{\alpha, \lambda}$. Secondly, we take a convex, proper, and weak* lower-semicontinuous smoothing functional $H : X \rightarrow [0, \infty]$. The typical choice is the elliptic energy $H(u) = \frac{1}{2} \|\nabla u\|^2$.

For parameters $\mu \geq 0$ and $\gamma \in (0, \infty]$, we consider as in De los Reyes et al. (2016) the problem

$$\min_{\alpha \in \mathcal{P}_\alpha^+, \lambda \in \mathcal{P}_\lambda^+} J(u_{\alpha, \lambda, \gamma, \mu}) \quad \text{s.t.} \quad u_{\alpha, \lambda, \gamma, \mu} \in \arg \min_{u \in X \cap \text{dom } \mu H} \mathcal{E}^{\gamma, \mu}(u; \lambda, \alpha) \quad (\text{P}^{\gamma, \mu})$$

with the regularized energy

$$\begin{aligned} \mathcal{E}^{\gamma, \mu}(u; \lambda, \alpha) := & \mu H(u) + \sum_{i=1}^M \int_{\Omega} \lambda_i(x) \phi_i(x, [Au](x)) dx \\ & + \sum_{j=1}^N \int_{\Omega} \alpha_j(x) d|B_j u|_{\gamma}(x). \end{aligned}$$

We denote by $|B_j u|_\gamma$ the Huberised total variation measure, where

$$|g|_\gamma = \begin{cases} \|g\|_2 - \frac{1}{2\gamma}, & \|g\|_2 \geq 1/\gamma, \\ \frac{\gamma}{2} \|g\|_2^2, & \|g\|_2 < 1/\gamma, \end{cases}$$

for $\gamma \in (0, \infty]$. Considering the Lebesgue decomposition of $\nu \in \mathcal{M}(\Omega; \mathbb{R}^n)$ into the absolutely continuous part $f \mathcal{L}^n$ and the singular part ν^s , we set

$$|\nu|_\gamma(V) := \int_V |f(x)|_\gamma \, dx + |\nu^s|(V), \quad (V \in \mathcal{B}(\Omega)).$$

The measure $|\nu|_\gamma$ corresponds to the Huber regularization of the total variation measure $|\nu|$.

Existence and Other Properties

The first questions to be answered concerning the bilevel problem (P) are related to the existence of optimal parameters as well as the structure of the optimizers. At least partially, some answers to these inquires have been given in De los Reyes et al. (2016) (see also the review paper Calatroni et al. 2017). We briefly summarize next the main results obtained in those references.

Considering the particular, but frequent, setup with quadratic loss functional and fidelity term

$$J(u) = \frac{1}{2} \|Au - u^{\text{train}}\|_{L^2(\Omega)}^2, \quad \text{and} \quad \phi_1(x, v) = \frac{1}{2} |f(x) - v|^2, \quad (33)$$

and with $M = 1$ and $\mathcal{P}_\lambda^+ = \{1\}$, we may obtain conditions for positivity of the parameters $\alpha = (\alpha_1, \dots, \alpha_N) \in \mathcal{P}_\alpha^+ = [0, \infty]^N$. In fact, suppose that $f, f_0 \in \text{BV}(\Omega) \cap L^2(\Omega)$ satisfy

$$\text{TV}(f) > \text{TV}(u^{\text{train}}), \quad (34)$$

then there exist $\bar{\mu}, \bar{\gamma} > 0$ such that any optimal solution $\alpha_{\gamma, \mu} \in [0, \infty]$ to the problem

$$\min_{\alpha \in [0, \infty]} \frac{1}{2} \|u^{\text{train}} - u_\alpha\|_{L^2(\Omega)}^2$$

with

$$u_\alpha \in \arg \min_{u \in \text{BV}(\Omega)} \left(\frac{1}{2} \|f - u\|_{L^2(\Omega)}^2 + \alpha |Du|_\gamma(\Omega) + \frac{\mu}{2} \|\nabla u\|_{L^2(\Omega; \mathbb{R}^n)}^2 \right)$$

satisfies $\alpha_{\gamma,\mu} > 0$, whenever $\mu \in [0, \bar{\mu}]$ and $\gamma \in [\bar{\gamma}, \infty]$. The choice $\gamma = \infty$ should be understood as the standard unregularized total variation measure or norm.

For fixed values $\gamma < \infty$ and $\mu > 0$, existence of an optimal parameter can be proven by the direct method of the calculus of variations. What condition 34 guarantees is existence of an optimal interior solution $\alpha > 0$ to (P) without any additional box constraints. Moreover, condition (34) also guarantees convergence of optimal parameters of the numerically regularized H^1 problems $(P^{\gamma,\mu})$ to a solution of the original BV(Ω) problem (P).

A similar structural result may be obtained for second-order total generalized variation Gaussian denoising, again assuming that the noisy data has to oscillate more in terms of TGV² than the ground truth does. Specifically, if the data $f, u^{\text{train}} \in L^2(\Omega) \cap \text{BV}(\Omega)$ satisfies for some $\alpha_2 > 0$ the condition

$$\text{TGV}_{(\alpha_2,1)}^2(f) > \text{TGV}_{(\alpha_2,1)}^2(u^{\text{train}}), \tag{35}$$

then there exists $\bar{\mu}, \bar{\gamma} > 0$ such that any optimal solution $\alpha_{\gamma,\mu} = ((\alpha_{\gamma,\mu})_1, (\alpha_{\gamma,\mu})_2)$ to the problem

$$\min_{\alpha \in [0,\infty]^2} \frac{1}{2} \|f_0 - v_\alpha\|_{L^2(\Omega)}^2$$

with

$$\begin{aligned} (v_\alpha, w_\alpha) \in \arg \min_{\substack{v \in \text{BV}(\Omega) \\ w \in \text{BD}(\Omega)}} & \left(\frac{1}{2} \|f - v\|_{L^2(\Omega)}^2 + \alpha_1 |Dv - w|_\gamma(\Omega) + \alpha_2 |Ew|_\gamma(\Omega) \right. \\ & \left. + \frac{\mu}{2} \|(\nabla v, \nabla w)\|_{L^2(\Omega; \mathbb{R}^n \times \mathbb{R}^{n \times n})}^2 \right) \end{aligned}$$

satisfies $(\alpha_{\gamma,\mu})_1, (\alpha_{\gamma,\mu})_2 > 0$, whenever $\mu \in [0, \bar{\mu}]$, $\gamma \in [\bar{\gamma}, \infty]$. Observe that we allow for infinite parameters α .

Additionally, a result on the approximation properties as $\gamma \nearrow \infty$ and $\mu \searrow 0$ is also obtained. In fact, for both previous settings, there exist $\bar{\gamma} \in (0, \infty)$ and $\bar{\mu} \in (0, \infty)$ such that the solution map $(\gamma, \mu) \mapsto \alpha_{\gamma,\mu}$ is outer semicontinuous within $[\bar{\gamma}, \infty] \times [0, \bar{\mu}]$. Roughly, the outer semicontinuity (Rockafellar and Wets 1998) of the solution map means that as the regularization vanishes, any optimal parameters for the regularized models $(P^{\gamma,\mu})$ tend to some optimal parameters of the original model (P).

Stationarity Conditions

The family of problems $(P^{\gamma,\mu})$ constitute PDE-constrained optimization instances, and, therefore, suitable techniques from this field may be utilized to derive optimality conditions. For the limiting cases $\gamma \rightarrow \infty$ or $\mu \rightarrow 0$, an additional

asymptotic analysis needs to be performed in order to get stationarity conditions for the optimal solutions.

Several instances of the abstract problem $(P^{\gamma,\mu})$ have been individually considered in previous contributions. The case with total variation regularization was considered in De los Reyes and Schönlieb (2013) in presence of several noise models, and, after proving the Gâteaux differentiability of the solution operator, an optimality system was derived. Thereafter, an asymptotic analysis with respect to $\gamma \rightarrow \infty$ was carried out (with $\mu > 0$), obtaining an optimality system for the corresponding problem. In that case the optimization problem corresponds to one with variational inequality constraints, and the characterization concerns C-stationary points. Differentiability properties of higher-order regularization solution operators were also investigated in De los Reyes et al. (2017), with the corresponding first-order optimality conditions.

For the general problem $(P^{\gamma,\mu})$, using the Lagrangian formalism, the following optimality system is obtained:

$$\begin{aligned} \mu \int_{\Omega} \langle \nabla u, \nabla v \rangle dx + \sum_{i=1}^M \int_{\Omega} \lambda_i \phi'_i(Au) Av dx \\ + \sum_{j=1}^N \int_{\Omega} \alpha_j \langle h_{\gamma}(B_j u), B_j v \rangle dx = 0, \quad \forall v \in V, \end{aligned} \tag{36}$$

$$\begin{aligned} \mu \int_{\Omega} \langle \nabla p, \nabla v \rangle dx + \sum_{i=1}^M \int_{\Omega} \langle \lambda_i \phi''_i(Au) Ap, Av \rangle dx \\ + \sum_{j=1}^N \int_{\Omega} \alpha_j \langle h_{\gamma}^{*'}(B_j u) B_j p, B_j v \rangle dx = -\nabla_u J(u)v, \quad \forall v \in V, \end{aligned} \tag{37}$$

$$\int_{\Omega} \phi_i(Au) Ap(\zeta - \lambda_i) dx \geq 0, \quad \forall \zeta \geq 0, \quad i = 1, \dots, M, \tag{38}$$

$$\int_{\Omega} h_{\gamma}(B_j u) B_j p(\eta - \alpha_j) dx \geq 0, \quad \forall \eta \geq 0, \quad j = 1, \dots, N, \tag{39}$$

where V stands for the Sobolev space where the regularized image lives (typically a subspace of $H^1(\Omega; \mathbb{R}^m)$), $p \in V$ stands for the adjoint state, and h_{γ} is a regularized version of the TV subdifferential, e.g.,

$$h_{\gamma}(z) := \begin{cases} \frac{z}{|z|} & \text{if } \gamma|z| - 1 \geq \frac{1}{2\gamma} \\ \frac{z}{|z|} (1 - \frac{\gamma}{2}(1 - \gamma|z| + \frac{1}{2\gamma})^2) & \text{if } \gamma|z| - 1 \in (-\frac{1}{2\gamma}, \frac{1}{2\gamma}) \\ \gamma z & \text{if } \gamma|z| - 1 \leq -\frac{1}{2\gamma}. \end{cases} \tag{40}$$

The rigorous derivation of the optimality system has to be justified for each specific combination of spaces, regularizers, noise models, and cost functionals.

With help of the adjoint equation (37), also gradient formulas for the reduced cost functional $\mathcal{J}(\lambda, \alpha) := J(u_{\alpha,\lambda}, \lambda, \alpha)$ are derived:

$$(\nabla_{\lambda} \mathcal{J})_i = \int_{\Omega} \phi_i(Au) A p \, dx, \quad (\nabla_{\alpha} \mathcal{J})_j = \int_{\Omega} h_{\gamma}(B_j u) B_j p \, dx, \quad (41)$$

for $i = 1, \dots, M$ and $j = 1, \dots, N$. The gradient information is of numerical importance in the design of solution algorithms. In the case of finite dimensional parameters, thanks to the structure of the minimizers reviewed previously, the corresponding variational inequalities (38)-(39) turn into equalities. This has important numerical consequences, since in such cases the gradient formulas (41) may be used without additional projection steps.

Dualization

An alternative technique for studying the bilevel problem, via duality, was proposed by Hintermüller and coauthors (2017), where the lower-level problem is replaced by its pre-dual version. In the case of total variation and with a weight solely on the regularizer, the bilevel problem becomes

$$\begin{aligned} & \min_{\underline{\alpha} \leq \alpha(x) \leq \bar{\alpha}} J(R(\operatorname{div} p)) + \frac{\beta}{2} \|\alpha\|_{H^1(\Omega)}^2 & (D) \\ \text{s.t. } p \in & \arg \min_{p \in \mathbb{H}_0^1(\Omega): |p(x)|_{\infty} \leq \alpha(x)} \left(\frac{\mu}{2} \|\nabla p\|_{L^2(\Omega)}^2 + \frac{\gamma}{2} \|p\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\operatorname{div} p + f\|_{L^2(\Omega)}^2 \right), \end{aligned}$$

where $\mu, \gamma > 0$ are regularization parameters and R stands for the localized residual function. As a consequence, the necessary and sufficient optimality condition for the lower-level problem becomes a variational inequality of the first kind, which may be reformulated as a complementarity system as well. The abstract problem then constitutes a *mathematical program with equilibrium constraints* in function space.

The treatment of this problem is, however, by no means any easier than the primal bilevel one. In fact, in order to carry out the analysis, the authors have to penalize the pointwise box constraint by means of a Moreau-Yosida function $\mathcal{P}_{\delta}(p, \alpha)$, yielding the problem

$$\begin{aligned} & \min_{\underline{\alpha} \leq \alpha(x) \leq \bar{\alpha}} J(R(\operatorname{div} p)) + \frac{\beta}{2} \|\alpha\|_{H^1(\Omega)}^2 \\ \text{s.t. } p \in & \arg \min_{p \in \mathbb{H}_0^1(\Omega)} \left(\frac{\mu}{2} \|\nabla p\|_{L^2(\Omega)}^2 + \frac{\gamma}{2} \|p\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\operatorname{div} p + f\|_{L^2(\Omega)}^2 + \frac{1}{\epsilon} \mathcal{P}_{\delta}(p, \alpha) \right), \end{aligned}$$

For each penalized problem, existence of Lagrange multipliers is then proven using standard Karush-Kuhn-Tucker theory in function spaces. Although no limit analysis is carried out in order to get an optimality system for problem (D), the authors provide some useful density and stability results.

Nonlocal Problems

As mentioned in section “Variational Inverse Problems Setting”, nonlocal models perform particularly well in problems where different textures are present in the image, as similar intensity patterns between pixels or patches in a given spatial subdomain are taken into account for the restoration (Yaroslavsky 1986; Tomasi and Manduchi 1998; Buades et al. 2005). In Gilboa and Osher (2007) and Gilboa and Osher (2008), an energy-based variational framework was introduced for nonlocal imaging models, allowing the analytical study of different underlying properties in function spaces. Moreover, nonlocal vector calculus has been developed in the last years, providing a very useful analytical toolbox for dealing with nonlocal models arising in different application areas (Gunzburger and Lehoucq 2010; Du et al. 2012).

Within this framework, a bilevel learning formulation for estimating the weights in nonlocal imaging problems was recently studied in d’Elia et al. (2019), considering both the case of a weight within the kernel and the one with the weight in front of the fidelity term. Assuming there is a single training pair of a clean and a noisy images (u^{train}, f) , the general problem reads as follows:

$$\min_{0 \leq \lambda, w \leq U} J(u) \tag{42}$$

$$\text{s.t. } u(\lambda, w) = \arg \min_u \frac{1}{2} \int_{\Omega} \int_{\Omega} (u(\mathbf{x}) - u(\mathbf{y}))^2 \gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} + \int_{\Omega} \lambda (u - f)^2 \tag{43}$$

where

$$\gamma_w(\mathbf{x}, \mathbf{y}) := \exp \left\{ - \int_{B_{\rho}(0)} w(\boldsymbol{\tau}) (f(\mathbf{x} + \boldsymbol{\tau}) - f(\mathbf{y} + \boldsymbol{\tau}))^2 \, d\boldsymbol{\tau} \right\} \chi(\mathbf{y} \in B_{\epsilon}(\mathbf{x}))$$

corresponds to the modified nonlocal means kernel. Alternative nonlocal kernels, pixelwise or patchwise, may be considered as well.

In this case, the unique solution to the lower-level problem belongs to the space $V_c^w := \{v \in L^2(\Omega \cup \Omega_I) : \|v\|_{V^w} < \infty, v|_{\Omega_I} = 0\}$, where $\Omega_I := \{\mathbf{y} \in \mathbb{R}^d \setminus \Omega : \|\mathbf{x} - \mathbf{y}\| \leq \epsilon, \forall \mathbf{x} \in \Omega\}$ is the so-called interaction domain where volume constraints are imposed, and

$$\|u\|_{V^w}^2 := \int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (u(\mathbf{x}) - u(\mathbf{y}))^2 \gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x}$$

is the nonlocal energy norm. If w is a constant weight, the space is simply denoted as V .

Existence of an optimal solution for the bilevel problem in each of the settings has been proven in d'Elia et al. (2019), under the inclusion of box constraints for the parameters. For the case of a spatially dependent coefficient in front of the fidelity, an extra Tikhonov regularization term has to be added to the loss functional to get existence of an optimal solution.

In contrast to the variational regularizers reviewed before, for the nonlocal problem (43), Gâteaux differentiability of the solution operator can be demonstrated. As a consequence, necessary optimality systems that characterize strong stationary points can be established in each of the cases (see d'Elia et al. 2019 for further details).

For the case when a spatially dependent weight $\lambda \in H^1(\Omega)$ is optimized, while keeping the kernel fixed, a necessary optimality condition is given by the following complementarity problem:

$$\int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (u(\mathbf{x}) - u(\mathbf{y}))(\psi(\mathbf{x}) - \psi(\mathbf{y}))\gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} + \int_{\Omega} \lambda (u - f) \psi \, d\mathbf{x} = 0, \quad \forall \psi \in V_c, \quad (44a)$$

$$\int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (p(\mathbf{x}) - p(\mathbf{y}))(\phi(\mathbf{x}) - \phi(\mathbf{y}))\gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} + \int_{\Omega} \lambda p \phi \, d\mathbf{x} = -\nabla_u J(u)\phi, \quad \forall \phi \in V_c, \quad (44b)$$

$$\begin{aligned} -\beta \Delta \lambda + \beta \lambda &= \sigma_{\Omega}^+ - \sigma_{\Omega}^- \quad \text{in } \Omega, \\ \beta \frac{\partial \lambda}{\partial \mathbf{n}} &= \sigma_{\Gamma}^+ - \sigma_{\Gamma}^- \quad \text{on } \Gamma, \end{aligned} \quad (44c)$$

$$\begin{aligned} 0 \leq \sigma_{\Omega}^+(\mathbf{x}) \perp \lambda(\mathbf{x}) \geq 0, \quad 0 \leq \sigma_{\Omega}^-(\mathbf{x}) \perp (U - \lambda(\mathbf{x})) \geq 0, \quad \forall \mathbf{x} \in \Omega, \\ 0 \leq \sigma_{\Gamma}^+(\mathbf{x}) \perp \lambda(\mathbf{x}) \geq 0, \quad 0 \leq \sigma_{\Gamma}^-(\mathbf{x}) \perp (U - \lambda(\mathbf{x})) \geq 0, \quad \forall \mathbf{x} \in \Gamma, \end{aligned} \quad (44d)$$

where σ_{Ω}^+ , $-\sigma_{\Omega}^-$, σ_{Γ}^+ , σ_{Γ}^- are Karush-Kuhn-Tucker multipliers associated to the box constraints. As can be observed, in this case, the optimality system couples local and nonlocal systems of equations with additional pointwise complementarity conditions.

On the other hand, for the case when the weight within the kernel $w \in L^2(B_{\rho}(\mathbf{0}))$ is optimized, the following optimality system is satisfied:

$$\int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (u(\mathbf{x}) - u(\mathbf{y}))(\psi(\mathbf{x}) - \psi(\mathbf{y}))\gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} + \int_{\Omega} \lambda (u - f) \psi \, d\mathbf{x} = 0, \quad \forall \psi \in V_c, \quad (45a)$$

$$\int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} (p(\mathbf{x}) - p(\mathbf{y}))(\phi(\mathbf{x}) - \phi(\mathbf{y}))\gamma_w(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \, d\mathbf{x} + \int_{\Omega} \lambda p \phi \, d\mathbf{x} = -J'(u)\phi, \quad \forall \phi \in V_c, \quad (45b)$$

$$\int_{\Omega \cup \Omega_I} \int_{\Omega \cup \Omega_I} [(u(w) - u(w'))(p - p')\tilde{\gamma}_{(h-w)}(\mathbf{x}, \mathbf{y})] \, d\mathbf{y} \, d\mathbf{x} \geq 0, \quad \forall h \in \mathcal{U}_{ad}. \quad (45c)$$

with $\mathcal{U}_{ad} := \{v \in L^2(B_\rho(\mathbf{0})) : 0 \leq w(\mathbf{t}) \leq U, \forall \mathbf{t} \in B_\rho(\mathbf{0})\}$ and the linearized kernel

$$\tilde{\gamma}_h(\mathbf{x}, \mathbf{y}) = \gamma_w(\mathbf{x}, \mathbf{y}) \int_{B_\rho(\mathbf{0})} -h(\boldsymbol{\tau})(f(\mathbf{x} + \boldsymbol{\tau}) - f(\mathbf{y} + \boldsymbol{\tau}))^2 \, d\boldsymbol{\tau}. \quad (45d)$$

In this case, even if “only” a scalar is determined, the computational cost becomes high since in principle the kernel changes in each iteration of any solution algorithm and, with it, the assembly of the nonlocal interaction matrix, which is in principle a dense one.

Neural Network Optimization

In previous sections, we presented variational image restoration as a special class of ill-posed inverse problems and considered a bilevel learning framework for determining the different parameters involved. This approach allows to incorporate a priori information about the solution, enabling a deeper understanding of the models at hand. Even though this setting allows us to get insight into the structural properties of the variational models, finding the optimal solution is often highly computationally demanding and in some cases not suitable for real-world applications.

On the other hand, neural networks and, in particular, convolutional neural networks (CNN) have been widely used for image restoration tasks, such as denoising (Burger et al. 2012), blind and non-blind deblurring (Xu et al. 2014), demosaicking (Wang 2014), and super-resolution (Dong et al. 2014), among others. Despite such success in practical cases, these learning structures still lack explainability and reliability (Szegedy et al. 2013). Indeed, the incorporation of a priori knowledge in these models is very complicated, and in most applications, it is treated as a black box.

Recently, the gap between both frameworks has started to be bridged. By using bilevel optimization and optimal control ideas, some approaches that combine the

best properties of variational models and neural networks have been proposed. We provide next a brief review on a few of them, with the sole purpose of highlighting the importance of these connections.

Deep Neural Networks as a Further Regularizer

Even though we have previously detailed bilevel learning strategies for variational problems, recently also bilevel optimization approaches that make use of neural networks have been proposed. In particular, *Deep Bilevel Optimization Neural Networks (BOONet)*, introduced by Antil and coworkers (2020), develop a strategy for finding optimal regularization parameters based on a bilevel optimization problem. Here, an upper-level optimization problem is used to measure the reconstruction error on a training dataset, while the lower-level problem measures the misfit of the data reconstruction. This reconstruction is based on a generalized regularizer that has a network-like structure, leading to insightful comparisons over regularizers and activation functions used in neural networks.

Now, regarding the regularization term in (2), it has been further improved recently by making use of a pretrained CNN. Indeed, in Lunz et al. (2018) a data-driven regularizer is built using modern generative adversarial network principles, leading to the *neural network Tikhonov (NETT)* approach, where a pretrained network is composed with a regularization functional (Li et al. 2020).

In Kobler et al. (2020), a different technique for learning regularizers is proposed, called *total deep variation*. In this case, the regularizer is built using a multi-scale convolutional neural network, which training is based on a sampled optimal control problem interpretation. This formulation allows the authors to provide a sensitivity analysis of the learned coefficients with respect to the training dataset. It is worth mentioning that this regularizer can be trained using a different dataset than the application at hand, resembling the properties of transfer learning strategies.

Deep Unrolling Within Optimization

Assuming we use an iterative scheme for solving (2) that is based on a proximal operator

$$\text{prox}_{\tau\mathcal{R}}(\hat{u}) = \arg \min_{u \in \mathbb{R}^n} \frac{1}{2\tau} \|u - \hat{u}\|^2 + \mathcal{R}(u), \quad (46)$$

it was observed in Venkatakrishnan et al. (2013) that this denoising subproblem may be replaced by a more sophisticated neural network such as *BM3D* (Dabov et al. 2007). However, this general purpose approach does not exploit the variational structure of the original problem, and, thus, the explainability provided by classical variational approaches is missing.

An alternative strategy consists in using deep neural network architectures to replace inner operators (such as prox, gradients, etc.) of an optimization scheme to

solve the variational imaging task of interest. Even though this training process is computationally expensive, this procedure is often performed as an off-line batch operation. Once trained, the network evaluation is less expensive when used in the reconstruction process. Gregor and LeCun (2010) were able to incorporate these ideas into an ISTA iterative scheme for solving a sparse coding problem (LISTA). This procedure is based on “unrolling” the iterative scheme and replacing its explicit updates with learned ones. Hersey et al. (2014) propose to unfold the iterations into a layer-based structure similar to a neural network with application to learning optimal parameters of Markov random fields and nonnegative matrix factorization.

In the imaging context, these ideas have been considered in the unrolling of iterative schemes for problems with the structure presented in (11), such as proximal gradient (Adler and Öktem 2017), primal-dual hybrid gradient (Adler and Öktem 2018), or ADMM (Sun et al. 2016). This technique generates new tailor-designed deep neural network architectures that make use of the structure within the problem at hand. In the case of a Field of Experts (FoE) regularizer, this approach led to the development of *variational networks* (Kobler et al. 2017), where the authors rely on unrolling a proximal gradient descent step for a finite number of iterations and the connections to residual neural networks (He et al. 2016) are highlighted.

Numerical Experiments

In this section, we compare different bilevel parameter optimization techniques for image denoising, both from a reconstruction quality perspective and from a *learning* point of view. In particular, we will learn optimal scalar and scale-dependent parameters for image denoising models using total variation, total generalized variation, and nonlocal means regularizers. For the scalar experiment, we will make use of a semismooth Newton solver for the corresponding lower-level problem and the proposed BFGS method described in De los Reyes et al. (2017) for the bilevel problem. Moreover, the scale- and patch-dependent experiments will be solved using the primal-dual hybrid gradient (PDHG) method for the lower-level problem, and a trust-region strategy will be implemented for finding stationary points in the bilevel problem along the lines of Christof et al. (2020).

For obtaining such parameters, we will use a dataset of faces based on the popular CelebA Faces. This dataset will be split into a *training* dataset that will be used to learn the optimal parameters and a *validation* dataset that will be used to estimate the generalization error, i.e., to get an idea of the performance of the learned parameter in a set not previously trained on. Both datasets are generated by converting the original images in black and white, balancing its contrast and adding Gaussian noise of different intensities. A subset of the used training images is depicted in Fig. 2.

As a first experiment, we use a bilevel strategy for learning a scalar parameter for the variational formulation presented in (19) and taking the particular cases of total variation (TV) and total generalized variation (TGV). In Fig. 3, a subset of the validation dataset is shown with the optimal parameter and the corresponding SSIM measure. As extensively reported in De los Reyes et al. (2017), TGV is superior



Fig. 2 Sample of the training CelebA dataset

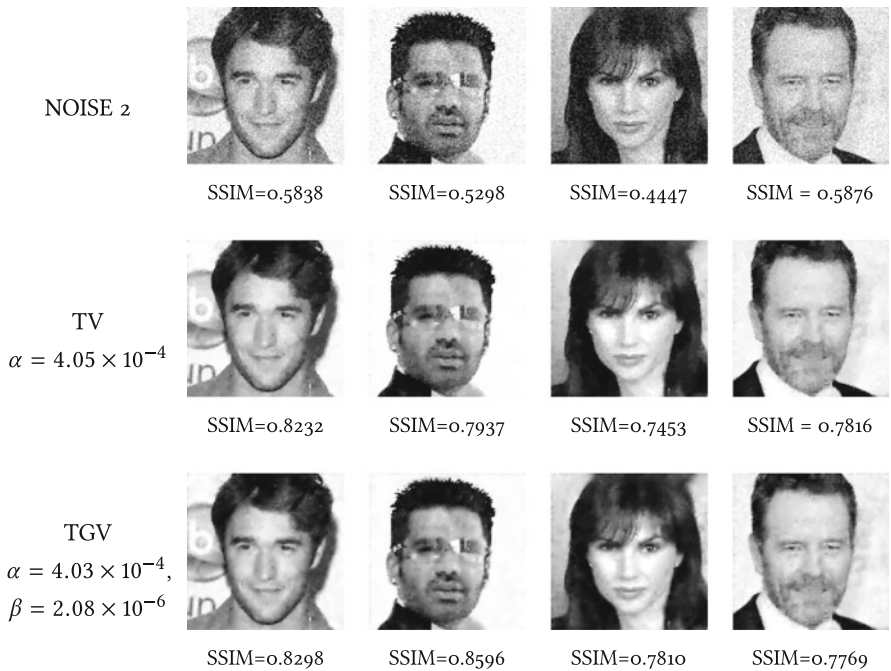


Fig. 3 Scalar regularization parameter: TV and TGV

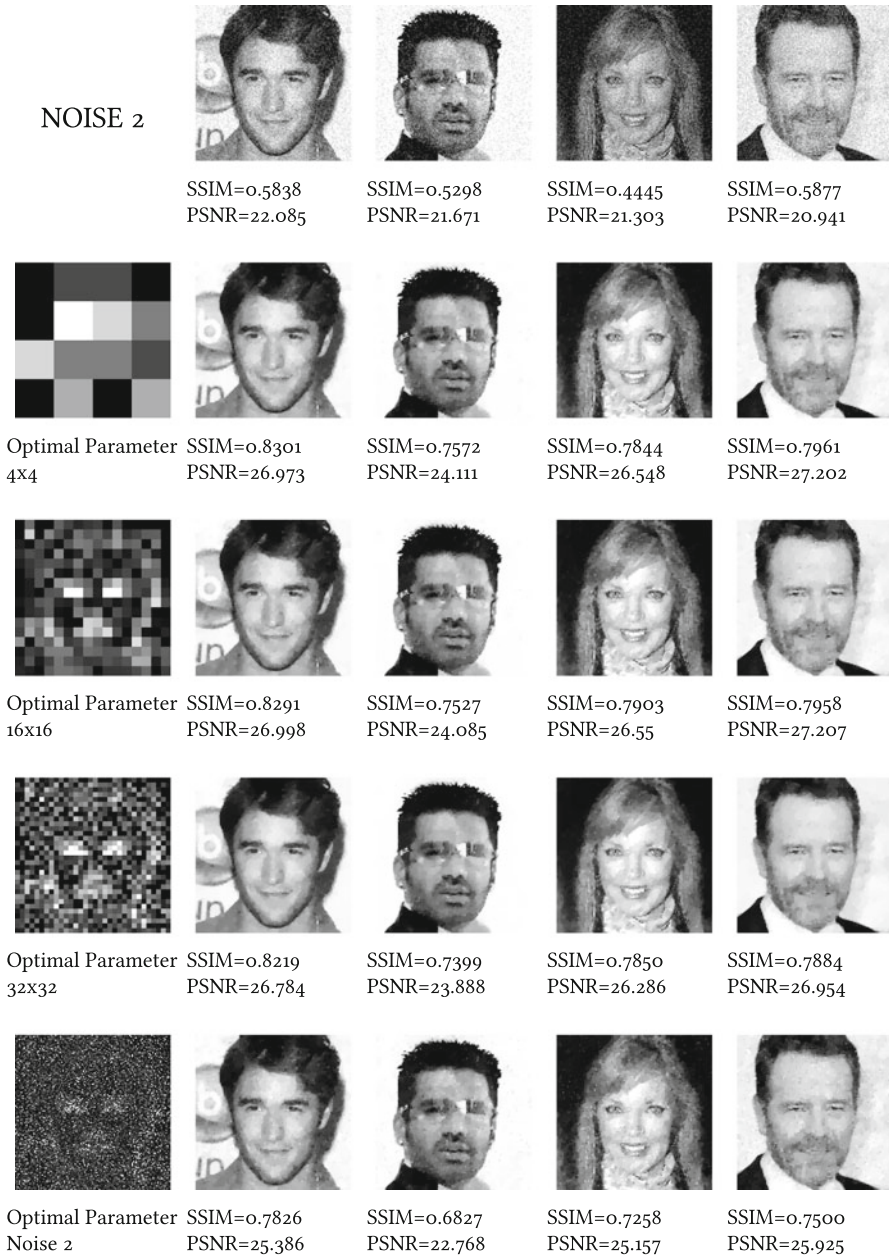


Fig. 4 Patch-dependent and scale-dependent regularization parameter Comparison of different learned patch-dependent and scale-dependent parameters used for denoising the validation dataset with noise 2

with respect to TV except in isolated instances. In the reconstruction of faces, as several gray scales are present within each structural component, TGV turns out to be a robust regularizer.

Moving further, we consider a scale-dependent parameter (18) and patch-dependent parameters (20), for the case of total variation denoising. The optimal learned parameters for the highest noise level in the training set are shown in Fig. 4. These learned parameters retrieve structural properties of the training dataset. In particular, as the number of degrees of freedom increases, a face structure can be identified in the weights.

It is of particular interest the behavior of these parameters in the validation dataset. Table 1 show the values of the optimal SSIM reconstruction (SD-TV) in both the training and validation datasets. Even though the more degrees of freedom for the regularization parameter allow for a better fitting in the training dataset, it performs poorly in the validation dataset according to the SSIM metric (see Fig. 5). This behavior is widely known in the machine learning community as *over-fitting*.

Table 1 SSIM quality measures Quality measures obtained in the training and validation dataset using the optimal parameter learned for different image denoising models

TRAINING								
num	noisy	TV	TGV	NL	SD-TV	PD-TV4	PD-TV16	PD-TV32
1	0.5838	0.8583	0.8715	0.7889	0.8441	0.8405	0.8433	0.8341
2	0.5298	0.8397	0.8463	0.7729	0.8226	0.8107	0.8121	0.8194
3	0.4447	0.8412	0.8612	0.8433	0.8713	0.8651	0.8655	0.8639
4	0.5877	0.8159	0.8270	0.8026	0.8531	0.8505	0.8544	0.8625
5	0.4865	0.7896	0.8234	0.8110	0.8398	0.8498	0.8457	0.8607
6	0.4699	0.8285	0.8469	0.7909	0.8343	0.8275	0.8283	0.8281
7	0.4827	0.8413	0.8564	0.7909	0.8218	0.7727	0.7785	0.8017
8	0.4884	0.8095	0.8325	0.7751	0.8389	0.8370	0.8381	0.8391
9	0.6144	0.8353	0.8654	0.7934	0.8505	0.8484	0.8484	0.8495
10	0.5029	0.8087	0.8366	0.7945	0.8298	0.7992	0.8087	0.8313
mean		0.8268	0.8467	0.7963	0.8407	0.8298	0.8323	0.8391
VALIDATION								
num	noisy	TV	TGV	NL	SD-TV	PD-TV4	PD-TV16	PD-TV32
1	0.6020	0.8232	0.8298	0.7847	0.7826	0.8301	0.8292	0.8219
2	0.5915	0.8557	0.8596	0.7094	0.6827	0.7572	0.7527	0.7399
3	0.5280	0.7480	0.7342	0.7707	0.7258	0.7844	0.7903	0.7850
4	0.5076	0.7816	0.7769	0.7221	0.7500	0.7961	0.7958	0.7884
5	0.4569	0.7944	0.7841	0.7728	0.7856	0.8254	0.8306	0.8284
6	0.5342	0.8215	0.8344	0.7258	0.7434	0.7847	0.7856	0.7783
7	0.4937	0.7865	0.7789	0.6591	0.7064	0.7628	0.7577	0.7375
8	0.5457	0.7453	0.7569	0.6903	0.7328	0.7780	0.7797	0.7708
9	0.4907	0.7567	0.7809	0.8092	0.7277	0.8036	0.7995	0.7855
10	0.5475	0.7937	0.8146	0.8359	0.8086	0.8586	0.8561	0.8452
mean		0.7907	0.7950	0.7480	0.7445	0.7981	0.7977	0.7881

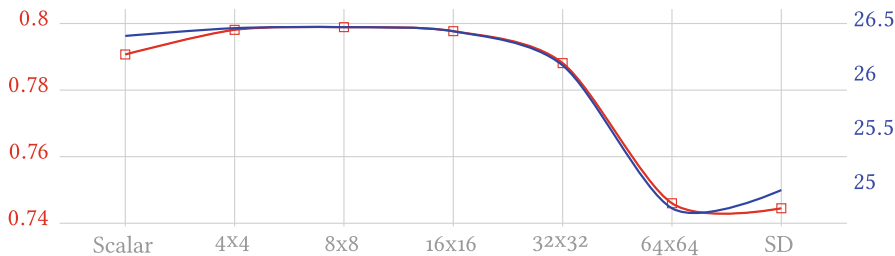


Fig. 5 Validation dataset reconstructions Average values of SSIM (red) and PSNR (blue) for the reconstruction of the validation dataset using different parameter models

To prevent the effect of *over-fitting* from happening and obtain better generalization properties, the patch-dependent regularization parameters (with few degrees of freedom) may be considered. To test this statement and realize how many degrees of freedom should serve that goal, we carry out an extra experiment. Specifically, the denoised results in the validation dataset for different dimensions are presented in Fig. 5. Indeed, the restriction on the degrees of freedom for the regularization parameter allows better generalization according to both the SSIM and the PSNR quality measures.

Conclusions

Bilevel optimization methods, in combination with energy models as lower-level problems, represent a state-of-the-art alternative for finding optimal quantities of interest in image processing tasks. Those quantities may be coefficients in the data fidelities, weights in the operators, or general functions involved in the different model terms. This methodology is particularly useful in combination with supervised learning techniques, where one can take advantage of the existence of training and validation sets.

These bilevel techniques have also the advantage that they can be mathematically analyzed and different results can be demonstrated, which allow an understanding of the structural characteristics of the problems under study. Issues such as the existence of optimal solutions and their regularity, and their characterization through first and second order optimality conditions, can be rigorously addressed. To achieve these objectives, however, the use of modern techniques of non-smooth optimization and variational analysis is important, in order to carry out a successful treatment of the non-convexities and non-differentiability of the problems.

On the basis of the studied properties, it is also possible to design efficient algorithms for solving the bilevel problems, as well as to design neural networks better adjusted to specific image processing tasks. These issues have already been addressed in the community, and represent a promising research direction for the future.

References

- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**(12), 124007 (2017)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Antil, H., Di, Z.W., Khatri, R.: Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography. *Inverse Probl.* **36**(6), 064001 (2020)
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Bartels, S., Weber, N.: Parameter learning and fractional differential operators: application in image regularization and decomposition. arXiv preprint arXiv:2001.03394 (2020)
- Beck, A.: *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia (2017)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
- Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: *IEEE CVPR*, vol. 2, pp. 60–65 (2005)
- Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399 (2012)
- Calatroni, L., De los Reyes, J.C., Schönlieb, C.-B.: Dynamic sampling schemes for optimal noise learning under multiple nonsmooth constraints. In: *IFIP Conference on System Modeling and Optimization*, pp. 85–95 (2013)
- Calatroni, L., Papafitsoros, K.: Analysis and automatic parameter selection of a variational model for mixed Gaussian and salt-and-pepper noise removal. *Inverse Probl.* **35**(11), 114001 (2019)
- Calatroni, L., Cao, C., De los Reyes, J.C., Schönlieb, C.-B., Valkonen, T.: *Bilevel Approaches for Learning of Variational Imaging Models*. Walter de Gruyter GmbH, pp. 252–290 (2017)
- Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numer. Math.* **76**(2), 167–188 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *JMIV* **40**, 120–145 (2011)
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
- Christof, C., De los Reyes, J.C., Meyer, C.: A nonsmooth trust-region method for locally Lipschitz functions with application to optimization problems constrained by variational inequalities. *SIAM J. Optim.* **30**(3), 2163–2196 (2020)
- D’Elia, M., De los Reyes, J.C., Miniguano, A.: Bilevel parameter optimization for nonlocal image denoising models. arXiv preprint arXiv:1912.02347 (2019)
- Dabov, K., et al.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Davoli, E., Fonseca, I., Liu, P.: Adaptive image processing: first order PDE constraint regularizers and a bilevel training scheme. arXiv preprint arXiv:1902.01122 (2019)
- Davoli, E., Liu, P.: One dimensional fractional order γ TV: gamma-convergence and bilevel training scheme. *Commun. Math. Sci.* **16**(1), 213–237 (2018)
- De los Reyes, J.C.: Optimal control of a class of variational inequalities of the second kind. *SIAM J. Control Optim.* **49**(4), 1629–1658 (2011)
- De los Reyes, J.C.: *Numerical PDE-Constrained Optimization*. Springer, Cham (2015)
- De los Reyes, J.C., Meyer, C.: Strong stationarity conditions for a class of optimization problems governed by variational inequalities of the second kind. *J. Optim. Theory Appl.* **168**(2), 375–409 (2016)
- De los Reyes, J.C., Schönlieb, C.-B., Valkonen, T.: The structure of optimal parameters for image restoration problems. *J. Math. Anal. Appl.* **434**(1), 464–500 (2016)

- De los Reyes, J.C., Schönlieb, C.-B., Valkonen, T.: Bilevel parameter learning for higher-order total variation regularisation models. *J. Math. Imaging Vision* **57**, 1–25 (2017)
- De los Reyes, J.C., Schönlieb, C.-B.: Image denoising: learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Probl. Imaging* **7**(4), 1183–1214 (2013)
- Dempe, S.: *Foundations of Bilevel Programming*. Springer Science & Business Media, Boston (2002)
- Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: *European Conference on Computer Vision*, pp. 184–199 (2014)
- Dontchev, A.L., Rockafellar, R.T.: *Implicit Functions and Solution Mappings*, vol. 543. Springer, New York (2009)
- Du, Q., et al.: Analysis and approximation of nonlocal diffusion problems with volume constraints. *SIAM Rev.* **54**(4), 667–696 (2012)
- Ehrhardt, M.J., Roberts, L.: Inexact Derivative-Free Optimization for Bilevel Learning. arXiv preprint arXiv:2006.12674 (2020)
- Ekeland, I., Temam, R.: *Convex Analysis and Variational Problems*. Society for Industrial and Applied Mathematics, Philadelphia (1999)
- Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2008)
- Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. *Multiscale Model. Simul.* **6**(2), 595–630 (2007)
- Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406 (2010)
- Gunzburger, M., Lehoucq, R.B.: A nonlocal vector calculus with application to nonlocal boundary value problems. *Multiscale Model. Simul.* **8**, 1581–1598 (2010)
- Haber, E., Horesh, L., Tenorio, L.: Numerical methods for experimental design of large-scale linear ill-posed inverse problems. *Inverse Probl.* **24**(5), 055012 (2008)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
- Hershey, J.R., Le Roux, J., Weninger, F.: Deep unfolding: model-based inspiration of novel deep architectures. arXiv preprint arXiv:1409.2574 (2014)
- Hintermüller, M., Papafitsoros, K.: *Generating Structured Nonsmooth Priors and Associated Primal-Dual Methods*, vol. 20, pp. 437–502. Elsevier (2019)
- Hintermüller, M., Rautenberg, C.N.: Optimal selection of the regularization function in a weighted total variation model. Part I: Modelling and theory. *J. Math. Imaging Vis.* **59**(3), 498–514 (2017)
- Hintermüller, M., Stadler, G.: An infeasible primal-dual algorithm for total bounded variation-based inf-convolution-type image restoration”. *SIAM J. Sci. Comput.* **28**(1), 1–23 (2006)
- Hintermüller, M., Wu, T.: Bilevel optimization for calibrating point spread functions in blind deconvolution. *Inverse Probl. Imaging* **9**(4), 1139–1170 (2015)
- Holler, G., Kunisch, K., Barnard, R.C.: A bilevel approach for parameter learning in inverse problems. *Inverse Probl.* **34**(11), 115012 (2018)
- Knoll, F., Bredies, K., Pock, T., Stollberger, R.: Second order total generalized variation (TGV) for MRI. *Magn. Reson. Med.* **65**(2), 480–491 (2011)
- Kobler, E., Effland, A., Kunisch, K., Pock, T.: Total Deep Variation for Linear Inverse Problems. arXiv preprint arXiv:2001.05005 (2020)
- Kobler, E., Klatzer, T., Hammernik, K., Pock, T.: Variational networks: connecting variational methods and deep learning. In: *German Conference on Pattern Recognition*, pp. 281–293 (2017)
- Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. *SIAM J. Imaging Sci.* **6**(2), 938–983 (2013)
- Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. *J. Math. Imaging Vis.* **27**(3), 257–263 (2007)
- Li, H., et al.: NETT: solving inverse problems with deep neural networks. *Inverse Probl.* **36**(6), 065005 (2020)
- Lou, Y., et al.: Image recovery via nonlocal operators. *J. Sci. Comput.* **42**(2), 185–197 (2010)

- Lunz, S., Öktem, O., Schönlieb, C.-B.: Adversarial regularizers in inverse problems. arXiv preprint arXiv:1805.11572 (2018)
- Mordukhovich, B.S.: Variational Analysis and Applications. Springer, Cham (2018)
- Nikolova, M.: A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**(1), 99–120 (2004)
- Nocedal, J., Wright, S.: Numerical Optimization. Springer Science & Business Media, New York (2006)
- Ochs, P., Ranftl, R., Brox, T., Pock, T.: Techniques for gradient-based bilevel optimization with non-smooth lower level problems. *J. Math. Imaging Vis.* **56**(2), 175–194 (2016)
- Outrata, J.V.: A generalized mathematical program with equilibrium constraints. *SIAM J. Control Optim.* **38**(5), 1623–1638 (2000)
- Outrata, J., Zowe, J.: A numerical approach to optimization problems with variational inequality constraints. *Math. Program.* **68**(1), 105–130 (1995)
- Ring, W.: Structural properties of solutions to total variation regularization problems. *ESAIM: Math. Model. Numer. Anal.* **34**(4), 799–810 (2000)
- Rockafellar, R.T., Wets, R.J.-B.: Variational Analysis, vol. 317. Springer Science & Business Media, Berlin (1998)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1–4), 259–268 (1992)
- Sawatzky, A., Brune, C., Müller, J., Burger, M.: Total Variation Processing of Images with Poisson Statistics, vol. 5702, pp. 533–540. Springer, Berlin/Heidelberg (2009)
- Schirotzek, W.: Nonsmooth Analysis. Springer Science & Business Media, Berlin/Heidelberg (2007)
- Sherry, F., et al.: Learning the sampling pattern for MRI. *IEEE Trans. Med. Imaging* **39**(12), 4310–4321 (2020)
- Sun, J., Li, H., Xu, Z.: Deep ADMM-Net for compressive sensing MRI. *Adv. Neural Inf. Process. Syst.* **29** (2016)
- Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
- Tappen, M.F.: Utilizing variational optimization to learn Markov random fields. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
- Tikhonov, A.N., Arsenin, V.: Solutions of Ill-Posed Problems, vol. 14. Winston, Washington, DC (1977)
- Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, p. 2 (1998)
- Valkonen, T.: A primal–dual hybrid gradient method for nonlinear operators with applications to MRI. *Inverse Probl.* **30**(5), 055012 (2014)
- Van Chung, C., De los Reyes, J.C., Schönlieb, C.-B.: Learning optimal spatially-dependent regularization parameters in total variation image denoising. *Inverse Probl.* **33**(7), 074005 (2017)
- Venkatakrishnan, S.V., C.A. Bouman, Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948 (2013)
- Villacís, D.: First order methods for high resolution image denoising. *Latin American Journal of Computing Faculty of Systems Engineering Escuela Politécnica Nacional Quito-Ecuador* **4**(3), 37–42 (2017)
- Wang, Y.-Q.: A multilayer neural network for image demosaicking. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 1852–1856 (2014)
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., others: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
- Xu, L., Ren, J.S.J., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: *Advances in Neural Information Processing Systems*, pp. 1790–1798 (2014)
- Yaroslavsky, L.P.: Digital picture processing: an introduction. *Appl. Opt.* **25**, 3127 (1986)



Multi-parameter Approaches in Image Processing

25

Markus Grasmair and Valeriya Naumova

Contents

Introduction	944
PDE-Based Approaches	946
Dictionary-Based Approaches	950
Parameter Selection	953
Multiparameter Discrepancy Principle	953
Balancing Principle and Balanced Discrepancy Principle	954
L -Hypersurface	955
Generalized Lasso Path	955
Parameter Learning	956
Numerical Solution	957
Numerical Examples	958
Conclusion	963
References	965

Abstract

Natural images often exhibit a highly complex structure that is difficult to describe using a single regularization term. Instead, many variational models for image restoration rely on different regularization terms in order to capture the different components of the image in question. While the resulting multipenalty approaches have in principle a greater potential for accurate image reconstructions than single-penalty models, their practical performance relies heavily on a

M. Grasmair (✉)
NTNU, Trondheim, Norway
e-mail: markus.grasmair@ntnu.no

V. Naumova (✉)
Machine Intelligence Department, Simula Consulting and SimulaMet, Oslo, Norway
e-mail: valeriya@simula.no

good choice of the regularization parameters. In this chapter, we provide a brief overview of existing multipenalty models for image restoration tasks. Moreover, we discuss different approaches to the problem of multiparameter selection. For the numerical examples, we will focus on the balanced discrepancy principle and the L-hypersurface method applied to PDE-based image denoising problems.

Keywords

Multiparameter regularization · Image restoration · Variational methods · Parameter selection · Discrepancy principle · L-hypersurface

Introduction

Image restoration aiming, for instance, at the recovery of lost information from noisy, blurred, and/or partially observed images plays an important role in many practical applications such as anomaly detection in medical images and galaxy analysis in astronomical images. With the massive production of digital images and videos, the need for efficient image restoration methods emerges even more. No matter how good cameras are, an improvement of the images is always desirable. Moreover, many image restoration tasks such as image denoising are necessary in many more applications than the ones mentioned above. Image denoising, being the simplest possible inverse problem, provides a useful and by now well-accepted framework in which different image processing ideas and techniques can be tested, compared, and perfected. Therefore, the field of image processing in general has received numerous contributions in the last decades from diverse scientific communities. Various statistical estimators, deep learning methods, adaptive filters, partial differential equations, transform-domain methods, splines, differential geometry-based methods, and regularization are only some of many areas and tools explored in studying image processing tasks.

This chapter does not intend to provide an overview of the vast amount of methods in image processing, but rather concentrate on variational multiparameter approaches for image restoration. These approaches have provided notable advances on different image restoration tasks in the last decades and continue to play an important role in this and other fields.

Mathematically speaking, we model an image restoration problem as follows:

$$y = Au + \delta, \tag{1}$$

where u is a ground truth image affected by the action of the imaging operator A and is measured in the presence of a random noise δ . For simplicity, we assume here that the noise is additive, although most of the argumentation and methods below still remain valid for more complicated scenarios. In the simplest case of denoising, the operator A is the identity; other typical examples are convolution operators in the case of deblurring and masking operators in inpainting tasks.

Classical variational approaches for the solution of (1) aim at solving an optimization problem of the form:

$$\hat{u} = \arg \min_u \{\ell(Au, y) + \lambda \Psi(u)\}, \quad (2)$$

where ℓ is a loss function that penalizes mismatch to the measurements, $\Psi(u)$ is a regularization term that penalizes mismatch to the image class of interest, and $\lambda > 0$ is a regularization parameter that balances the two terms. Such simple variational models cannot easily account for the highly complex and heterogeneous structure of natural images. As a potential remedy for this, an alternative approach based on the idea of imposing different penalization on the image u or its components u_k has been proposed. This leads to the model:

$$\hat{u} = \arg \min_u \{\ell(Au, y) + \sum_{i=1}^K \lambda_i \Psi_i(u)\}.$$

In the specific case when we are interested in separating different components of the image, such as cartoon and texture, we impose different penalization terms on the different components. This results in the model:

$$\hat{u} = \arg \min_{u=u_1+\dots+u_K} \{\ell(Au, y) + \lambda_1 \Psi_1(u_1) + \dots + \lambda_K \Psi_K(u_K)\}. \quad (3)$$

Again, ℓ is a loss function penalizing the mismatch to the measurements. Moreover, each regularization term Ψ_k with corresponding regularization parameter $\lambda_k > 0$ penalizes a different aspect of the combined image $u = u_1 + \dots + u_K$.

Based on the general formulations (2) or (3), one can differentiate at least two large classes of mathematical image restoration methods. On the one hand, there are PDE-based or, more general, variational methods for image restoration where the penalty terms use local (or in some recent approaches also nonlocal) potentially higher-order gradient information of the image. Typical approaches in that direction are variants of total variation regularization (cf. Rudin et al. 1992; Chambolle and Lions 1997) or, within a multi-penalty context, Mumford-Shah regularization (cf. Mumford and Shah 1989) or total generalized variation (cf. Bredies et al. 2010). A large overview of such methods can be, for instance, found in Scherzer et al. (2009) or Aubert and Kornprobst (2006).

On the other hand, there are approaches based on (generalized) wavelet decompositions or similar approaches based on computational harmonic analysis, which typically assume some type of sparsity of the images with respect to a suitable basis or dictionary. A classical example in that direction is the sub-quadratic wavelet-based penalization promoted in Daubechies et al. (2004). Multi-penalty approaches based on a collection of different dictionaries have been studied in Bobin et al. (2007); see also the references therein. These approaches allow to separate several morphologies in the image; typical examples of which are again texture and cartoon.

PDE-Based Approaches

The simplest PDE-based approaches use quadratic regularization terms, leading to linear, elliptic PDEs. The most basic example is the single-penalty model:

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|Au - y\|^2 + \frac{\lambda}{2} \int_{\Omega} |\nabla u|^2 \right),$$

where $\Omega \subset \mathbb{R}^2$ is the imaging domain. This leads to the Euler-Lagrange equation (or optimality condition):

$$A^*A\hat{u} - \lambda\Delta\hat{u} = A^*y$$

with homogeneous Neumann boundary conditions. Multi-penalty approaches replace the H^1 -norm in the regularization term by a composite of several terms of different orders. One of the simplest examples here uses in addition a squared norm of the Laplacian, leading to the model:

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|Au - y\|^2 + \frac{\lambda_1}{2} \int_{\Omega} |\nabla u|^2 + \frac{\lambda_2}{2} \int_{\Omega} (\Delta u)^2 \right), \quad (4)$$

or the corresponding Euler-Lagrange equation:

$$A^*A\hat{u} - \lambda_1\Delta\hat{u} + \lambda_2\Delta^2\hat{u} = A^*y.$$

Such models have been attractive for a long time mainly because of their computational simplicity: they only require the solution of a linear system, which moreover has in many cases a very simple structure. However, the usage of the squared H^1 -norm leads to very smooth, blurred results, a problem that may be made even worse by the addition of higher-order terms.

In Rudin et al. (1992), it has been argued that the “correct” way for treating image restoration problems is the usage of the total variation as the regularization term. There, one uses the L^1 -norm of the image gradient as penalization term, that is, $\Psi(u) = TV(u) = \int |\nabla u| dx$. In contrast to a quadratic penalization of the gradient, this has the advantage of a much weaker penalization of large gradients, allowing edges to remain in the restored image. While the total variation is well suited for capturing large uniform regions in images, and also edges, it destroys the other important feature of natural images: textured patterns. In order to be able to reconstruct realistic images, it is therefore necessary to find a way for incorporating textures into the regularization functionals.

One possibility, suggested by Meyer (2001) (see also Vese and Osher (2003), which contains the first numerical implementation of the method), is to decompose the image into a geometry part u_1 , which can be treated by the total variation, and a texture part u_2 , for the treatment of which he introduced a norm that is dual to total

variation. The resulting model has the form:

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|A(u_1 + u_2) - y\|^2 + \lambda_1 TV(u_1) + \lambda_2 \|u_2\|_G \right), \quad (5)$$

where the G -norm $\|\cdot\|_G$ is defined as follows:

$$\|v\|_G = \inf \{ \|v\|_\infty : v = \nabla \cdot \mathbf{v} \}.$$

Equivalently, this can be formulated as follows:

$$\hat{u} = \arg \min_{u, \mathbf{v}} \left(\frac{1}{2} \|A(u + \nabla \cdot \mathbf{v}) - y\|^2 + \lambda_1 TV(u) + \lambda_2 \|\mathbf{v}\|_\infty \right).$$

For a more precise definition of the involved spaces, see Meyer (2001). The intuition behind the introduction of the G -norm is the idea that textures mainly consist of rapidly oscillating, relatively uniform patterns. For such repeating structures, however, their G -norm is inversely proportional to the frequency of the oscillations: in the one-dimensional case, for instance, the G -norm of the function $u_2(x) = \sin(kx)$ would be $1/k$. More complex-related decomposition models, where an image is decomposed into more than two parts, have been suggested, for instance, in Bertalmio et al. (2003) and Aujol et al. (2005). An example of the resulting decomposition of an image into its cartoon part and texture part is given in Fig. 1. Note that only the positive part of the texture is shown and that it has been rescaled to fill the whole color range.

An alternative image decomposition approach can be derived from a model of image formation, which originates from the fact that natural images are projections of three-dimensional objects onto the two-dimensional image plane. Assuming that the depicted objects are up to a certain degree “homogeneous,” this gives rise to the model of images consisting of several distinct, smooth regions, bordered by the different objects’ silhouettes, which coincide with discontinuities in the image u . Based on this assumption, Mumford and Shah formulated their famous model



Fig. 1 Decomposition of an image into a cartoon and texture part according to Meyers model (5). *Left:* Original image. *Middle:* Resulting cartoon part. *Right:* Rescaled, positive texture part

(Mumford and Shah 1989):

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|Au - y\|^2 + \lambda_1 \int_{\Omega \setminus K} |\nabla u|^2 + \lambda_2 \text{len}(K) \right), \quad (6)$$

where K denotes the (one-dimensional) edge set in the image u and $\text{len}(K)$ its length. Originally, this model has been only formulated in the context of denoising, whereas its application to deblurring problems requires some additional constraints to be included (e.g., see Fornasier et al. 2013). Moreover, in contrast to the other models discussed in this chapter, it has the disadvantage of being highly non-convex. In addition, its numerical minimization requires in general some form of either approximation or parametrization of the edge set K . Different approaches to that end have been suggested using, e.g., phase-field approaches (see Ambrosio and Tortorelli 1990), nonlocal approximations (see Braides and Dal Maso 1997), singular perturbations (see Braides 1998), topological gradients (see Grasmair et al. 2013; Beretta et al. 2014), finite difference approximations (see Chambolle 1995; Gobbino 1998), or convex relaxations (see Pock et al. 2010). Note that this list is by no means exhaustive. In the numerical experiments in Section “Numerical Examples,” we have used the phase-field approach due to Ambrosio and Tortorelli (1990); see also Aubert and Kornprobst (2006, Chap. 4.2.4). Here, the edge set K is approximated by a *phase-field* v , which is a function on Ω that is approximately 0 in a thin strip surrounding K and approximately 1 outside this strip. This yields the model:

$$\min_{u,v} \left(\frac{1}{2} \|Au - y\|^2 + \lambda_1 \int_{\Omega} v^2 |\nabla u|^2 + \lambda_2 \int_{\Omega} \left(\epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (v - 1)^2 \right) \right) \quad (7)$$

for some small parameter $\epsilon > 0$, which roughly corresponds to the width of the strip that approximates K . As ϵ tends to zero, the solutions (u_ϵ, v_ϵ) of (7) converge to solutions of (6) in the sense that the functions u_ϵ converge to a solution \hat{u} of (6), whereas the phase fields v_ϵ converge to an indicator function of $\Omega \setminus K$. An example for this approximation of the Mumford-Shah model is presented in Fig. 2.

One of the drawbacks of total variation regularization is the so-called *stair-casing effect* that is often observed in the obtained results: in the reconstructed images,



Fig. 2 Application of the Mumford-Shah model to the parrots image. *Left:* Original image. *Middle:* Resulting cartoon part. *Right:* Resulting edge indicator according the Ambrosio-Tortorelli approximation (Ambrosio and Tortorelli 1990)

small edges are often inserted, and smooth changes of the intensities are broken up and replaced by gradual transitions. Convex approaches for improving this behavior are often formulated in terms of infimal convolutions of several convex functionals penalizing different smoothness properties of the component functions. The most basic approach in that direction is the infimal convolution of a total variation term and a quadratic penalization of the gradient, that is, the model:

$$\hat{u} = \arg \min_{u_1, u_2} \left(\frac{1}{2} \|A(u_1 + u_2) - y\|^2 + \lambda_1 \int_{\Omega} |\nabla u_1| + \lambda_2 \frac{1}{2} \int_{\Omega} |\nabla u_2|^2 \right).$$

Equivalently, this can be seen as a regularization method using the Huber-type functional:

$$\Psi_{\varepsilon}(u) = \int_{|\nabla u| \geq \varepsilon} |\nabla u| + \frac{1}{2\varepsilon} \int_{|\nabla u| < \varepsilon} |\nabla u|^2$$

with parameter $\varepsilon = \lambda_1/\lambda_2$. The quadratic term that becomes active at small gradients limits the stair-casing and allows for smooth, slow-intensity transitions, whereas the linear term penalizing large gradients allows for edges to remain in the reconstructed image.

Other common methods combine derivatives of several orders. The first idea in this direction can be found in Chambolle and Lions (1997), where the authors propose the model:

$$\hat{u} = \arg \min_{u_1, u_2} \left(\frac{1}{2} \|A(u_1 + u_2) - y\|^2 + \lambda_1 \int_{\Omega} |\nabla u_1| + \lambda_2 \int_{\Omega} |\nabla^2 u_2| \right). \tag{8}$$

This can be rewritten as follows:

$$\hat{u} = \arg \min_{u, v} \left(\frac{1}{2} \|Au - y\|^2 + \lambda_1 \int_{\Omega} |\nabla u - \nabla v| + \lambda_2 \int_{\Omega} |\nabla^2 v| \right). \tag{9}$$

Assuming that $\lambda_2 \gg \lambda_1$, we can interpret this model as a two-stage process, where we construct first a preliminary approximation ∇v of the gradient of the image which itself has a very low total variation and then construct the final approximation u in such a way that the total variation of the difference $u - v$ is small. As a consequence, the final result u will not be piecewise constant any more.

The paper Bredies et al. (2010) (see also Bredies and Holler 2014) introduced the concept of *total generalized variation* as a further generalization of total variation regularization with enhanced smoothing capabilities. In its second-order variant, it reduces to the model:

$$\hat{u} = \arg \min_{u, \mathbf{v}} \left(\frac{1}{2} \|Au - y\|^2 + \lambda_1 \int_{\Omega} |\nabla u - \mathbf{v}| + \lambda_2 \int_{\Omega} |\mathcal{E}\mathbf{v}| \right), \tag{10}$$



Fig. 3 Application of TGV regularization. *Left:* Resulting image \hat{u} . *Middle:* Norm of $D\hat{u}$. *Right:* Norm of \mathbf{v}

where:

$$\mathcal{E}\mathbf{v} = \frac{1}{2}(\nabla\mathbf{v} + (\nabla\mathbf{v})^T)$$

is the symmetrized gradient of the vector function \mathbf{v} . Here it is the gradient of u that is decomposed into two parts, the first being sparse, the second having a sparse symmetrized gradient. Compared to (9), the difference is that the second total variation has been replaced by the total deformation and the vector field \mathbf{v} that forms the first approximation of the gradient of u is no longer required to be irrotational. In the one-dimensional case, the two models (9) and (10) are equivalent. Figure 3 shows an example of the application of TGV regularization and the resulting decomposition of the gradient.

Dictionary-Based Approaches

PDE-based approaches were among the first ones that have considered the separation of an image into several distinct components. However, the actual solution of the resulting models can be very demanding numerically, in particular for non-quadratic models. In order to overcome this bottleneck, a complementary direction of work, inspired by ideas and advances from signal processing, is to consider image reconstruction and separation from the point of view of sparsity and compressed sensing.

Sparsity has become important prior for many image processing applications. Since natural images typically are not sparse in their pixel domain, different transforms such as wavelet transforms and different generalizations have been proposed in the last decades with the goal of finding better and more efficient image representations. In the sparse model, each datum (signal) can be approximated by the linear combination of a small (sparse) number of elementary signals, called atoms, from a prespecified basis or frame, called dictionary. The natural next

question is how to select or learn a dictionary Φ providing sparse representations for a given data class.

There are mainly two approaches: one is to employ some analytically defined dictionaries such as wavelets or an overcomplete discrete cosine transform, which are fast and easy to implement, but are suited only for a specific data type. The other approach is to learn a dictionary from the given training dataset for a specific task; see, for instance, Field and Olshausen (1996), Aharon et al. (2006), Gribonval and Schnass (2010), and Mairal et al. (2012). Even though the latter approach provides state-of-the-art results for many image processing tasks, it is very computationally and data demanding.

When one works with dictionary transforms, one can largely distinguish between synthesis-based approaches, which are purely formulated in terms of the dictionary coefficients, and analysis-based approaches, which essentially start from the resulting image. A single-penalty synthesis-based model has the form:

$$\min_{\alpha} \left(\frac{1}{2} \|A(\Phi\alpha) - y\|^2 + \lambda \Psi(\alpha) \right),$$

and the resulting image is given as follows:

$$u = \Phi\alpha.$$

Here α are the (sparse) coefficients and Φ is the dictionary.

A corresponding analysis-based approach would take the form:

$$\min_u \left(\frac{1}{2} \|Au - y\|^2 + \lambda \Psi(\Phi^\dagger u) \right),$$

with Φ^\dagger being the pseudo-inverse of the synthesis operator Φ . If one works with bases instead of frames, the matrix Φ is square and invertible, $\Phi^\dagger = \Phi^{-1}$, and the two approaches are equivalent. Moreover, in the case of orthonormal bases like the Fourier basis, we have $\Phi^{-1} = \Phi^T$.

We will first discuss two approaches that use a single analytic dictionary together with a multiparameter approach: the first approach, which is also the probably best known multiparameter approach, is the elastic net (Zou and Hastie 2005), which takes the form:

$$\min_{\alpha} \left(\frac{1}{2} \|A(\Phi\alpha) - y\|^2 + \lambda_1 \|\alpha\|_1 + \lambda_2 \frac{1}{2} \|\alpha\|_2^2 \right).$$

It has been widely used in statistics for robust regression and within imaging for various tasks like feature selection.

The second approach in this category uses the multi-scale nature of the wavelets and imposes different regularization parameters for different frequency bands of the wavelet regularization operator. This idea was pursued in Lu et al. (2007) for the

recovery of the high-resolution images. The regularization operator for the ill-posed problem is decomposed in a multiscale manner by using bi-orthogonal wavelets or tight frames. Specifically, the authors propose a multi-resolution framework which introduces different regularization parameters for different frequency bands of the regularization operator resulting in:

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|Au - y\|^2 + \frac{1}{2} \sum_{s=0}^p \lambda_s \|R_s^T u\|_2^2 \right),$$

where R_s and \bar{R}_s are obtained from a wavelet or frame system with:

$$\sum_{s=0}^p \bar{R}_s^T R_s = \mathbb{I}.$$

Here \mathbb{I} is the identity matrix. This model has the explicit solution:

$$(A^T A + \sum_{s=0}^p \lambda_s \bar{R}_s^T R_s) \hat{u} = A^T y.$$

The authors demonstrate in extensive numerical examples the superiority of the method for high-resolution image recovery from a set of shifted low-resolution images compared to single-penalty methods such as H^1 and total variation. Moreover, the proposed multiparameter approach requires less computational resources than the single-penalty total variation method.

Another approach more common in harmonic analysis and signal processing literature is based on imposing different penalization for different image components for cartoon-texture separation task. These ideas were pursued by Daubechies and Teschke (2005), who proposed a multi-penalty formulation with an ℓ_1 - and a weighted ℓ_2 -norm as a numerically efficient substitute for the variational problem in Vese and Osher (2003). In particular, penalization in the wavelet shrinkage substitutes the total variation constraint considered in Vese and Osher (2003) and allows to recover cartoon, whereas the weighted ℓ_2 -norm is used for texture recovery in the Fourier domain.

In signal processing, a wider class of potential transforms for recovery of different morphological components has been considered. In particular, morphological component analysis (MCA) (Starck et al. 2004, 2005; Bobin et al. 2007) has been devised to solve the problem of recovering the different components from their combination. MCA assumes that a dictionary of bases $\Phi_1 = [\phi_{11}, \dots, \phi_{1K}]$ and $\Phi_2 = [\phi_{21}, \dots, \phi_{2K}]$ exists such that u_1 is sparse in Φ_1 and not in Φ_2 and vice versa. In Starck et al. (2004, 2005) it was proposed to estimate the components u_1 and u_2 by solving the constrained optimization problem:

$$\min_{u_1, u_2} \|\Phi_1^\dagger u_1\|_1 + \|\Phi_2^\dagger u_2\|_1 \text{ s.t. } \|y - A(u_1 + u_2)\|_2^2 \leq \sigma. \tag{11}$$



Fig. 4 Decomposition of an image into piecewise smooth and texture parts according to MCA. The addition of the texture part and the piecewise smooth part reproduces the original image. *Left:* Original image. *Middle:* Piecewise smooth content part. *Right:* Texture part

The parameter σ should take into account both the noise level and the model inaccuracies in representing sparsely u_1 and u_2 . Figure 4 illustrates the performance of the MCA method for image separation with two analytic dictionaries: curvelets for the cartoon part and the discrete cosine transform for the texture part.

The benefit of such a separation is obvious, as there is an agreement that images are in fact a mixture of cartoon and texture parts. By treating each of the parts separately using a proper dictionary, the image is processed much better and still efficiently by using analytic-based dictionaries. Moreover, MCA can be run either on the complete image or on small and overlapping patches. The immediate benefits of the latter mode are the locality of the processing, allowing for efficient parallel implementation, and the ability to incorporate learned dictionaries into the MCA.

Parameter Selection

Despite the promising performance of multiparameter models on various tasks, they lead to additional challenges related to the need for multiple parameter selection. This topic has been largely underexplored with some scarce efforts from different communities. Below we provide an overview of the existing approaches on the parameter selection, most of which were extended from a single-parameter to a multiparameter setting, whereas others are more data-driven and focus on learning the parameters from a given training dataset.

Multiparameter Discrepancy Principle

The authors of Lu and Pereverzev (2011) consider the multiparameter functional:

$$\hat{u} = \arg \min_u \|Au - y\|^2 + \sum_{i=1}^K \lambda_i \Psi_i(u) + \beta \|u\|^2, \quad (12)$$

where $\{\lambda_i\}_{i=1}^K > 0$ and $\beta > 0$ are the regularization parameters, $\Psi_i(u) = \|R_i u\|^2$, and R_i is a penalizing operator. They proposed an a posteriori strategy based on the extension of the classical discrepancy principle for choosing the parameter set $(\{\lambda_i\}_{i=1}^K, \beta)$. Specifically, the parameters are chosen as to satisfy the equation:

$$\|Au(\{\lambda_i\}_{i=1}^K, \beta) - y\| = c\delta,$$

where δ is the (assumed) noise level and $c \geq 1$ is some a priori specified parameter. Typically, c is chosen slightly larger than 1, e.g., $c = 1.2$, in order to obtain a stable solution in case of a slight underestimation of the noise level.

The authors also propose a numerical realization of this principle based on the model function approximation, which approximates the discrepancy term locally by means of some simple model function $m(\{\lambda_i\}_{i=1}^K, \beta)$ and allows to find subsequent parameters based on some simple equations. The scheme results in a nonunique parameter selection rule, which limits its applicability in practice. To overcome this issue, the follow-up work Lu et al. (2010) introduced a quasi-optimality criterion to facilitate a unique choice.

Balancing Principle and Balanced Discrepancy Principle

The nonuniqueness of the discrepancy principle was also addressed in Ito et al. (2014), where the authors consider augmented Tikhonov regularization and revisit the balancing principle for two parameter regularization. As a result, the balanced discrepancy principle was suggested, which incorporates the constraints into the augmented approach and allows to partially resolve the nonuniqueness issue.

As a first step, we consider the following balancing principle, where we choose the parameters λ_i in such a way that the system

$$\begin{cases} \hat{u} = \arg \min_u \|Au - y\|^2 + \lambda_1 \Psi_1(u) + \lambda_2 \Psi_2(u) \\ \lambda_i = \frac{1}{\gamma} \frac{\|y - A\hat{u}\|}{\Psi_i(\hat{u})}, \quad i = 1, 2, \end{cases}$$

is satisfied. That is, we are interested in finding parameters λ_i which balance the data fidelity with the respective penalty term. Here $\gamma > 0$ is a weighting parameter.

The balanced discrepancy principle combines this idea with the discrepancy principle. That is, we choose the weighting parameter γ in such a way that the residual satisfies $\|Au - y\| = c\delta$. For two-parameter regularization, this leads to the system:

$$\begin{cases} \|Au - y\| = c\delta, \quad c \geq 1, \\ \lambda_1 \Psi_1(u) = \lambda_2 \Psi_2(u). \end{cases} \tag{13}$$

Again, $c \geq 1$ is a safety parameter tasked to ensure stability of the method and is usually chosen slightly larger than 1. Efficient algorithms based on Broyden's method and fixed point methods have been suggested in Ito et al. (2014) for the numerical solution of (13).

***L*-Hypersurface**

In Belge et al. (1998, 2002), a parameter selection rule for functional (12) with $\beta = 0$ has been proposed, which is based on the generalization of the *L*-curve method (Hansen 1992) to the multiparameter setting. Similar to the one-dimensional case, one plots on the appropriate scale the residual norm

$$z(\lambda) = \|y - A\hat{u}(\lambda)\|^2$$

against the constraint norms

$$u_j(\lambda) = \Psi_j(\hat{u}), \quad j = 1, \dots, K.$$

Here $\lambda = [\lambda_1, \dots, \lambda_K]^T$. Given an appropriate scaling function ϕ , e.g., $\phi(u) = \log(u)$, the *L*-hypersurface is defined as the graph of the map $\beta(\lambda) : \mathbb{R}_+^K \rightarrow \mathbb{R}^{K+1}$:

$$\beta(\lambda) = [\phi(u_1(\lambda)), \dots, \phi(u_K(\lambda)), \phi(z(\lambda))].$$

A point on the *L*-hypersurface around which the surface is maximally warped corresponds to a point where the regularization and data-fitting errors are approximately balanced. The surface warpedness can be measured by calculating the Gaussian curvature. However, since evaluation of the Gaussian curvature for a large number of regularization parameters can be a computationally expensive task, which also might yield multiple extrema, the authors propose a surrogate minimum distance function (MDF) to approximate the curvature. However, the accuracy of the *L*-hypersurface approximation with MDF sometimes depends on the MDF origin. The authors provide some heuristic rule for the origin selection, which seems to work in specific cases. However, a robust means for selecting the origin is needed to promote practical usability of the method.

Generalized Lasso Path

In Grasmair et al. (2018) a fully adaptive approach for parameter selection was proposed for a multi-penalty functional of the form:

$$\min_{u,v} \frac{1}{2} \|A(u+v) - y\|^2 + \lambda_1 \|u\|_1 + \lambda_2 \|v\|_2^2,$$

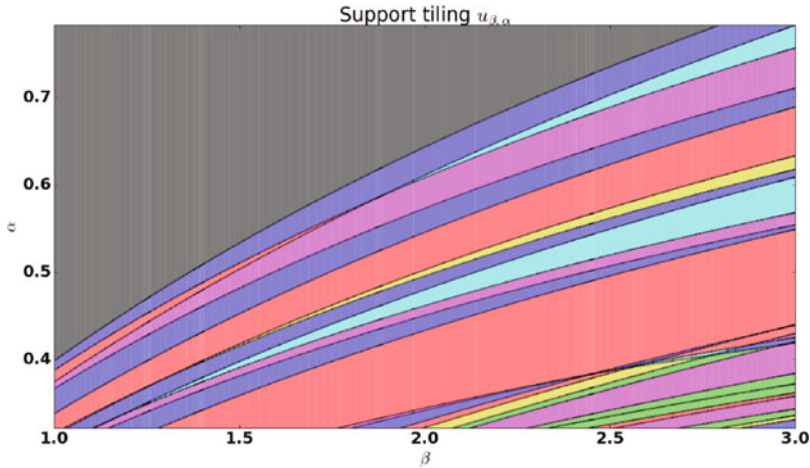


Fig. 5 Part of the parameter space detailing the different solutions. Each of the different tiles corresponds to a different support or sign pattern of the solution of interest

where u is a sparse signal of interest and v is some pre-measurement noise. The authors first extended the single-penalty lasso path algorithm (Efron et al. 2004) to a multiparameter lasso path algorithm, which partitions the parameter space $(\lambda_1, \lambda_2) \in \mathbb{R}_+^2$ into disjoint connected tiles such that the solution $\hat{u}(\lambda_1, \lambda_2)$ has a constant support and sign pattern on each tile; see Fig. 5. Such partitioning also allows to get additional insights into the problem at hand and understand the sensitivity of the solution with respect to the parameters. Once the tiles are constructed, one can employ a data-driven rule that adaptively selects a tile and corresponding parameters (λ_1, λ_2) , based on maximizing the signal-to-noise ratio of the solution. The authors provide an efficient algorithm for tile construction. Moreover, the superiority of the algorithm with respect to the state-of-the-art methods such as orthogonal matching pursuit, iterative hard thresholding, and the lasso is demonstrated in extensive numerical results. The approach can be extended to other non-homogenous norms as soon as the sparsity of the solution is relatively low.

Parameter Learning

An alternative approach to the methods above is based on a learning framework, where one uses a training set $\{u_i, y_i\}_{i=1}^N$ of independent pairs of clean and noisy images to select the optimal parameter by minimizing a certain objective functional. A prominent example in regularization theory are bi-level optimization methods (Kunisch and Pock 2013; De los Reyes and Schönlieb 2013), where the lower-level problem is defined through a parametrized variational model such as (9), (10), or (12), and the upper-level problem measures the error of the lower-level solution with respect to the ground truth, i.e., $\|\hat{u} - u\|_2^2$. Parameters are then learnt by minimizing

the mean error of the lower-level problem on the given training set. In many cases, the solution of the lower-level problem presupposes the PDE-based optimization and can be very computationally demanding.

In many real-life applications, the access to the ground truth image cannot be granted or might indeed be impossible, such as in X-ray tomography. Therefore, the recent efforts were dedicated to the development of an unsupervised parameter learning rule for different regularization methods (de Vito et al. 2018). The attractiveness of the suggested approach lies in the fact that one requires only a training set of noisy samples $\{y_i\}_{i=1}^N$ for learning the optimal parameter for given class of images or data in general. The idea behind the method is that the ground truth images follow intrinsically a lower dimensional geometry (i.e., they belong to a lower dimensional manifold), which can be approximated by using a training set of noisy samples. Once the proxy \tilde{u} is calculated, one can use it to guide the selection of the parameter by minimizing, for instance, the discrepancy $\|\hat{u} - \tilde{u}\|^2$. The first step of finding a suitable proxy \tilde{u} is completely independent of the regularization method and explores only the structure of the solution, whereas the second step of selecting the optimal parameter is dependent on the regularization method. The authors also showed that a learned parameter can be used on new images with similar structure without any retraining.

Numerical Solution

With the exception of the Mumford-Shah model, all approaches discussed above require the minimization of a convex functional composed of three or more subparts. However, many of the models include some non-smooth, sparsity-promoting terms, which make the application of non-smooth optimization algorithms necessary. In the recent years, convex analysis-based methods have been the method of choice in many imaging applications, particularly methods based on the augmented Lagrangian or alternating direction method of multipliers (ADMM) and various splitting methods. A large overview of different algorithms can be found in Bauschke and Combettes (2011), Combettes and Pesquet (2011), and Komodakis and Pesquet (2015). A specific mention here is deserved by the Chambolle-Pock algorithm (Chambolle and Pock 2011), which has been demonstrated to be very efficient in several total variation-based applications. We refer the reader not familiar with convex analysis to Komodakis and Pesquet (2015), which contains a succinct introduction into the main concepts and results necessary for the implementation of the different algorithms.

There are at least two notable differences between single-penalty and multi-penalty methods when it comes to their practical implementation: first, by their very nature, they require the minimization of functionals consisting of three or more separate terms. However, many of the more efficient primal-dual methods are primarily formulated only for a sum of two functionals, that is, a loss term and a single regularization term. In the situation of pure decomposition-based models like (8) or (11), which take the form:

$$\min_{u_1, \dots, u_K} \left(\frac{1}{2} \|A(u_1 + \dots + u_K) - y\|^2 + \lambda_1 \Psi(u_1) + \dots + \lambda_K \Psi(u_K) \right),$$

it is nevertheless possible to apply the standard approaches by splitting the whole model into the two subparts:

$$\begin{aligned} F_1(u_1, \dots, u_K) &= \frac{1}{2} \|A(u_1 + \dots + u_K) - y\|^2, \\ F_2(u_1, \dots, u_K) &= \lambda_1 \Psi_1(u_1) + \dots + \lambda_K \Psi_K(u_K). \end{aligned} \tag{14}$$

In this case, the *prox-operator* (see Komodakis and Pesquet 2015) for F_2^* , which is the central ingredient in all the aforementioned algorithms, decouples into the sum of the prox-operators for the regularization functionals $\Psi_1^*, \dots, \Psi_K^*$. As long as the latter ones can be efficiently evaluated, an efficient implementation of these algorithms is possible.

In situations where this split is not possible, the direct application of many well-known splitting methods can be numerically more challenging. However, there exist generalizations to the sum of three or more convex functionals. For instance, examples of how ADMM and Douglas-Rachford splitting can be adapted to this more general setting can be found in Combettes and Pesquet (2011, Chap 10.7). In addition, there exists a growing number of algorithms specifically aimed at the minimization of a sum of three convex functionals. One notable example here is due to Condat (2013) and Vũ (2013). We refer again to Komodakis and Pesquet (2015), where a large number of similar algorithms are collected.

The second difference to single-parameter settings is the numerical realization of the parameter choice: heuristic rules for single-parameter regularization like balancing principle or the L -curve require the minimization of the regularization functional for a number of different regularization parameters in order to find the optimal choice. However, the situation becomes notably more complicated for multi-penalty methods, as one has to find optimal parameters within an at least two-dimensional set. Methods like L -hypersurfaces therefore require many more solutions to yield reasonable results than in the single-penalty case. In order to speed up computations, it is therefore necessary to implement good stopping criteria for the optimization algorithms that not only take into account the convergence of the algorithm but also the question whether the current parameter setting may be feasible or not; in the latter case, an early termination of the optimization algorithm can lead to a significant gain in efficiency.

Numerical Examples

In the following, we will demonstrate the behavior of several parameter choice rules for different approaches to image denoising. We restrict ourselves in this section to PDE-based models. The main reason is that dictionary-based approaches

that provide state-of-the-art results are based on learned dictionaries rather than predefined ones (Starck et al. 2015). Learning a dictionary is a problem by itself, requiring a proper tuning of many parameters that influence the performance of the algorithm. The bi-level approaches presuppose the existence of training set for finding a parameter that minimizes the error between the ground truth and the reconstructed image. This type of setting falls within machine learning framework and is outside the scope of the current chapter.

We compare the performance of the balanced discrepancy principle, the L -hypersurface method, and the discrepancy principle without additional balancing. We chose to omit a comparison with the generalized lasso and with machine learning approaches, as the former is a method that is applicable only in very specialized settings and the latter require a large amount of training data of sufficiently good quality.

As a test example, we have used the “baboon” image, as it contains sharp edges and a high contrast between different image regions as well as parts characterized by a marked texture. For the denoising, we have added to each of the three color channels pixel-wise i.i.d. Gaussian noise with a standard deviation $\sigma = 50$, the true image taking values in the range $[0, 255]$. See Fig. 6 for the true and the noisy image.

We consider first the H^1 -Laplacian model (4) applied to denoising, that is, the model:

$$\hat{u} = \arg \min_u \left(\frac{1}{2} \|u - y\|^2 + \frac{\lambda_1}{2} \int_{\Omega} |\nabla u|^2 + \frac{\lambda_2}{2} \int_{\Omega} (\Delta u)^2 \right), \quad (15)$$

for some given noisy image y . Here all terms are applied separately, but with the same regularization parameters λ_1 and λ_2 , to the three color channels of the image. As discussed above, this is a quadratic optimization problem with the Euler-Lagrange equation (optimality condition):

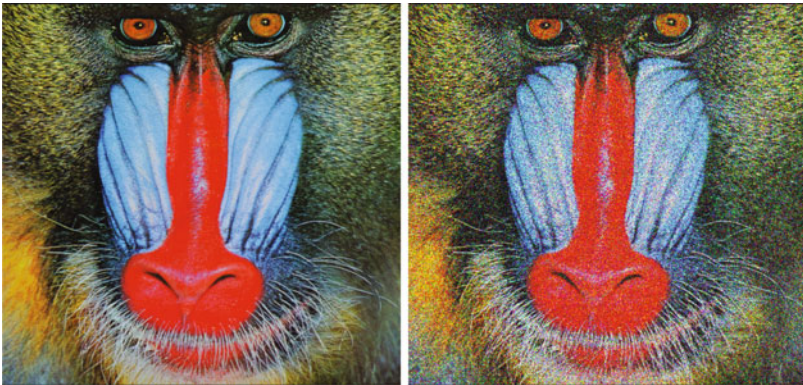


Fig. 6 Test image used for the numerical examples. *Left*: Original, noise-free image. *Right*: Noisy image

$$u - \lambda_1 \Delta u + \lambda_2 \Delta^2 u = y,$$

again applied separately to the different color channels.

Figures 7 and 8 show the results obtained by this approach together with an analysis of the parameter settings. Figure 7 shows the resulting L -hypersurface, the parameters for which the residual is smaller than the noise level, and results for the balanced discrepancy principle (13); here we have chosen $c = 1$, as the precise

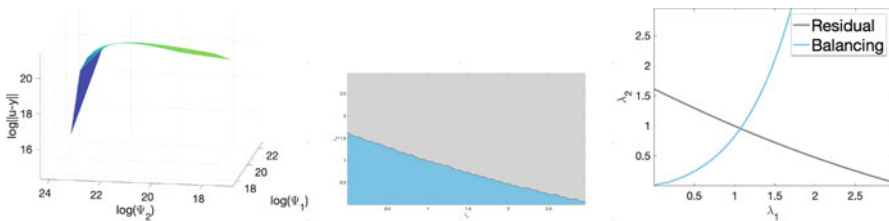


Fig. 7 Analysis of the parameter settings for the H^1 -Laplace denoising model (15) applied to the noisy baboon image. *Left*: Resulting L -hypersurface. *Middle*: Admissible (blue) versus inadmissible (gray) parameter settings according to the discrepancy principle. *Right*: The gray curve depicts the parameters that satisfy the discrepancy principle with equality, the blue curve the parameters that satisfy the balancing principle. The parameter setting chosen according to the balanced discrepancy principle is the intersection of the two curves

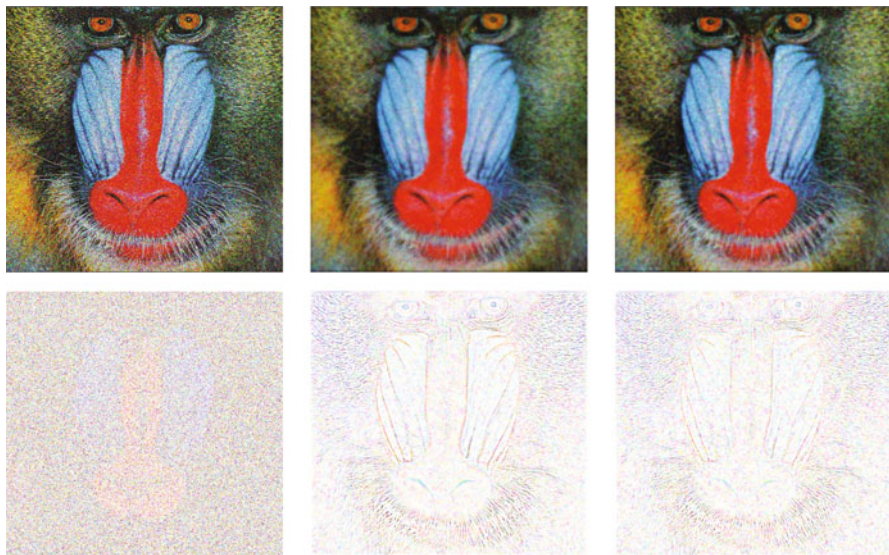


Fig. 8 Results of the H^1 -Laplace denoising model for different parameter choices. *First row*: Resulting denoised image. *Second row*: Error, that is, difference between reconstruction and true, noise-free image. *Left*: Optimal reconstruction according to MSE, obtained by full grid search; PSNR = 15.18. *Middle*: Optimal reconstruction subject to discrepancy principle; PSNR = 20.59. *Right*: Result with balanced discrepancy principle; PSNR = 20.68

noise level was available. The latter yields a unique parameter pair, which has been used to obtain the right hand images in Fig. 8. In addition, we have performed a full grid search in order to find the parameter pair that minimizes the mean square error (MSE) as well as the pair minimizing the MSE subject to the constraint that the discrepancy principle is satisfied with equality. The resulting images as well as the PSNR for the different results are shown in Fig. 8. Note that the latter uses the knowledge of the actual noise-free image, which of course is not available in practice. Moreover, it is necessary to mention that both the MSE and the PSNR are somehow dubious quality measures for images, as they ignore all structural information that is present in the images and only consider point-wise discrepancies.

Next, we perform a similar numerical study for the Ambrosio-Tortorelli approximation (7) of the Mumford-Shah model (6), that is, the model:

$$\min_{u,v} \left(\frac{1}{2} \|u - y\|^2 + \lambda_1 \int_{\Omega} v^2 |\nabla u|^2 + \lambda_2 \int_{\Omega} \left(\epsilon |\nabla v|^2 + \frac{1}{4\epsilon} (v - 1)^2 \right) \right).$$

This functional is non-convex because of the interaction between v and u in the second term, and thus the convex optimization methods discussed in the previous section are not readily applicable. Instead, we apply an alternating minimization procedure, where we alternate between minimizing with respect to u for fixed v and with respect to v for fixed u . This results in the iteration:

$$\begin{aligned} u_{k+1} &\leftarrow \text{solution of } u - 2\lambda_1 \nabla \cdot (v_k^2 \nabla u) = y, \\ v_{k+1} &\leftarrow \text{solution of } (1 + 4\lambda_1 \epsilon |\nabla u_{k+1}|^2) v - 4\lambda_2 \epsilon^2 \Delta v = 1, \end{aligned}$$

where both PDEs are solved with homogeneous Neumann boundary conditions. The parameter ϵ was chosen to be 1 pixel-width; this results in an edge-indicator function that is highly localized around the detected edges.

The results for this approximation of the Mumford-Shah model are shown in Figs. 9 and 10. Again, we have compared the result for the balanced discrepancy principle with the optimal results according to MSE obtained by a full grid search. As can be expected from the Mumford-Shah model, which completely disregards texture, the results are more cartoon-like than with even the H^1 -Laplace model, leading to a slightly lower PSNR. At the same time, the result includes distinct edges, which have been blurred in the other model.

We can also observe for the Mumford-Shah model that the balancing of the two regularization terms is crucial even in the presence of the discrepancy principle. This can be seen clearly in Fig. 11, where we have compared the results according to the balanced discrepancy principle with a result that satisfies the discrepancy principle, but where the second regularization parameter has been chosen to small. One can clearly see that this results in a general under-smoothing of the image.

As final example, we consider the Chambolle-Lions model (8) applied to the noisy parrots image, that is, the model:

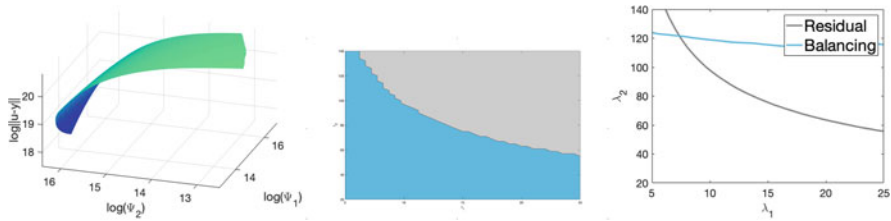


Fig. 9 Analysis of the parameter settings for the Mumford-Shah denoising model applied to the noisy baboon image. *Left:* Resulting L -hypersurface. *Middle:* Admissible (blue) versus inadmissible (gray) parameter settings according to the discrepancy principle. *Right:* The gray curve depicts the parameters that satisfy the discrepancy principle with equality, the blue curve the parameters that satisfy the balancing principle. The parameter setting chosen according to the balanced discrepancy principle is the intersection of the two curves

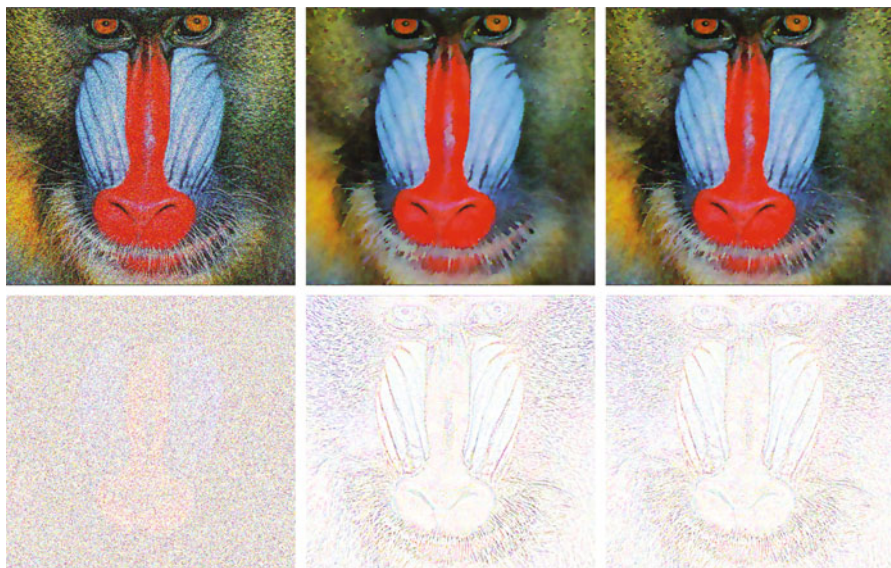


Fig. 10 Results of the Mumford-Shah denoising model for different parameter choices. *First row:* Resulting denoised image. *Second row:* Error, that is, difference between reconstruction and true, noise-free image. *Left:* Optimal reconstruction according to MSE, obtained by full grid search; PSNR = 15.28. *Middle:* Optimal reconstruction subject to discrepancy principle; PSNR = 19.93. *Right:* Result with balanced discrepancy principle; PSNR = 20.29

$$(\hat{u}_1, \hat{u}_2) = \arg \min_{u_1, u_2} \left(\frac{1}{2} \|u_1 + u_2 - y\|^2 + \lambda_1 \int_{\Omega} |\nabla u_1| + \lambda_2 \int_{\Omega} |\nabla^2 u_2| \right). \quad (16)$$

In this case, the result is decomposition of the restored image \hat{u} into a part \hat{u}_1 mostly containing the cartoon-like components of \hat{u} and a part \hat{u}_2 mostly containing the texture-like components. Moreover, we have a convex but non-smooth optimization

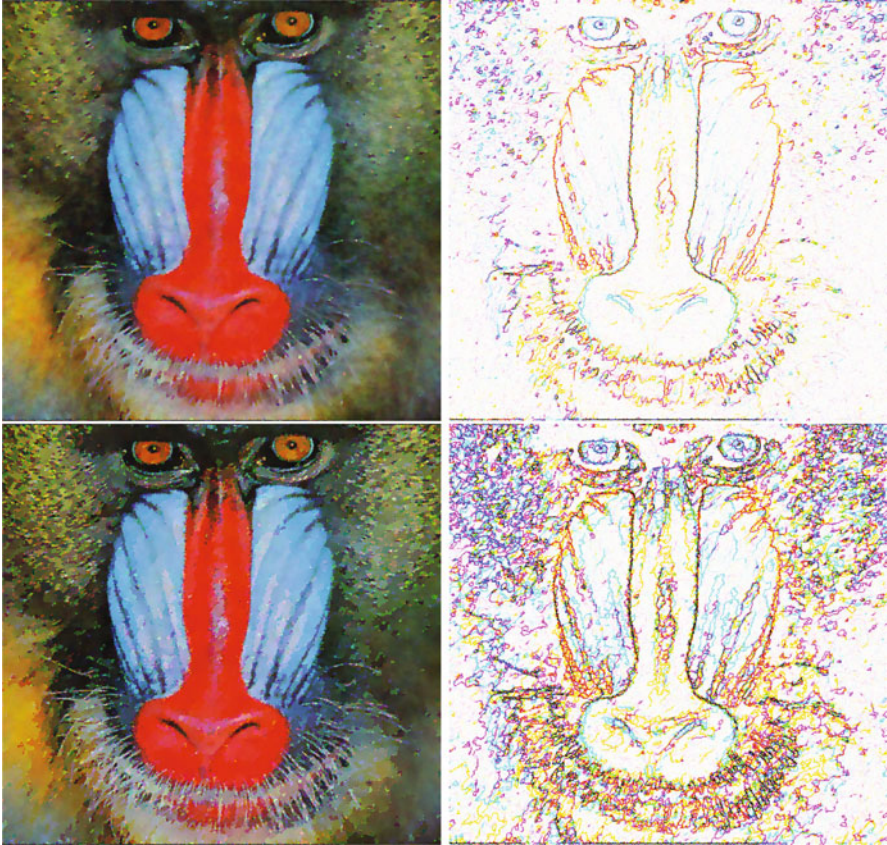


Fig. 11 Application of Mumford-Shah denoising to the noisy baboon image Fig. 6. *Upper row:* Denoised image and edge indicator using the balanced discrepancy principle. *Lower row:* Denoised image and edge indicator satisfying the discrepancy principle, but not the additional balancing principle

problem, which can be solved by any of the methods described in Section “[Numerical Solution](#)”. Specifically, we have used the Chambolle-Pock algorithm (Chambolle and Pock 2011) using the splitting (14). We note here that the solution of (16) is not unique, as neither regularization term penalizes constant functions. In order to obtain a unique solution, we have therefore added the restriction $\int_{\Omega} \hat{u}_1 dx = 0$. The results for this model are shown in Figs. 12 and 13.

Conclusion

Multiparameter regularization is a theoretically sound and efficient framework for various image processing applications ranging from the basic task of image denoising to inpainting and deblurring. Both PDE-based and data-driven approaches

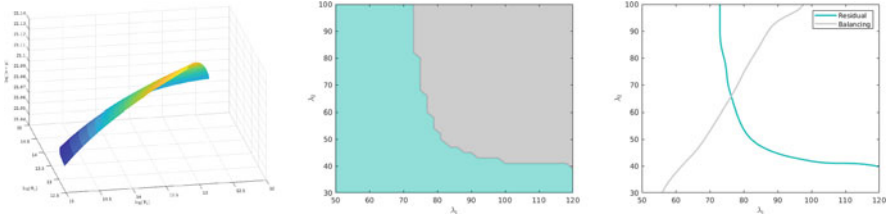


Fig. 12 Analysis of the parameter settings for the Chambolle-Lions model applied to the noisy parrots image. *Left:* Resulting L -hypersurface. *Middle:* Admissible (blue) versus inadmissible (gray) parameter settings according to the discrepancy principle. *Right:* The blue curve depicts the parameters that satisfy the discrepancy principle with equality, the gray curve the parameters that satisfy the balancing principle. The parameter setting chosen according to the balanced discrepancy principle is the intersection of the two curves



Fig. 13 Application of the Chambolle-Lions model to the denoising of the parrots image. *Upper row, left:* Noisy data. *Upper row, right:* Total result $\hat{u}_1 + \hat{u}_2$ using the balanced discrepancy principle. *Lower row, left:* Cartoon part \hat{u}_1 of the solution. *Lower row, right:* Texture part \hat{u}_2 of the solution

have been utilizing multiparameter regularization to obtain a good reconstruction quality and reduce the number of degrees of freedom. In this chapter, we provided an overview of the state of the art for multiparameter methods for image processing applications, also discussing aspects related to parameter selection and numerical realization. For clarification, we have also illustrated the performance of certain methods on simple denoising and decomposition examples.

There are several interesting open questions related to both numerical and theoretical aspects of multiparameter regularization. Specifically, further systematic studies of parameter learning from noisy data (unsupervised learning) not only could be beneficial for the specific methods but also could provide new insights into efficient construction of unsupervised deep learning algorithms.

References

- Aharon, M., Elad, M., Bruckstein, A.M.: On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *J. Linear Algebra Appl.* **416**, 48–67 (2006)
- Ambrosio, L., Tortorelli, V.M.: Approximation of functionals depending on jumps by elliptic functionals via Γ -convergence. *Commun. Pure Appl. Math.* **43**(8), 999–1036 (1990)
- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing. Applied Mathematical Sciences*, vol. 147, 2nd edn. Springer, New York. Partial differential equations and the calculus of variations, With a foreword by Olivier Faugeras (2006)
- Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vis.* **22**(1), 71–88 (2005)
- Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York (2011)
- Belge, M., Kilmer, M.E., Miller, E.L.: Simultaneous multiple regularization parameter selection by means of the l-hypersurface with applications to linear inverse problems posed in the wavelet transform domain. In: *Bayesian Inference for Inverse Problems*, vol. 3459, pp. 328–336. International Society for Optics and Photonics (1998)
- Belge, M., Kilmer, M.E., Miller, E.L.: Efficient determination of multiple regularization parameters in a generalized l-curve framework. *Inverse Prob.* **18**(4), 1161 (2002)
- Beretta, E., Grasmair, M., Muszkieta, M., Scherzer, O.: A variational algorithm for the detection of line segments. *Inverse Prob. Imaging* **8**(2), 389–408 (2014)
- Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.* **12**(8), 882–889 (2003)
- Bobin, J., Starck, J.-L., Fadili, J.M., Moudden, Y., Donoho, D.L.: Morphological component analysis: an adaptive thresholding strategy. *IEEE Trans. Image Process.* **16**(11), 2675–2681 (2007)
- Braides, A.: *Approximation of Free-Discontinuity Problems*. Lecture Notes in Mathematics, vol. 1694. Springer, Berlin (1998)
- Braides, A., Dal Maso, G.: Non-local approximation of the Mumford-Shah functional. *Calc. Var. Partial Differ. Equ.* **5**, 293–322 (1997)
- Bredies, K., Holler, M.: Regularization of linear inverse problems with total generalized variation. *J. Inverse Ill-Posed Probl.* **22**(6), 871–913 (2014)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
- Chambolle, A.: Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM J. Appl. Math.* **55**(3), 827–863 (1995)
- Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numer. Math.* **76**(2), 167–188 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Combettes, P.L., Pesquet, J.-C.: Proximal splitting methods in signal processing. In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer Optimization and Its Applications, vol. 49, pp. 185–212. Springer, New York (2011)

- Condat, L.: A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.* **158**(2), 460–479 (2013)
- Daubechies, I., Teschke, G.: Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising. *Appl. Comput. Harmon. Anal.* **19**(1), 1–16 (2005)
- Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**(11), 1413–1457 (2004)
- De los Reyes, J.C., Schönlieb, C.-B.: Image denoising: learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Probl. Imaging* **7**(4), 1183–1214 (2013)
- de Vito, E., Kereta, Z., Naumova, V.: Unsupervised parameter selection for denoising with the elastic net (2018)
- Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. *Ann. Stat.* **32**(2), 407–499 (2004)
- Field, D.J., Olshausen, B.A.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996)
- Fornasier, M., March, R., Solombrino, F.: Existence of minimizers of the Mumford-Shah functional with singular operators and unbounded data. *Ann. Mat. Pura Appl.* **192**(3), 361–391 (2013)
- Gobbino, M.: Finite difference approximation of the Mumford-Shah functional. *Commun. Pure Appl. Math.* **51**(2), 197–228 (1998)
- Grasmair, M., Muszkieta, M., Scherzer, O.: An approach to the minimization of the Mumford-Shah functional using Γ -convergence and topological asymptotic expansion. *Interfaces Free Bound.* **15**(2), 141–166 (2013)
- Grasmair, M., Klock, T., Naumova, V.: Adaptive multi-penalty regularization based on a generalized lasso path. *Appl. Comput. Harmon. Anal.* **49**(1), 30–55 (2018)
- Gribonval, R., Schnass, K.: Dictionary identifiability – sparse matrix-factorisation via l_1 -minimisation. *IEEE Trans. Inf. Theory* **56**(7), 3523–3539 (2010)
- Hansen, P.C.: Analysis of discrete ill-posed problems by means of the L -curve. *SIAM Rev.* **34**(4), 561–580 (1992)
- Ito, K., Jin, B., Takeuchi, T.: Multi-parameter tikhonov regularization – an augmented approach. *Chin. Ann. Math. Ser. B* **35**(3), 383–398 (2014)
- Komodakis, N., Pesquet, J.: Playing with duality: an overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Sig. Process. Mag.* **32**(6), 31–54 (2015)
- Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. *SIAM J. Imaging Sci.* **6**(2), 938–983 (2013)
- Lu, S., Pereverzev, S.V.: Multi-parameter regularization and its numerical realization. *Numer. Math.* **118**(1), 1–31 (2011)
- Lu, Y., Shen, L., Xu, Y.: Multi-parameter regularization methods for high-resolution image reconstruction with displacement errors. *IEEE Trans. Circuits Syst. I: Regul. Pap.* **54**(8), 1788–1799 (2007)
- Lu, S., Pereverzev, S.V., Shao, Y., Tautenhahn, U.: Discrepancy curves for multi-parameter regularization. *J. Inverse Ill-Posed Prob.* **18**(6), 655–676 (2010)
- Mairal, J., Bach, F., Ponce, J.: Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 791–804 (2012)
- Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. University Lecture Series, vol. 22. American Mathematical Society, Providence (2001). The fifteenth Dean Jacqueline B. Lewis memorial lectures
- Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
- Pock, T., Cremers, D., Bischof, H., Chambolle, A.: Global solutions of variational models with convex regularization. *SIAM J. Imaging Sci.* **3**(4), 1122–1145 (2010)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Applied Mathematical Sciences, vol. 167. Springer, New York (2009)

- Starck, J.-L., Elad, M., Donoho, D.: Redundant multiscale transforms and their application for morphological component separation. In: *Advances in Imaging and Electron Physics*, pp. 287–348. Elsevier, London (2004)
- Starck, J.-L., Elad, M., Donoho, D.L.: Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Process.* **14**(10), 1570–1582 (2005)
- Starck, J.-L., Murtagh, F., Fadili, J.: *Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis*, 2nd edn. Cambridge University Press, New York (2015)
- Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.* **19**(1–3), 553–572 (2003). Special issue in honor of the sixtieth birthday of Stanley Osher
- Vũ, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.* **38**(3), 667–681 (2013)
- Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**(2), 301–320 (2005)



Generative Adversarial Networks for Robust Cryo-EM Image Denoising

26

Hanlin Gu, Yin Xian, Ilona Christy Unarta, and Yuan Yao

Contents

Introduction	971
Robust Denoising in Deep Learning	971
Challenges of Cryo-EM Image Denoising	971
Outline	973
Background: Data Representation and Mapping	974
Autoencoder	974
GAN	975
Robust Denoising Method	976
Huber Contamination Noise Model	976
Robust Denoising Method	977
Robust Recovery via β -GAN	978
Stabilized Robust Denoising by Joint Autoencoder and β -GAN	981
Application: Robust Denoising of Cryo-EM Images	981
Datasets	981
Evaluation Method	984

This research made use of the computing resources of the X-GPU cluster supported by the Hong Kong Research Grant Council Collaborative Research Fund: C6021-19EF. The research of Hanlin Gu and Yuan Yao is supported in part by HKRGC 16308321, ITF UIM/390, the Hong Kong Research Grant Council NSFC/RGC Joint Research Scheme N_HKUST635/20, as well as awards from Tencent AI Lab and Si Family Foundation. We would like to thank Dr. Xuhui Huang for helpful discussions.

H. Gu · I. C. Unarta · Y. Yao (✉)

Hong Kong University of Science and Technology, Hong Kong, China
e-mail: hguaf@connect.ust.hk; icunarta@connect.ust.hk; yuany@ust.hk

Y. Xian

Hong Kong Applied Science and Technology Research Institute (ASTRI), Hong Kong, China

Network Architecture and Hyperparameter.....	986
Results for RNAP.....	986
Results for EMPIAR-10028.....	990
Conclusion.....	991
Appendix.....	992
Influence of Parameter(α , β) Brings in β -GAN.....	992
Clustering to Solve the Conformational Heterogeneity.....	992
Convolution Network.....	994
Test RNAP Dataset with PGGAN Strategy.....	995
Influence of the Regularization Parameter: λ	997
References.....	997

Abstract

The cryo-electron microscopy (cryo-EM) becomes popular for macromolecular structure determination. However, the 2D images captured by cryo-EM are of high noise and often mixed with multiple heterogeneous conformations and contamination, imposing a challenge for denoising. Traditional image denoising methods and simple denoising autoencoder cannot work well when the signal-to-noise ratio (SNR) of images is meager and contamination distribution is complex. Thus it is desired to develop new effective denoising techniques to facilitate further research such as 3D reconstruction, 2D conformation classification, and so on. In this chapter, we approach the robust denoising problem for cryo-EM images by introducing a family of generative adversarial networks (GANs), called β -GAN, which is able to achieve robust estimation of certain distributional parameters under Huber contamination model with statistical optimality. To address the denoising challenges, for example, the traditional image generative model might be contaminated by a small portion of unknown outliers, β -GANs are exploited to enhance the robustness of denoising autoencoder. Our proposed method is evaluated by both a simulated dataset on the *Thermus aquaticus* RNA polymerase (RNAP) and a real-world dataset on the *Plasmodium falciparum* 80S ribosome dataset (EMPIAR-10028), in terms of mean square error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and 3D reconstruction as well. Quantitative comparisons show that equipped with some designs of β -GANs and the robust ℓ_1 -autoencoder, one can stabilize the training of GANs and achieve the state-of-the-art performance of robust denoising with low SNR data and against possible information contamination. Our proposed methodology thus provides an effective tool for robust denoising of cryo-EM 2D images and helpful for 3D structure reconstruction.

Keywords

Generative adversarial networks · Autoencoder · Robust statistics · Denoising · Cryo-electron microscopy

Introduction

Robust Denoising in Deep Learning

Deep learning technique has rapidly entered into the field of image processing. One of the most popular methods was the denoising autoencoder (DA) motivated by Vincent et al. (2008). It used the reference data to learn a compressed representation (encoder) for the dataset. One extension of DA was presented in Xie et al. (2012), which exploited the sparsity regularization and the reconstruction loss to avoid over-fitting. Other developments, such as Zhang et al. (2017), made use of the residual network architecture to improve the quality of denoised images. In addition, Agostinelli et al. (2013) combined several sparse denoising autoencoders to enhance the robustness under different noise.

The generative adversarial network (GAN) recently gained its popularity and provides a promising new approach for image denoising. GAN was proposed by Goodfellow et al. (2014) and was mainly composed of two parts: the generator (G : generate the new samples) and the discriminator (D : determine whether the samples are real or generated (fake)). Original GAN (Goodfellow et al. 2014) aimed to minimize the Jensen-Shannon (JS) divergence between distributions of the generated samples and the true samples, hence called JS-GAN. Various GANs were then studied, and in particular, Arjovsky et al. (2017) proposed the Wasserstein GAN (WGAN), which replaced the JS divergence with Wasserstein distance. Gulrajani et al. (2017) further improved the WGAN with the gradient penalty that stabilized the model training. For image denoising problem, GAN could better describe the distribution of original data by exploiting the common information of samples. Consequently, GANs were widely applied in the image denoising problem (Tran et al. 2020; Tripathi et al. 2018; Yang et al. 2018; Chen et al. 2018; Dong et al. 2020).

Recently, Gao et al. (2019, 2020) showed that a general family of GANs (β -GANs, including JS-GAN and TV-GAN) enjoyed robust reconstruction when the datasets contain outliers under Huber contamination models (Huber 1992). In this case, observed samples are drawn from a complex distribution, which is a mixture of contamination distribution and real data distribution. A particular example is provided by cryo-electron microscopy (cryo-EM) imaging, where the original noisy images are likely contaminated with outliers as broken or non-particles. The main challenges of cryo-EM image denoising are summarized in the subsequent section.

Challenges of Cryo-EM Image Denoising

The cryo-electron microscopy (cryo-EM) has become one of the most popular techniques to resolve the atomic structure. In the past, cryo-EM was limited to large complexes or low-resolution models. Recently the development of new detector

hardware has dramatically improved the resolution in cryo-EM (Kühlbrandt 2014), which made cryo-EM widely used in a variety of research fields. Different from X-ray crystallography (Warren 1990), cryo-EM had the advantage of preventing the recrystallization of inherent water and recontamination. Also, cryo-EM was superior to nuclear magnetic resonance spectroscopy (Wüthrich 1986) in solving macromolecules in the native state. In addition, both X-ray crystallography and nuclear magnetic resonance spectroscopy required large amounts of relatively pure samples, whereas cryo-EM required much fewer samples (Bai et al. 2015). For this celebrated development of cryo-EM for the high-resolution structure determination of biomolecules in solution, the Nobel Prize in Chemistry in 2017 was awarded to three pioneers in this field (Shen 2018).

However, it is a computational challenge in processing raw cryo-EM images, due to heterogeneity in molecular conformations and high noise. Macromolecules in natural conditions are usually heterogeneous, i.e., multiple metastable structures might coexist in the experimental samples (Frank 2006; Scheres 2016). Such conformational heterogeneity adds extra difficulty to the structural reconstruction as we need to assign each 2D image to not only the correct projection angle but also its corresponding conformation. This imposes a computational challenge that one needs to denoise the cryo-EM images without losing the key features of their corresponding conformations. Moreover, in the process of generating cryo-EM images, one needs to provide a view using the electron microscope for samples that are in frozen condition. Thus there are two types of noise: one is from ice, and the other is from the electron microscope. Both of them are significant in contributing high noise in cryo-EM images and leave a difficulty to the detection of particle structures (Fig. 1 shows a typical noisy cryo-EM image with its reference image, which is totally non-identifiable to human eyes). In extreme cases, some experimental images even do not contain any particles, rendering it difficult for particle picking either manually or automatically (Wang et al. 2016). How to

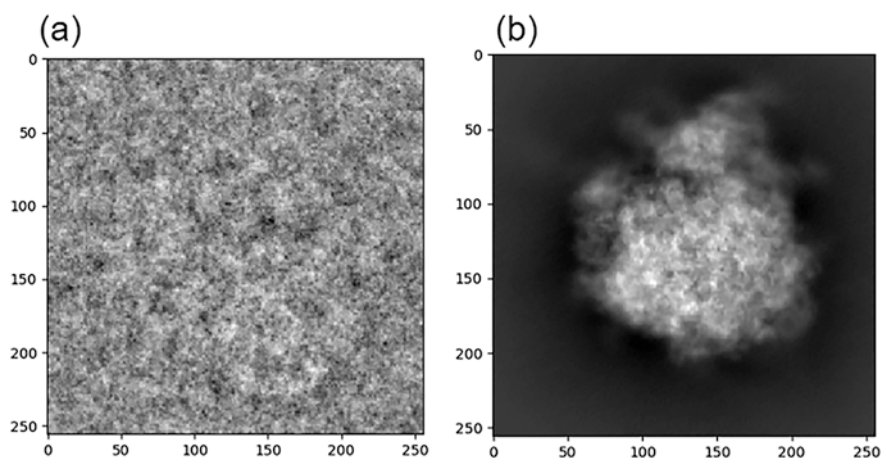


Fig. 1 (a) A noisy cryo-EM image; (b) a reference image

achieve robust denoising against such kind of contamination thus becomes a critical problem. Therefore, it is a great challenge to develop robust denoising methods for cryo-EM images to reconstruct heterogeneous biomolecular structures.

There are a plethora of denoising methods developed in applied mathematics and machine learning that could be applied to cryo-EM image denoising. Most of them in cryo-EM are based on unsupervised learning, which don't need any reference image data to learn. Wang and Yin (2013) proposed a filtering method based on nonlocal means, which made use of the rotational symmetry of some biological molecules. Also, Wei and Yin (2010) designed the adaptive nonlocal filter, which made use of a wide range of pixels to estimate the denoised pixel values. Besides, Xian et al. (2018) compared transform domain filtering method, BM3D (Dabov et al. 2007), and dictionary learning method, KSVD (Aharon et al. 2006), in denoising problem in cryo-EM. However, all of these didn't work well in low signal-to-noise ratio (SNR) situations. In addition, Covariance Wiener Filtering (CWF) (Bhamre et al. 2016) was proposed for image denoising. However, CWF needed large sample size of data in order to estimate the covariance matrix correctly, although it had an attractive denoising effect. Therefore, a robust denoising method in cryo-EM images was needed.

Outline

In this chapter, we propose a robust denoising scheme of cryo-EM images by exploiting joint training of both autoencoders and a new type of GANs β -GANs. Our main results are summarized as follows:

- Both autoencoder and GANs help each other for cryo-EM denoising in low signal-to-noise ratio scenarios. On the one hand, autoencoder helps stabilize GANs during training, without which the training processes of GANs are often collapsed due to high noise; on the other hand, GANs help autoencoder in denoising by sharing information in similar samples via distribution learning. For example, WGAN combined with autoencoder often achieve state-of-the-art performance due to its ability of exploiting information in similar samples for denoising.
- To achieve robustness against partial contamination of samples, one needs to choose both robust reconstruction loss for autoencoder (e.g., ℓ_1 loss) and robust β -GANs (e.g., (.5, .5)-GAN or (1, 1)-GAN,¹ which is proved to be robust against Huber contamination) that achieve competitive performance with WGANs in contamination-free scenarios, but do not deteriorate that much with data contamination.
- Numerical experiments are conducted with both a heterogeneous conformational dataset on the *Thermus aquaticus* RNA polymerase (RNAP) and a homogenous

¹ β -GAN has two parameters: α and β , written as (α, β) -GAN in this chapter.

dataset on the *Plasmodium falciparum* 80S ribosome dataset (EMPIAR-10028). The experiments on those datasets show the validity of the proposed methodology and suggest that while WGAN, (.5, .5)-GAN, and (1, 1)-GAN combined with ℓ_1 -autoencoder are among the best choices in contamination-free cases, the latter two are overall the most recommended for robust denoising.

To achieve the goals above, this chapter is to provide an overview on various developments of GANs with their robustness properties. After that we focus on the application to the challenge of cryo-EM image robust denoising problem.

The chapter is structured as follows. In section “[Background: Data Representation and Mapping](#),” we provide a general overview of autoencoder and GAN. In section “[Robust Denoising Method](#),” we model the tradition denoising problem based on Huber contamination firstly and discuss β -GAN and its statistics. Finally, we give our algorithm based on combination of β -GAN and autoencoder, which is training stable. The evaluation of the algorithm in cryo-EM data is shown in the section “[Application: Robust Denoising of Cryo-EM Images](#).” The section “[Conclusion](#)” concludes the chapter. In addition, we implement the supplementary experiment in the section “[Appendix](#).”

Background: Data Representation and Mapping

Efficient representation learning of data distribution is crucial for many machine learning-based models. For a set of the real data samples X , the classical way to learn the probability distribution of this data (P_r) is to find P_θ by minimizing the distance between P_r and P_θ , such as Kullback-Leibler divergence $D_{KL}(P_r||P_\theta)$. This means we can pass a random variable through a parametric function to generate samples following a certain distribution P_θ instead of directly estimating the unknown distribution P_r . By varying θ , we can change this distribution and make it close to the real data distribution P_r . Autoencoder and GANs are two well-known methods in data representation. Autoencoder is good at learning the representation of data with low dimensions with an explicit characterization of P_θ , while GAN offers flexibility in defining the objective function (such as the Jensen-Shannon divergence) by directly generating samples without explicitly formulating P_θ .

Autoencoder

Autoencoder (Baldi 2012) is a type of neural network used to learn efficient codings of unlabeled data. It learns a representation (encoding) for a set of data, typically for dimensional reduction by training the network. An autoencoder has two main parts: encoder and decoder. The encoder maps the input data x ($\in X$) into a latent representation z , while the decoder maps the latent representation back to the data space:

$$z \sim \text{Enc}(x) \quad (1)$$

$$\hat{x} \sim \text{Dec}(z). \quad (2)$$

Autoencoders are trained to minimize reconstruction errors, such as $\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$.

Various techniques have been developed to improve data representation ability for autoencoder, such as imposing regularization on the encoding layer:

$$\mathcal{L}(x, \hat{x}) + \Omega(h), \quad (3)$$

where h is the mapping function of the encoding layer and $\Omega(h)$ is the regularization term. The autoencoder is good at data denoising and dimension reduction.

GAN

The generative adversarial network (GAN), firstly proposed by Goodfellow (Goodfellow et al. 2014, called JS-GAN), is a class of machine learning framework. The goal of GAN is to learn to generate new data with the same statistics as the training set. Though original GAN is proposed as a form of generative model for unsupervised learning, GAN has proven useful for semi-supervised learning, fully supervised learning, and reinforcement learning (Hua et al. 2019; Sarmad et al. 2019; Dai et al. 2017).

Although GAN has shown great success in machine learning, the training of GAN is not easy and is known to be slow and unstable. The problems of GAN (Bau et al. 2019; Arjovsky et al. 2017) include:

- *Hard to achieve Nash equilibrium.* The updating process of the generator and the discriminator models are hard to guarantee a convergence.
- *Vanishing gradient.* The gradient update is slow when the discriminator is well trained.
- *Mode collapse.* The generator fails to generate samples with enough representative.

JS-GAN

JS-GAN proposed in Goodfellow et al. (2014) took Jensen-Shannon (JS) distance to measure the difference between different data distributions. The mathematics expression is follows:

$$\min_G \max_D \mathbb{E}_{x \sim P(X), z \sim P(Z)} [\log D(x) + \log(1 - D(G(z)))] \quad (4)$$

where G is a generator which maps disentangled noise $z \sim P(Z)$ (usually Gaussian $N(0, I)$) to fake image data in a purpose to confuse the discriminator D from real data. The discriminator D is simply a classifier, which makes an effort to distinguish

real data from the fake data generated by G . $P(X)$ is the input data distribution. z is noise. $P(Z)$ is the noise distribution, and it is used for data generation. Training of GANs is a minimax game by alternatively updating generators and discriminators, where the purpose of generators is to fool the discriminator as an adversarial process.

WGAN and WGANgp

Wasserstein GAN (Arjovsky et al. 2017) replaced the JS distance with the Wasserstein distance:

$$\min_G \max_D \mathbb{E}_{x \sim P(X), z \sim P(z)} \{D(x) - D(G(z))\}. \quad (5)$$

In reality, WGAN applied weight clipping of neural network to satisfy Lipschitz condition for discriminator. Moreover, Gulrajani et al. (2017) proposed WGANgp based on WGAN, which introduced a penalty in gradient to stabilize the training:

$$\min_G \max_D \mathbb{E}_{(x,z) \sim P(X,z)} \{D(x) - D(G(z)) + \mu \mathbb{E}_{\tilde{x}} (\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2\}, \quad (6)$$

where \tilde{x} is uniformly sampled along straight lines connecting pairs of the generated and real samples and μ is a weighting parameter. In WGANgp, the last layer of the sigmoid function in the discriminator network is removed. Thus D 's output range is the whole real \mathbb{R} , but its gradient is close to 1 to achieve Lipschitz-1 condition.

Robust Denoising Method

Huber Contamination Noise Model

Let $x \in \mathbb{R}^{d_1 \times d_2}$ be a clean image, often called reference image in the sequel. The generative model of noisy image $y \in \mathbb{R}^{d_1 \times d_2}$ under the linear, weak phase approximation (Bhamre et al. 2016) could be described by

$$y = a * x + \zeta, \quad (7)$$

where $*$ denotes the convolution operation, a is the point spread function of the microscope convolving with the clean image, and ζ is an additive noise, usually assumed to be Gaussian noise that corrupts the image. In order to remove the noise the microscope brings, traditional denoising autoencoder could be exploited to learn from examples $(y_i, x_i)_{i=1, \dots, n}$ the inverse mapping a^{-1} from the noisy image y to the clean image x .

However, this model is not sufficient in the real case. In the experimental data, the contamination will significantly affect the denoising efficiency if the denoising methods continuously depend on the sample outliers. Therefore we introduce the

following Huber contamination model to extend the image formation model (see Eq. (7)).

Consider that the pair of reference image and experimental image (x, y) is subject to the following mixture distribution P_ϵ :

$$P_\epsilon = (1 - \epsilon)P_0 + \epsilon Q, \quad \epsilon \in [0, 1], \quad (8)$$

a mixture of true distribution P_0 of probability $(1 - \epsilon)$ and arbitrary contamination distribution Q of probability ϵ . P_0 is characterized by Eq. (7), and Q accounts for the unknown contamination distribution possibly due to ice, broken of data, and so on such that the image sample does not contain any particle information. This is called the Huber contamination model in statistics (Huber 1992). Our purpose is that given n samples $(x_i, y_i) \sim P_\epsilon$ ($i = 1, \dots, n$), possibly contaminated with unknown Q , learn a robust inverse map $a^{-1}(y)$.

Robust Denoising Method

Exploit a neural network to approximate the robust inverse mapping $G_\theta : \mathbb{R}^{d_1 \times d_2} \rightarrow \mathbb{R}^{d_1 \times d_2}$. The neural network is parameterized by $\theta \in \Theta$. The goal is to ensure that discrepancy between reference image x and reconstructed image $\hat{x} = G_\theta(y)$ is small. Such a discrepancy is usually measured by some nonnegative loss function: $\ell(x, \hat{x})$. Therefore, the denoising problem minimizes the following expected loss:

$$\arg \min_{\theta \in \Theta} \mathcal{L}(\theta) := \mathbb{E}_{x,y}[\ell(x, G_\theta(y))]. \quad (9)$$

In practice, given a set of training samples $S = \{(x_i, y_i) : i = 1, \dots, n\}$, we aim to solve the following empirical loss minimization problem:

$$\arg \min_{\theta \in \Theta} \widehat{\mathcal{L}}_S(\theta) := \frac{1}{n} \sum_{i=1}^n \ell(x_i, G_\theta(y_i)). \quad (10)$$

The following choices of loss functions are considered:

- (**ℓ_2 -Autoencoder**) $\ell(x, \hat{x}) = \frac{1}{2} \|x - \hat{x}\|_2^2 := \frac{1}{2} \sum_{i,j} (x_{ij} - \hat{x}_{ij})^2$, or $\mathbb{E}\ell(x, \hat{x}) = D_{KL}(p(x) \| q(\hat{x}_\theta))$ equivalently, where $\hat{x}_\theta \sim \mathcal{N}(x, \sigma^2 I_D)$;
- (**ℓ_1 -Autoencoder**) $\ell(x, \hat{x}) = \|x - \hat{x}\|_1 := \sum_{i,j} |x_{ij} - \hat{x}_{ij}|$, or $\mathbb{E}\ell(x, \hat{x}) = D_{KL}(p(x) \| q(\hat{x}_\theta))$ equivalently, where $\hat{x}_\theta \sim \text{Laplace}(x, b)$;
- (**Wasserstein-GAN**) $\ell(x, \hat{x}) = W_1(p(x), q_\theta(\hat{x}))$, where W_1 is the 1-Wasserstein distance between distributions of x and \hat{x} ;
- (**β -GAN**) $\ell(x, \hat{x}) = D(p(x) \| q_\theta(\hat{x}))$, where D is some divergence function to be discussed below between distributions of x and \hat{x} .

Both the ℓ_2 and ℓ_1 losses consider the reconstruction error of G_θ . The ℓ_2 loss above is equivalent to assume that $G_\theta(y|x)$ follows a Gaussian distribution $\mathcal{N}(x, \sigma^2 I_D)$, and the ℓ_1 loss instead assumes a Laplacian distribution centered at x . As a result, the ℓ_2 loss pushes the reconstructed image \hat{x} toward mean by averaging out the details and thus blurs the image. On the other hand, the ℓ_1 loss pushes \hat{x} toward the coordinate-wise median, keeping the majority of details while ignoring some large deviations. It improves the reconstructed image and is more robust than the ℓ_2 loss against large outliers. Although ℓ_1 -autoencoder has a more robust loss than ℓ_2 , both of them are not sufficient to handle the contamination. In the framework of the Huber contamination model (Eq. (8)), β -GAN is introduced below.

Robust Recovery via β -GAN

Recently Gao et al. (2019, 2020) came up with a more general form of β -GAN. It aims to solve the following minimax optimization problem to find the G_θ :

$$\min_{G_\theta} \max_D \mathbb{E}[S(D(x), 1) + S(D(G_\theta(y)), 0)], \quad (11)$$

where $S(t, 1) = -\int_t^1 c^{\alpha-1}(1-c)^\beta dc$, $S(t, 0) = -\int_0^t c^\alpha(1-c)^{\beta-1} dc$, $\alpha, \beta \in [-1, 1]$. For simplicity, we denote this family with parameters α, β by (α, β) -GAN in this chapter.

The family of (α, β) -GAN includes many popular members. For example, when $\alpha = 0, \beta = 0$, it becomes the JS-GAN (Goodfellow et al. 2014), which aims to solve the minmax problem (Eq. (4)) whose loss is the Jensen-Shannon divergence. When $\alpha = 1, \beta = 1$, the loss is a simple mean square loss; when $\alpha = -0.5, \beta = -0.5$, the loss is boost score.

However, the Wasserstein GAN (WGAN) is not a member of this family. By formally taking $S(t, 1) = t$ and $S(t, 0) = -t$, we could derive the type of WGAN as Eq. (5).

Robust Recovery Theory

Extend the traditional image generative model to a Huber contamination model, and exploit the β -GAN toward robust denoising under unknown contamination. Below includes a brief introduction to robust β -GAN, which achieves provable robust estimate or recovery under Huber contamination model. Recently, Gao establishes the statistical optimality of β -GANs for robust estimate of mean (location) and covariance (scatter) of the general elliptical distributions (Gao et al. 2019, 2020). Here we introduce the main results.

Definition 1 (Elliptical Distribution). A random vector $X \in \mathbb{R}^p$ follows an elliptical distribution if and only if it has the representation $X = \theta + \xi AU$, where

$\theta \in \mathbb{R}^p$ and $A \in \mathbb{R}^{p \times r}$ are model parameters. The random variable U is distributed uniformly on the unit sphere $\{u \in \mathbb{R}^r : \|u\| = 1\}$, and $\xi \geq 0$ is a random variable in \mathbb{R} independent of U . The vector θ and the matrix $\Sigma = AA^T$ are called the location and the scatter of the elliptical distribution.

Normal distribution is just a member in this family characterized by mean θ and covariance matrix Σ . Cauchy distribution is another member in this family whose moments do not exist.

Definition 2 (Huber Contamination Model). $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} (1 - \epsilon)P_{\text{ell}} + \epsilon Q$, where we consider the P_{ell} an elliptical distribution in its canonical form.

A more general data-generating process than Huber contamination model is called the strong contamination model below, as the TV neighborhood of a given elliptical distribution P_{ell} :

Definition 3 (Strong Contamination Model). $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} P$, for some P satisfying

$$TV(P, P_{\text{ell}}) < \epsilon.$$

Definition 4 (Discriminator Class). Let $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$, $\text{ramp}(x) = \max(\min(x+1/2, 1), 0)$, and $\text{ReLU}(x) = \max(x, 0)$. Define a general discriminator class of deep neural nets: firstly define the a ramp bottom layer

$$\mathcal{G}_{\text{ramp}} = g(x) = \text{ramp}(u^t x + b), u \in \mathbb{R}^p, b \in \mathbb{R}. \tag{12}$$

Then, with $\mathcal{G}_1(B) = \mathcal{G}_{\text{ramp}}$, inductively define

$$\mathcal{G}_{l+1}(B) = \left\{ g(x) = \text{ReLU}\left(\sum_{h \geq 1} v_h g_h(x)\right) : \sum_{h \geq 1} |v_h| \leq B, g_h \in \mathcal{G}_l(B) \right\}. \tag{13}$$

Note that the neighboring two layers are connected via ReLU activation functions. Finally, the network structure is defined by

$$\mathcal{D}^L(\kappa, B) = \left\{ D(x) = \text{sigmoid}\left(\sum_{j \geq 1} w_j g_j(x)\right) : \sum_{j \geq 1} |w_j| \leq \kappa, g_j \in \mathcal{G}_L(B) \right\}. \tag{14}$$

This is a network architecture consisting of L hidden layers.

Now consider the following β -GAN induced by a proper scoring rule $S : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}$ with the discriminator class above:

$$(\hat{\theta}, \hat{\Sigma}) = \arg \min_{(\theta, \Sigma)} \max_{D \in \mathcal{D}^L(\kappa, B)} \frac{1}{n} \sum_{i=1}^n S(D(x_i), 1) + \mathbb{E}_{x \sim P_{\text{ell}}(\theta, \Sigma)} S(D(x), 0). \tag{15}$$

The following theorem shows that such a β -GAN may give a statistically optimal estimate of location and scatter of the general family of elliptical distributions under strong contamination models.

Theorem 1 (Gao et al. 2020). *Consider the (α, β) -GANs with $|\alpha - \beta| < 1$. The discriminator class $D = \mathcal{D}^L(k, B)$ is specified by Eq. (14). Assume $\frac{p}{n} + \epsilon^2 \leq c$ for some sufficiently small constant $c > 0$. Set $1 \leq L = O(1)$, $1 \leq B = O(1)$, and $\kappa = O(\sqrt{\frac{p}{n} + \epsilon})$. Then for any $X_1, \dots, X_n \stackrel{iid}{\sim} P$, for some P satisfying $TV(P, P_{\text{ell}}) < \epsilon$ with small enough ϵ , we have*

$$\begin{aligned} \|\hat{\theta} - \theta\|^2 &< C\left(\frac{p}{n} \vee \epsilon^2\right), \\ \|\hat{\Sigma} - \Sigma\|_{op}^2 &< C\left(\frac{p}{n} \vee \epsilon^2\right), \end{aligned} \tag{16}$$

with probability at least $1 - e^{C'(p+n\epsilon^2)}$ (universal constants C and C') uniformly over all $\theta \in \mathbb{R}^p$ and all $\|\Sigma\|_{op} \leq M$.

The theorem established that for all $|\alpha - \beta| < 1$, (α, β) -GAN family is robust in the sense that one can learn a distribution P_{ell} from contaminated distributions P_ϵ such that $TV(P_\epsilon, P_{\text{ell}}) < \epsilon$, which includes Huber contamination model as a special case. Therefore a (α, β) -GAN with suitable choice of network architecture can robustly learn the generative model from arbitrary contamination Q when ϵ is small (e.g., no more than $1/3$).

In the current case, the denoising autoencoder network is modified to $G_\theta(y)$, providing us a universal approximation of the location (mean) of the inverse generative model as Eq. (7), where the noise can be any member of the elliptical distribution. Moreover, the discriminator is adapted to the image classification problem in the current case. Equipped with this design, the proposed (α, β) -GAN may help enhance the denoising autoencoder robustness against unknown contamination, e.g., the Huber contamination model for real contamination in the image data. The experimental results in fact confirm the efficacy of such a design.

In addition, Wasserstein GAN (WGAN) is not a member of this β -GAN family. Compared to JS-GAN, WGAN aims to minimize the Wasserstein distance between the sample distribution and the generator distribution. Therefore, WGAN is not robust in the sense of contamination models above as arbitrary ϵ portion of outliers can be far away from the main distribution P_0 such that the Wasserstein distance is arbitrarily large.

Stabilized Robust Denoising by Joint Autoencoder and β -GAN

Although β -GAN can robustly recover model parameters with contaminated samples, as a zero-sum game involving a non-convex-concave minimax optimization problem, training GANs is notoriously unstable with typical cyclic dynamics and possible mode collapse entrapped by local optima (Arjovsky et al. 2017). However, in this section we show that the introduction of autoencoder loss is able to stabilize the training and avoid the mode collapse. In particular, autoencoder can help stabilize GAN during training, without which the training processes of GAN are often oscillating and sometimes collapsed due to the presence of high noise.

Compared with the autoencoder, β -GAN can further help denoising by exploiting common information in similar samples during distribution training. In GAN, the divergence or Wasserstein distance between the reference image set and the denoised image set is minimized. The similar images can therefore help boost signals for each other.

For these considerations, a combined loss is proposed with both β -GAN and autoencoder reconstruction loss:

$$\widehat{\mathcal{L}}_{GAN}(x, \widehat{x}) + \lambda \|x - \widehat{x}\|_p^p, \quad (17)$$

where $p \in \{1, 2\}$ and $\lambda \geq 0$ is a trade-off parameter for ℓ_p reconstruction loss. Algorithm 1 summarizes the procedure of joint training of autoencoder and GAN, which will be denoted as “GAN+ ℓ_p ” in the experimental section depending on the proper choice of GAN and p . The main algorithm is shown in Algorithm 1.

Stability of Combining Autoencoder into GAN

We illustrate that autoencoder is indispensable to GANs in stabilizing the training in the joint training of autoencoder and GAN scheme.

As an illustration, Fig. 2 shows the comparison of training a JS-GAN and a joint JS-GAN + ℓ_1 -autoencoder. Training and test mean square error curves are plotted against iteration numbers in the RNAP data under $SNR = 0.1$ as Fig. 2. It shows that JS-GAN training suffers from drastic oscillations, while joint training of JS-GAN + ℓ_1 -autoencoder exhibits a stable process. In fact, with the aid of autoencoder here, one does not need the popular “log D trick” in JS-GAN.

Application: Robust Denoising of Cryo-EM Images

Datasets

RNAP: Simulation Dataset

We design a conformational heterogeneous dataset obtained by simulations. We use *Thermus aquaticus* RNA polymerase (RNAP) in complex with σ^A factor (*Taq* holoenzyme) for our dataset. RNAP is the enzyme that transcribes RNA

Algorithm 1 Joint training of (α, β) -GAN and ℓ_p -autoencoder

Input:

1. (α, β) for $S(t, 1) = - \int_t^1 c^{\alpha-1} (1-c)^\beta dc$, $S(t, 0) = - \int_0^t c^\alpha (1-c)^{\beta-1} dc$
 or $S(t, 1) = t$, $S(t, 0) = -t$ for WGAN
 2. λ regularization parameter of the ℓ_p -Autoencoder
 3. k_d number of iterations for discriminator, k_g number of iterations for generator
 4. η_d learning rate of discriminator, η_g learning rate of generator
 5. ω weights of discriminator, θ weights of generator
- 1: **for** number of training iterations **do**
 - 2: • Sample minibatch of m examples $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ from reference-noisy image pairs.
 - 3: **for** $k = 1, 2 \dots, k_d$ **do**
 - 4: • Update the discriminator by gradient ascent:
 - 5: $g_\omega \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\omega [S(D_\omega(x_i), 1) + S(D_\omega(G_\theta(y_i)), 0) + \mu(\|\nabla_{\tilde{x}} D_\omega(\tilde{x}_i)\|_2 - 1)^2]$
 where $\mu > 0$ for WGANgp only;
 - 6: $\omega \leftarrow \omega + \eta_d g_\omega$
 - 7: **end for**
 - 8: **for** $k = 1, 2 \dots, k_g$ **do**
 - 9: • Update the generator by gradient descent:
 - 10: $g_\theta \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_\theta [S(D_\omega(G_\theta(y_i)), 0) + \lambda |G_\theta(y_i) - x_i|^p]$, $p \in \{1, 2\}$;
 - 11: $\theta \leftarrow \theta - \eta_g g_\theta$
 - 12: **end for**
 - 13: **end for**

Return: Denoised image: $\hat{x}_i = G_\theta(y_i)$

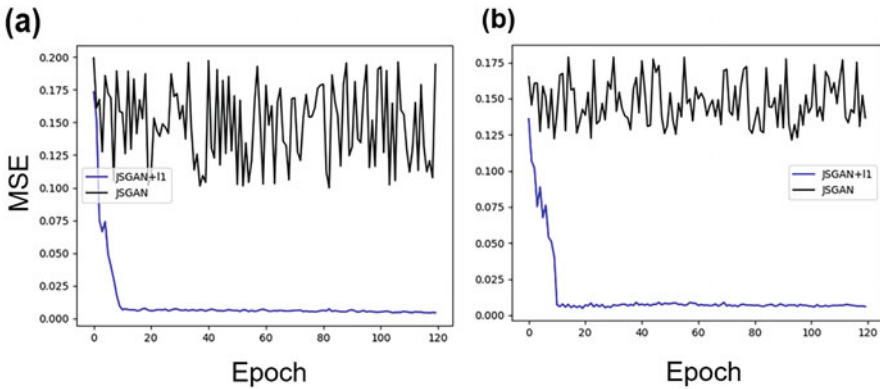


Fig. 2 Comparison between JS-GAN (black) and joint JS-GAN- ℓ_1 -autoencoder (blue). (a) and (b) are the change of MSE in training and testing data. Joint training of JS-GAN- ℓ_1 -autoencoder is much more stable than pure JS-GAN training that oscillates a lot

from DNA (transcription) in the cell. During the initiation of transcription, the holoenzyme must bind to the DNA and then separate the double-stranded DNA into single-stranded (Browning and Busby 2004). *Taq* holoenzyme has a crab-claw-like structure, with two flexible domains, the clamp and β pincers. The clamp, especially,

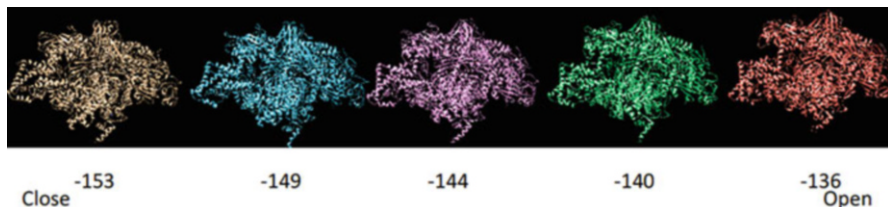


Fig. 3 Five conformations in RNAP heterogeneous dataset; from left to right are close conformation to open conformation of different angles

has been suggested to play an important role in the initiation, as it has been captured in various conformations by cryo-EM during initiation (Chen et al. 2020). Thus, we focus on the movement of the clamp in this study. To generate the heterogeneous dataset, we start with two crystal structures of *Taq* holoenzyme, which vary in their clamp conformation, open (PDB ID: 1L9U (Murakami et al. 2002)) and closed (PDB ID: 4XLN (Bae et al. 2015)) clamp. For the closed-clamp structure, we remove the DNA and RNA in the crystal structure, leaving only the RNAP and σ^A for our dataset. The *Taq* holoenzyme has about 370 kDa molecular weight. We then generate the clamp intermediate structures between the open and closed clamp using multiple-basin coarse-grained (CG) molecular dynamic (MD) simulations (Okazaki et al. 2006; Kenzaki et al. 2011). CG-MD simulations simplify the system such that the atoms in each amino acid are represented by one particle. The structures from CG-MD simulations are refined back to all-atom or atomic structures using PD2 ca2main (Moore et al. 2013) and SCRWL4 (Krivov et al. 2009). Five structures with equally spaced clamp opening angle are chosen for our heterogeneous dataset (shown in Fig. 3). Then, we convert the atomic structures to $128 \times 128 \times 128$ volumes using `Xmipp` package (Marabini et al. 1996) and generate the 2D projections with an image size of 128×128 pixels. We further contaminate those clean images with additive Gaussian noise at different signal-to-noise ratio (SNR): $SNR = 0.05$. The SNR is defined as the ratio of signal power and the noise power in the real space. For simplicity, we did not apply the contrast transfer function (CTF) to the datasets, and all the images are centered. Figure 3 shows the five conformation pictures.

Training data size is 25,000 paired images (noisy and reference images). Test data to calculate the MSE, PSNR, and SSIM is another 1500 paired images.

EMPIAR-10028: Real Dataset

This is a real-world experimental dataset that was firstly studied in the *Plasmodium falciparum* 80S ribosome dataset (EMPIAR-10028) (Wong et al. 2014). They recover the cryo-EM structure of the cytoplasmic ribosome from the human malaria parasite, *Plasmodium falciparum*, in complex with emetine, an anti-protozoan drug, at 3.2\AA resolution. Ribosome is the essential enzyme that translates RNA to protein molecules, the second step of central dogma. The inhibition of ribosome activity of *Plasmodium falciparum* would effectively kill the parasite (Wong et al. 2014). We can regard this dataset to have homogeneous property. This dataset contains

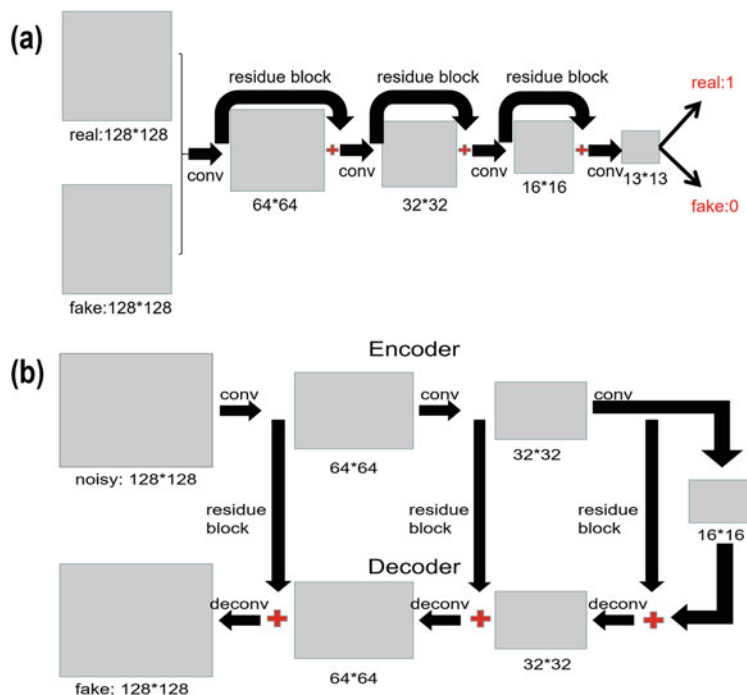


Fig. 4 The architectures of (a) discriminator D and (b) generator G , which borrow the residue structure. The input image size (128×128) here is adapted to RNAP dataset, while input image size of EMPIAR-10028 dataset is 256×256

105,247 noisy particles with an image size of 360×360 pixels. In order to decrease the complexity of the computing, we pick up the center square of each image with a size of 256×256 , since the surrounding area of the image is entirely useless that does not lose information in such a preprocessing. Then the 256×256 images are fed as the input of the G_{θ} -network (Fig. 4). Since the GAN-based method needs clean images as reference, we prepare their clean counterparts in the following way: we first use cryoSPARC1.0 (Punjani et al. 2017) to build a 3.2\AA resolution volume and then rotate the 3D volume by the Euler angles obtained by cryoSPARC to get projected 2D images. The training data size we pick is 19,500, and the test data size is 500.

Evaluation Method

We exploit the following three metrics to determine whether the denoising result is good or not. They are the mean square error (MSE), the peak signal-to-noise ratio (PSNR), and the structural similarity index measure (SSIM).

- **(MSE)** For images of size $d_1 \times d_2$, the mean square error (MSE) between the reference image x and the denoised image \hat{x} is defined as

$$\text{MSE} := \frac{1}{d_1 d_2} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (x(i, j) - \hat{x}(i, j))^2.$$

The smaller is the MSE, the better the denoising result is.

- **(PSNR)** Similarly, the peak signal-to-noise ratio (PSNR) between the reference image x and the denoised image \hat{x} whose pixel value range is $[0, t]$ (1 by default) is defined by

$$\text{PSNR} := 10 \log_{10} \frac{t^2}{\frac{1}{d_1 d_2} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} (x(i, j) - \hat{x}(i, j))^2}.$$

The larger is the PSNR, the better the denoising result is.

- **(SSIM)** The third criterion which is the structural similarity index measure (SSIM) between reference image x and denoised image \hat{x} is defined in (Wang et al. 2004):

$$\text{SSIM} = \frac{(2\mu_x \mu_{\hat{x}} + c_1)(2\sigma_x \sigma_{\hat{x}} + c_2)(\sigma_{x\hat{x}} + c_3)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)(\sigma_x \sigma_{\hat{x}} + c_3)}.$$

where μ_x ($\mu_{\hat{x}}$) and σ_x ($\sigma_{\hat{x}}$) are the mean and variance of x (\hat{x}), respectively; $\sigma_{x\hat{x}}$ is covariance of x and \hat{x} ; $c_1 = K_1 L^2$, $c_2 = K_2 L^2$, and $c_3 = \frac{c_2}{2}$ are three variables to stabilize the division with weak denominator ($K_1 = 0.01$, $K_2 = 0.03$ by default); and L is the dynamic range of the pixel value (1 by default). The value of SSIM lies in $[0, 1]$, where the closer it is to 1, the better the result is.

Although these metrics are widely used in image denoising, they might not be the best metrics for cryo-EM images. In Appendix “[Influence of the Regularization Parameter: \$\lambda\$](#) ,” it shows an example that the best-reconstructed images perhaps do not meet the best MSE/PSNR/SSIM metrics.

In addition to these metrics, we consider the 3D reconstruction based on denoised images. Particularly, we take the 3D reconstruction by RELION to validate the denoised result. The procedure of our RELION reconstruction is as follows: firstly creating the 3D initial model, then doing 3D classification, followed by operating 3D auto-refine. Moreover, for heterogeneous conformations in simulation data, we further turn the denoising results into a clustering problem to measure the efficacy of denoising methods, whose details will be discussed in Appendix “[Clustering to Solve the Conformational Heterogeneity](#).”

Network Architecture and Hyperparameter

In the experiments of this chapter, the best results come from the ResNet architecture (Su et al. 2018) shown in Fig. 4, which has been successfully applied to study biological problems such as predicting protein-RNA binding. The generator in such GANs exploits the autoencoder network architecture, while the discriminator is a binary classification ResNet. In Appendix “Convolution Network” and “Test RNAP Dataset with PGGAN Strategy,” we also discuss a convolutional network without residual blocks and the PGGAN (Karras et al. 2018) strategy with their experimental results, respectively.

We chose Adam (Kingma and Ba 2015) for the optimization. The learning rate of the discriminator is $\eta_d = 0.001$, and the learning rate of the generator is $\eta_g = 0.01$. We choose $m = 20$ as our batch size, $k_d = 1$, and $k_g = 2$ in Algorithm 1.

For (α, β) -GAN, we report two types of choices, (1) $\alpha = 1, \beta = 1$ and (2) $\alpha = 0.5, \beta = 0.5$ since they show the best results in our experiments, while the others are collected in Appendix “Influence of Parameter (α, β) Brings in β -GAN.” For WGAN, the gradient penalty with parameter $\mu = 10$ is used to accelerate the speed of convergence, and hence the algorithm is denoted as WGANgp below. The trade-off (regularization) parameter of ℓ_1 or ℓ_2 reconstruction loss is set to be $\lambda = 10$ throughout this section, while an ablation study on varying λ is discussed in Appendix “Influence of the Regularization Parameter: λ .”

Results for RNAP

Denoising Without Contamination

In this part, we attempt to denoise the noisy image without the contamination (i.e., $\epsilon = 0$ in Eq. (8)). In order to present the advantage of GAN, we compare the denoising result in different methods. Table 1 shows the MSE and PSNR of different methods in SNR 0.05 and 0.1. We recognize the traditional methods such as KSVD, BM3D, nonlocal mean, and CWF can remove the noise partially and extract the general outline, but they still leave the unclear piece. However, deep learning methods can perform much better. Specifically, we observe that GAN-based methods, especially WGANgp + ℓ_1 loss and (.5, .5)-GAN + ℓ_1 loss, perform better than denoising autoencoder methods, which only optimizes ℓ_1 or ℓ_2 loss. The adversarial process inspires the generation process, and the additional ℓ_1 loss optimization speeds up the process of generation toward reference images. Notably, WGANgp and (5, .5)- or (1, 1)-GANs are among the best methods, where the best mean performances up to one standard deviation are all marked in bold font. Specifically, compared with (.5, .5)-GAN, the WGANgp get better PSNR and SSIM in SNR 0.1; the (.5, .5)-GAN shows the advantage in PSNR and SSIM in SNR 0.05, while (1, 1)-GAN is competitive within one standard deviation. Also, Fig. 5a presents the denoised images of denoising methods in SNR 0.05. For the convenience of comparison, we choose a clear open conformation (the rightmost

Table 1 Denoising result without contamination in simulated RNAP dataset: MSE, PSNR, and SSIM of different models, such as BM3D (Dabov et al. 2007), KSVD (Aharon et al. 2006), nonlocal means (Wei and Yin 2010), CWF (Bhamre et al. 2016), DA, and GAN-based methods

Method/SNR	MSE		PSNR		SSIM	
	0.1	0.05	0.1	0.05	0.1	0.05
BM3D	3.52e-2 (7.81e-3)	5.87e-2 (9.91e-3)	14.54(0.15)	12.13(0.14)	0.20(0.01)	0.08(0.01)
KSVD	1.84e-2(6.58e-3)	3.49e-2(7.62e-3)	17.57(0.16)	14.61(0.14)	0.33(0.01)	0.19(0.01)
Nonlocal means	5.02e-2(5.51e-3)	5.81e-2(8.94e-3)	13.04(0.50)	12.40(0.65)	0.18(0.01)	0.09(0.01)
CWF	2.53e-2(2.03e-3)	9.28e-3(8.81e-4)	16.06(0.33)	20.31(0.41)	0.25(0.01)	0.08(0.01)
ℓ_2 -Autoencoder ^a	3.13e-3(7.97e-5)	4.02e-3(1.48e-4)	25.10(0.11)	23.67(0.77)	0.79(0.02)	0.79(0.01)
ℓ_1 -Autoencoder ^b	3.16e-3(7.05e-5)	4.23e-3(1.32e-4)	25.05(0.09)	23.80(0.13)	0.77(0.02)	0.76(0.01)
(0, 0)-GAN + ℓ_1 ^c	3.06e-3(5.76e-5)	4.02e-3(5.67e-4)	25.25(0.04)	24.00(0.06)	0.78(0.03)	0.78(0.03)
WGAN _{gp} + ℓ_1	2.95e-3(1.41e-5)	4.00e-3(8.12e-5)	25.42(0.04)	24.06(0.05)	0.83(0.02)	0.80(0.03)
(1, 1)-GAN + ℓ_1	2.99e-3(3.51e-5)	4.01e-3(1.54e-4)	25.30(0.05)	24.07(0.16)	0.82(0.03)	0.79(0.03)
(.5, .5)-GAN+ ℓ_1	3.01e-3(2.81e-5)	3.98e-3(4.60e-5)	25.27(0.04)	24.07(0.05)	0.79(0.04)	0.80(0.03)

^a ℓ_2 -Autoencoder represents ℓ_2 loss

^b ℓ_1 -Autoencoder represents ℓ_1 loss

^c GAN + ℓ_1 represents adding ℓ_1 regularization in GAN generator loss

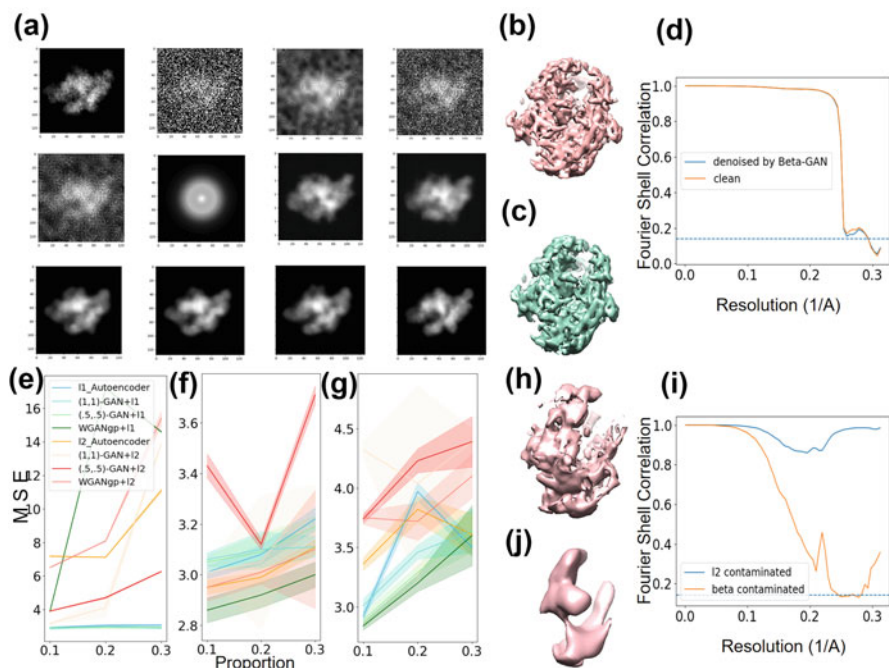


Fig. 5 Results for RNAP dataset. (a) is denoised images in different denoised methods (from left to right, top to bottom): clean, noisy, BM3D, KSVD, nonlocal means, CWF, ℓ_1 -autoencoder, ℓ_2 -autoencoder, (1,1)-GAN + ℓ_1 , (0,0)-GAN + ℓ_1 , (.5, .5)-GAN + ℓ_1 , and WGANgp + ℓ_1 . (b) and (c) are reconstruction of clean images and (.5, .5)-GAN + ℓ_1 denoised images. (d) is FSC curve of (b) and (c). (e), (f), and (g) are robustness tests of various methods under $\epsilon \in \{0.1, 0.2, 0.3\}$ -proportion contamination in three types of contamination: (e) type A, replacing the reference images with random noise; (f) type B, replacing the noisy images with random noise; (g) type C, replacing both with random noise. (h) and (j) are reconstructions of images with (.5, .5)-GAN + ℓ_1 and ℓ_2 -autoencoder under type A contamination, respectively, where ℓ_2 -autoencoder totally fails but (.5, .5)-GAN + ℓ_1 is robust. (i) shows FSC curves of (h) and (j)

conformation of Fig. 3) to present, and the performances show that WGANgp and (α , β)-GAN can grasp the “open” shape completely and derive the more explicit pictures than other methods.

What’s more, in order to test the denoised results of β -GAN, we reconstruct the 3D volume by RELION in 200,000 images of SNR 0.1, which are denoised by (.5, .5)-GAN + ℓ_1 . The value of pixel size, amplitude contrast, spherical aberration, and voltage are 1.6, 2.26, 0.1, and 300. For the other terms, retain the default settings in RELION software. Figure 5b and c separately shows the 3D volume recovered by clean images and denoised images. Also, the related FSC curves are shown in Fig. 5d. Specifically, the blue curve, which represents the denoised images in (.5, .5)-GAN + ℓ_1 , is closed to red curves representing the clean images. We use the 0.143 cutoff criterion in literature (the resolution as Fourier shell correlation reaches 0.143, shown by dash lines in Fig. 5d) to choose the final resolution: 3.39Å.

The structure recovered by (.5, .5)-GAN + ℓ_1 and FSC curve are as good as the original structure, which illustrates that the denoised result of β -GAN can identify the details of image and be helpful in 3D reconstruction.

In addition, Appendix “Clustering to Solve the Conformational Heterogeneity” also shows an example that GAN with ℓ_1 -autoencoder helps heterogeneous conformation clustering.

Robustness Under Contamination

We also consider the contamination model $\epsilon \neq 0$ and Q from purely noisy images. We randomly replace partial samples of our training dataset of RNAP by noise to test whether our model is robust or not. There are three types of contaminations to test: (A) only replacing the clean reference images (it implies the reference images are wrong or missing, such that we do not have the reference images to compare; this is the worst contamination case), (B) only replacing the noisy images (it means the cryo-EM images which the machine produces are broken), and (C) replacing both, which indicates both A and B happen. The latter two are mild contamination cases, especially C that replaces both reference and noisy images by Gaussian noise whose ℓ_1 or ℓ_2 loss is thus well-controlled.

Here we test our robustness of various deep learning-based methods using the RNAP data of SNR 0.1, and the former three types of contamination are applied to randomly replace the samples in the proportion of $\epsilon \in \{0.1, 0.2, 0.3\}$ of the whole dataset.

Figure 5e, f, and g compares the robustness of different methods. In all the cases, some β -GANs ((.5, .5)- and (1, 1)-) with ℓ_1 -autoencoder exhibit relatively universal robustness. Particularly, (1) the MSE with ℓ_1 loss is less than the MSE with ℓ_2 loss, which represents the ℓ_1 loss is more robust than ℓ_2 as desired. (2) The autoencoder method in ℓ_2 loss and WGANgp show certain robustness in cases B and C but are largely influenced by contamination in case A (shown in Fig. 5e), indicating the most serious damage arising from type A, merely replacing only the reference image by Gaussian noise. The reason is that the ℓ_2 -autoencoder and WGANgp method are confused by the wrong reference images so that they cannot learn the mapping from data distribution to reference distribution accurately. (3) In the type C, the standard deviations of the five best models are larger compared the other two types. The contamination of both noisy y and clean x images influence the stability of model more than the other two types.

Furthermore, we take an example in type A contamination with $\epsilon = 0.1$ for 3D reconstruction. The 3D reconstructions in denoised images with (.5, .5)-GAN + ℓ_1 and ℓ_2 -autoencoder are shown in Fig. 5h and j, and related FSC curve is Fig. 5i. Specifically, on the one hand, the blue FSC curve of ℓ_2 -autoencoder doesn't drop, which leads to the worse reconstruction; on the other hand, the red FSC curve of (.5, .5)-GAN + ℓ_1 drops quickly but begins to rise again, whose reason is that some unclear detail of structure mixed angular information in reconstruction. When applying 0.143 cutoff criterion (dashed line in FSC curve), the resolution of (.5, .5)-GAN + ℓ_1 is about 4Å. Although reconstruction of images and final resolution is not better than the clean images, it is much clearer than ℓ_2 -autoencoder which totally

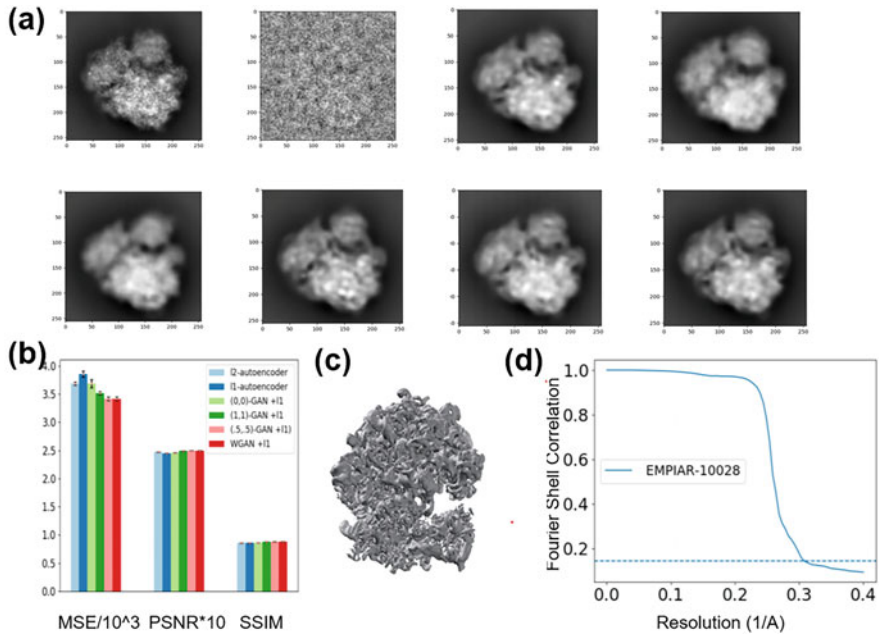


Fig. 6 Results for EMPIAR-10028. **(a)** Comparison in EMPIAR-10028 dataset in different deep learning methods (from left to right, top to bottom): clean image, noisy image, ℓ_1 -autoencoder, ℓ_2 -autoencoder, (0, 0)-GAN + ℓ_1 , (1, 1)-GAN + ℓ_1 , (.5, .5)-GAN + ℓ_1 , WGANgp + ℓ_1 . **(b)** is the MSE, PSNR, and SSIM in different denoised methods. **(c)** and **(d)** are the 3D reconstruction of denoised images by (.5, .5)-GAN + ℓ_1 and the FSC curve, respectively. The resolution of reconstruction from (.5, .5)-GAN + ℓ_1 denoised images is 3.20Å, which is as good as the original resolution

fails in the contamination case. The outcome of the reconstruction demonstrates that (.5, .5)-GAN + ℓ_1 is relatively robust, whose 3D result is consistent with the clean image reconstruction.

In summary, some (α, β) -GAN methods, such as the ((.5, .5)-GAN and (1, 1)-GAN, with ℓ_1 -autoencoder are more resistant to sample contamination, which are better to be applied into the denoising of cryo-EM data.

Results for EMPIAR-10028

The following Fig. 6a and b shows the denoising results by different deep learning methods in experimental data, ℓ_1 - or ℓ_2 -autoencoders, JS-GAN ((0, 0)-GAN), WGANgp, and (α, β) -GAN, where we add ℓ_1 loss in all of the GAN-based structures. Although the autoencoder can grasp the shape of macromolecules, it is a little blur in some parts. What is more, WGANgp and (.5, .5)-GAN perform better than other deep learning methods according to MSE and PSNR, which is largely

consistent with the result of the RNAP dataset. The improvements of such GANs over pure autoencoders lie in their ability of utilizing structural information among similar images to learn the data distribution better.

Finally, we implement reconstruction via RELION of 100,000 images, which are denoised by (.5, .5)-GAN + ℓ_1 . The parameters are the same as the ones set in the paper (Wong et al. 2014). The reconstruction results are shown in Fig. 6c. It is demonstrated that the final resolution is 3.20Å, which is derived by FSC curve in Fig. 6d using the same 0.143 cutoff (dashed line) to choose the final resolution. We note that the final resolution by RELION after denoising is as good as the original resolution 3.20Å reported in Wong et al. (2014).

Conclusion

In this chapter, we set a connection between the traditional image forward model and Huber contamination model in solving the complex contamination in the cryo-EM dataset. The joint training of autoencoder and GAN has been proved to substantially improve the performance in cryo-EM image denoising. In this joint training scheme, the reconstruction loss of autoencoder helps GAN to avoid mode collapse and stabilize training. GAN further helps autoencoder in denoising by utilizing the highly correlated cryo-EM images since they are 2D projections of one or a few 3D molecular conformations. To overcome the low signal-to-noise ratio challenge in cryo-EM images, joint training of ℓ_1 -autoencoder combined with (.5, .5)-GAN, (1, 1)-GAN, and WGAN with gradient penalty is often among the best performances in terms of MSE, PSNR, and SSIM when the data is contamination-free. However, when a portion of data is contaminated, especially when the reference data is contaminated, WGAN with ℓ_1 -autoencoder may suffer from the significant deterioration of reconstruction accuracy. Therefore, robust ℓ_1 -autoencoder combined with robust GANs ((.5, .5)-GAN and (1, 1)-GAN) is the overall best choice for robust denoising with contaminated and high-noise datasets.

Part of the results in this chapter is based on a technical report (Gu et al. 2020). Most of the deep learning-based techniques in image denoising need reference data, limiting themselves in the application of cryo-EM denoising. For example, in our experimental dataset EMPIAR-10028, the reference data is generated by the cryoSPARC, which itself becomes problematic in highly heterogeneous conformations. Therefore, the reference image we learn may follow a fake distribution. How to denoise without the reference image thus becomes a significant problem. It is still open how to adapt to different experiments and those without reference images. In order to overcome this drawback, an idea called “image-blind denoising” was offered by the literature (Lehtinen et al. 2018; Krull et al. 2019), which viewed the noisy image or void image as the reference image to denoise. Besides, Chen et al. (2018) tried to extract the noise distribution from the noisy image and gain denoised images through removing the noise for noisy data; Quan et al. (2020) augmented the data by Bernoulli sampling and denoise image with dropout. Nevertheless, all of the methods need noise is independent of the elements themselves. Thus it is hard

to remove noise in cryo-EM because the noise from ice and machine is related to the particles.

In addition, for reconstruction problems in cryo-EM, Zhong et al. (2020) proposed an end-to-end 3D reconstruction approach based on the network from cryo-EM images, where they attempt to borrow the variational autoencoder (VAE) to approximate the forward reconstruction model and recover the 3D structure directly by combining the angle information and image information learned from data. This is one future direction to pursue.

Appendix

Influence of Parameter(α , β) Brings in β -GAN

In this part, we have applied β -GAN into denoising problem. How to pick up a good parameter: (α , β) in the β -GAN becomes an important issue. Therefore, we investigate the impact of the parameter (α , β) on the outcome of denoising. We choose eight significant groups of α , β . Our result is shown in Table 2. It is demonstrated that the effect of these groups in different parameters is not large. The best result appears in $\alpha = 1$, $\beta = 1$ and $\alpha = 0.5$, $\beta = 0.5$

Clustering to Solve the Conformational Heterogeneity

In this part, we try to analyze whether the denoised result is good in solving conformation heterogeneity in simulated RNAP dataset. Specifically, for heterogeneous conformations in simulation data, we mainly choose the following two typical conformations: *open* and *close* conformations (the leftmost and rightmost conformations in Fig. 3) as our testing data. Our goal is to distinguish these two classes of conformations. However, different from the paper (Xian et al. 2018), we do not have the template images to calculate the distance matrix, so what we try is unsupervised learning – clustering. Our clustering method is firstly using manifold learning, Isomap (Tenenbaum et al. 2000), to reduce the dimension of the denoised images and then making use of k -means ($k = 2$) to group the different conformations.

Figure 7a displays the 2D visualizations of two conformations about the clustering effect in different denoised methods. Here the SNR of noisy data is 0.05. In correspondence to those visualizations, the accuracy of competitive methods is reported: (1, 1)-GAN+ ℓ_1 , 54/60 (54 clustering correctly in 60); WGANgp+ ℓ_1 , 54/60; ℓ_2 -autoencoder, 44/60; BM3D, 34/60; and KSVD, 36/60. This result shows that clean images separate well; (α , β)-GAN and WGANgp with ℓ_1 -autoencoder can distinguish the open and close structure partially, although there exist several wrong points; ℓ_2 -autoencoder and traditional techniques have poor performance because it is hard to detect the clamp shape.

Table 2 The result of β -GANs with ResNet architecture: MSE, PSNR, and SSIM of different (α, β) in β -GAN under various levels of Gaussian noise corruption in RNAP dataset

Parameter/SNR	MSE		PSNR		SSIM	
	0.1	0.05	0.1	0.05	0.1	0.05
$\alpha = 1, \beta = 1$	2.99e-3(3.51e-5)	4.01e-3(1.54e-4)	25.30(0.05)	24.07(0.16)	0.82(0.03)	0.79(0.03)
$\alpha = 0.5, \beta = 0.5$	3.01e-3(2.81e-5)	3.98e-3(4.60e-5)	25.27(0.04)	24.07(0.05)	0.79(0.04)	0.80(0.03)
$\alpha = -0.5, \beta = -0.5$	3.02e-3(1.69e-5)	4.15e-3(5.05e-5)	25.27(0.02)	23.91(0.05)	0.80(0.03)	0.80(0.03)
$\alpha = -1, \beta = -1$	3.05e-3(3.54e-5)	4.12e-3(8.30e-5)	25.23(0.05)	23.93(0.08)	0.80(0.05)	0.77(0.04)
$\alpha = 1, \beta = -1$	3.05e-3(4.30e-5)	4.10e-3(5.80e-5)	25.24(0.06)	23.96(0.06)	0.82(0.02)	0.76(0.03)
$\alpha = 0.5, \beta = -0.5$	3.09e-3(6.79e-5)	4.05e-3(6.10e-5)	25.17(0.04)	24.01(0.06)	0.79(0.04)	0.77(0.05)
$\alpha = 0, \beta = 0$	3.06e-3(5.76e-5)	4.02e-3(5.67e-4)	25.23(0.04)	24.00(0.06)	0.78(0.03)	0.78(0.03)
$\alpha = 0.1, \beta = -0.1$	3.07e-3(5.62e-5)	4.05e-3(8.55e-5)	25.23(0.08)	23.98(0.04)	0.78(0.02)	0.79(0.03)

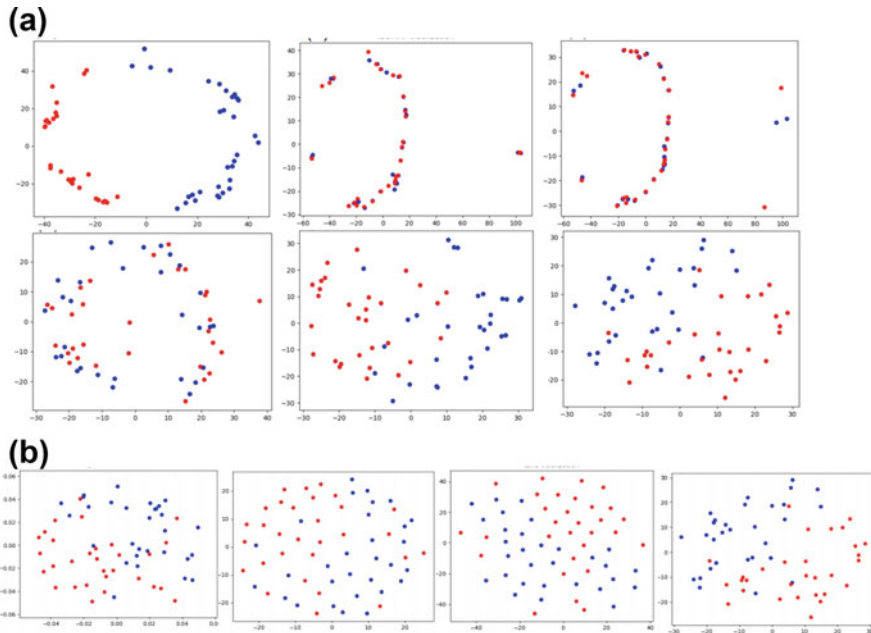


Fig. 7 2D visualization of two-conformation images in manifold learning. Red point and blue point separately represent the open and closed conformation. (a) is 2D visualization of two-conformation image by ISOMAP in different methods (from the left and top to the right and bottom): clean image, BM3D, KSVD, ℓ_2 -autoencoder, (1, 1)-GAN+ ℓ_1 , WGANgp+ ℓ_1 . (b) is 2D visualization of two-conformation image in different manifold learning methods (from left to right): spectral methods, MDS, TSNE, and ISOMAP

Furthermore, the reason we use Isomap is it performs the best in our case, and comparisons of different manifold learning methods are shown in Fig. 7b. It demonstrates that blue and red points separate most in the graph of ISOMAP. Specifically, the accuracy of these four methods are 50/60 (spectral method), 46/50 (MDS), 46/50 (TSNE), and 54/60 (ISOMAP). It is shown that Isomap can distinguish best in the two structures' images compared to other methods, such as the spectral method (Ng et al. 2002), MDS (Cox and Cox 2008), and TSNE (Maaten and Hinton 2008).

Convolution Network

We present the result of simple deep convolution network (remove the ResNet block); the performances in all of criterion are worse than performances of the residue's architecture work. Table 3 compares the MSE and PSNR performance of various methods in the RNAP dataset with SNR 0.1 and 0.05. And Fig. 8a displays the denoised image of different methods in the RNAP dataset with SNR 0.05.

Table 3 MSE and PSNR of different models under various levels of Gaussian noise corruption in RNAP dataset, where the architectures of GANs or autoencoders are simply convolution network

Method/SNR	MSE		PSNR	
	0.1	0.05	0.1	0.05
BM3D	3.5e-2(7.8e-3)	5.9e-2(9.9e-3)	14.535(0.1452)	12.134(0.1369)
KSVD	1.8e-2(6.6e-3)	3.5e-2(7.6e-3)	17.570(0.1578)	14.609(0.1414)
Nonlocal means	5.0e-2(5.5e-3)	5.8e-2(8.9e-3)	13.040(0.4935)	12.404(0.6498)
CWF	2.5e-2(2.0e-3)	9.3e-3(8.8e-4)	16.059(0.3253)	20.314(0.4129)
ℓ_2 -Autoencoder	4.0e-3(6.0e-4)	6.7e-3(9.0e-4)	24.202(0.6414)	21.739(0.7219)
(0, 0)-GAN + ℓ_1	3.8e-3(6.0e-4)	5.6e-3(8.0e-4)	24.265(0.6537)	22.594(0.6314)
WGAN _{gp} + ℓ_1	3.1e-3(5.0e-4)	5.0e-3(8.0e-4)	25.086(0.6458)	23.010(0.6977)
(1, -1)-GAN + ℓ_1	3.4e-3(5.0e-4)	4.9e-3(9.0e-4)	24.748(0.7233)	23.116(0.7399)
(.5, -.5)-GAN + ℓ_1	3.5e-3(5.0e-4)	5.6e-3(9.0e-4)	24.556(0.6272)	22.575(0.6441)

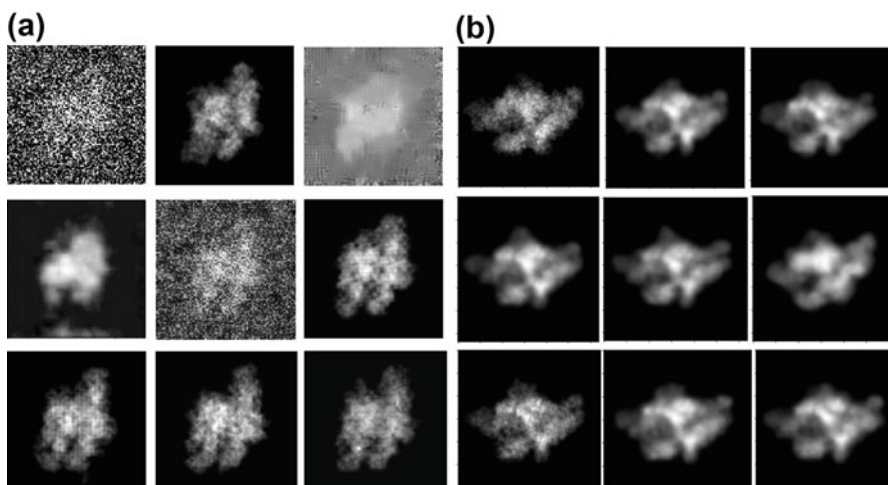


Fig. 8 (a) Denoised images with convolution network without ResNet structure in different methods in RNAP dataset with SNR 0.05 (from left to right, top to bottom): clean, noisy, BM3D, ℓ_2 -autoencoder, KSVD, JS-GAN + ℓ_1 , WGAN_{gp} + ℓ_1 , (1, -1)-GAN + ℓ_1 , (.5, -.5)-GAN + ℓ_1 . (b) Denoised and reference images in different regularization λ (we use (.5, .5)-GAN + ℓ_1 as an example) in corresponding to Table 4. From left to right, top to bottom, the image is clean image, $\lambda = 0.1, \lambda = 1, \lambda = 5, \lambda = 10, \lambda = 50, \lambda = 100, \lambda = 500, \lambda = 10,000$

It shows the advantage of residue structure in our GAN-based denoising cryo-EM problem.

Test RNAP Dataset with PGGAN Strategy

PGGAN (Karras et al. 2018) is a popular method to generate high-resolution images from low-resolution ones by gradually adding layers of generator and discriminator.

It accelerates and stabilizes the model training. Since cryo-EM images are in large pixel size that fits well the PGGAN method, here we choose its structure² instead of the ResNet and convolution structures above to denoise cryo-EM images. Our experiments partially demonstrate two things: (1) the denoised images sharpen more, though the MSE changes to be higher; (2) we do not need to add ℓ_1 regularization to make model training stable; it can also detect the outlier of images for both real and simulated data without regularization.

In detail, based on the PGGAN architecture and parameters, we test the following two objective functions developed in the section “Robust Denoising Method”: WGANgp and WGANgp + ℓ_1 , in the RNAP simulated dataset with SNR 0.05 as an example to explain. The denoised images are presented in Fig. 9; it is noted that the model is hard to collapse regardless of adding ℓ_1 regularization. The MSE of adding regularization is 8.09e-3(1.46e-3), which is less than 1.01e-2(1.81e-3)

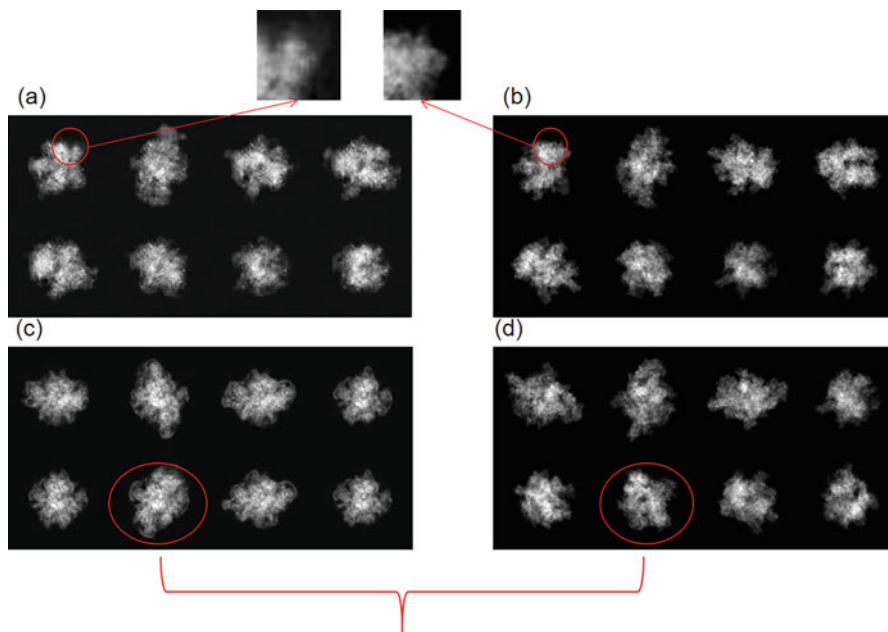


Fig. 9 Denoised and reference images by PGGAN instead of simple ResNet and convolution structure in RNAP dataset with SNR 0.05. The PGGAN strategy is tested in two objective functions: WGANgp + ℓ_1 and WGANgp. (a) and (b) are denoised and reference images using PGGAN with WGANgp + ℓ_1 ; (c) and (d) are denoised and reference images using PGGAN in WGANgp, respectively. Specifically, the images highlighted in red color show the structural difference between denoised images and reference images. It demonstrates that denoised images are different from reference images using PGGAN strategy

²We set the same architecture and parameters as <https://github.com/nashory/pggan-pytorch> and the input image size is 128×128 .

Table 4 MSE, PSNR, and SSIM of different λ in (.5,.5)-GAN + λl_1 in RNAP dataset

λ /criterion	MSE	PSNR	SSIM
0.1	3.06e-3(4.50e-5)	25.22(0.07)	0.82(0.06)
1	3.05e-3(4.49e-5)	25.24(0.06)	0.81(0.05)
5	3.03e-3(2.80e-5)	25.26(0.04)	0.80(0.04)
10	3.01e-3(2.81e-5)	25.27(0.04)	0.79(0.04)
50	3.07e-3(3.95e-5)	25.20(0.06)	0.79(0.02)
100	3.11e-3(5.96e-5)	25.15(0.06)	0.80(0.02)
500	3.17e-3(5.83e-5)	25.01(0.07)	0.78(0.04)
10,000	3.17e-3(2.90e-5)	25.03(0.04)	0.79(0.04)

without adding regularization. Nevertheless, both of them don't exceed the results based on the ResNet structure above. This shows that PGGAN architecture does not have more power than the ResNet structure. But an advantage of PGGAN lies in its efficiency in training. So it is an interesting problem to improve PGGAN toward the accuracy of ResNet structure.

Another thing that needs to be highlighted is MSE may not be a good criterion because denoised images by PGGAN are clearer in some details than the front methods we propose. This phenomenon is also shown in Appendix “[Influence of the Regularization Parameter: \$\lambda\$.](#)” So how to find a better criterion to evaluate the model and combine two strengths of ResNet-GAN and PGGAN await us to explore.

Influence of the Regularization Parameter: λ

In this chapter, we add ℓ_1 regularization to make model stable, but how to choose λ of ℓ_1 regularization becomes a significant problem. Here we take (.5, .5)-GAN to denoise in RNAP dataset with SNR 0.1. According to some results in different λ in Table 4, we find as the λ tends to infinity, the MSE results tend to ℓ_1 -autoencoder, which is reasonable. Also, the MSE result becomes the smallest as the $\lambda = 10$.

What's more, an interesting phenomenon is found that a much clearer result could be obtained at $\lambda = 100$ than that at $\lambda = 10$, although the MSE is not the best (shown in Fig. 8b).

References

- Agostinelli, F., Anderson, M., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: Advances in Neural Information Processing Systems, pp. 1493–1501 (2013)
- Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Sig. Process. **54**(11), 4311–4322 (2006)
- Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceeding of the International Conference on Machine Learning, pp. 214–223 (2017)
- Bae, B., Feklistov, A., Lass-Napiorkowska, A., Landick, R., Darst, S.: Structure of a bacterial RNA polymerase holoenzyme open promoter complex. Elife **4**, e08504 (2015)

- Bai, X.C., McMullan, G., Scheres, S.: How Cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* **40**(1), 49–57 (2015)
- Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings*, pp. 37–49 (2012)
- Bau, D., Zhu, J.Y., Wulff, J., Peebles, W., Strobel, H., Zhou, B., Torralba, A.: Seeing what a gan cannot generate. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4502–4511 (2019)
- Bhamre, T., Zhang, T., Singer, A.: Denoising and covariance estimation of single particle Cryo-EM images. *J. Struct. Biol.* **195**(1), 72–81 (2016)
- Browning, D., Busby, S.: The regulation of bacterial transcription initiation. *Nat. Rev. Microbiol.* **2**(1), 57–65 (2004)
- Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3155–3164 (2018)
- Chen, J., Chiu, C., Gopalkrishnan, S., Chen, A., Olinares, P., Saecker, R., Winkelman, J., Maloney, M., Chait, B., Ross, W. et al.: Stepwise promoter melting by bacterial RNA polymerase. *Mol. Cell* **78**, 275–288.e6 (2020)
- Cox, M., Cox, T.: Multidimensional scaling. In: *Handbook of Data Visualization*. Springer, Berlin, pp. 315–347 (2008)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Dai, Z., Yang, Z., Yang, F., Cohen, W.W., Salakhutdinov, R.: Good semi-supervised learning that requires a bad gan. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6513–6523 (2017)
- Dong, Z., Liu, G., Ni, G., Jerwick, J., Duan, L., Zhou, C.: Optical coherence tomography image denoising using a generative adversarial network with speckle modulation. *J. Biophotonics* **13**(4), e201960135 (2020)
- Frank, J.: *Three-dimensional electron microscopy of macromolecular assemblies: visualization of biological molecules in their native state*. Oxford University Press, New York (2006)
- Gao, C., Liu, J., Yao, Y., Zhu, W.: Robust estimation and generative adversarial nets. In: *International Conference on Learning Representation*, New Orleans (2019)
- Gao, C., Yao, Y., Zhu, W.: Generative adversarial nets for robust scatter estimation: a proper scoring rule perspective. *J. Mach. Learn. Res.* **21**, 160–161 (2020)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
- Gu, H., Unarta, I.C., Huang, X., Yao, Y.: Robust autoencoder gan for cryo-em image denoising (2020)
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of Wasserstein GANs. In: *Advances in Neural Information Processing Systems*, pp. 5767–5777 (2017)
- Hua, Y., Li, R., Zhao, Z., Chen, X., Zhang, H.: Gan-powered deep distributional reinforcement learning for resource management in network slicing. *IEEE J. Sel. Areas Commun.* **38**(2), 334–349 (2019)
- Huber, P.: Robust estimation of a location parameter. In: *Breakthroughs in Statistics*. Springer, New York, pp. 492–518 (1992)
- Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: *International Conference on Learning Representation*, Vancouver (2018)
- Kenzaki, H., Koga, N., Hori, N., Kanada, R., Li, W., Okazaki, K., Yao, X.Q., Takada, S.: CafeMol: a coarse-grained biomolecular simulator for simulating proteins at work. *J. Chem. Theory Comput.* **7**(6), 1979–1989 (2011)
- Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: *International Conference on Learning Representation*, San Diego (2015)

- Krivov, G., Shapovalov, M., Dunbrack R.L. Jr.: Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Struct. Funct. Bioinform.* **77**(4), 778–795 (2009)
- Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137 (2019)
- Kühlbrandt, W.: The resolution revolution. *Science* **343**(6178), 1443–1444 (2014)
- Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: learning image restoration without clean data. In: *Proceeding of the International Conference on Machine Learning*, pp. 2965–2974 (2018)
- Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
- Marabini, R., Masegosa, I., San Martín, M., Marco, S., Fernandez, J., De la Fraga, L., Vaquerizo, C., Carazo, J.: Xmipp: an image processing package for electron microscopy. *J. Struct. Biol.* **116**(1), 237–240 (1996)
- Moore, B., Kelley, L., Barber, J., Murray, J., MacDonald, J.: High-quality protein backbone reconstruction from alpha carbons using Gaussian mixture models. *J. Comput. Chem.* **34**(22), 1881–1889 (2013)
- Murakami, K., Masuda, S., Darst, S.: Structural basis of transcription initiation: Rna polymerase holoenzyme at 4 Å resolution. *Science* **296**(5571), 1280–1284 (2002)
- Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: *Advances in Neural Information Processing Systems*, pp. 849–856 (2002)
- Okazaki, K., Koga, N., Takada, S., Onuchic, J., Wolynes, P.: Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: structure-based molecular dynamics simulations. *Proc. Natl. Acad. Sci.* **103**(32), 11844–11849 (2006)
- Punjani, A., Rubinstein, J.L., Fleet, D.J., Brubaker, M.A.: CryoSPARC: algorithms for rapid unsupervised Cryo-EM structure determination. *Nat. Methods* **14**(3), 290 (2017)
- Quan, Y., Chen, M., Pang, T., Ji, H.: Self2self with dropout: Learning self-supervised denoising from single image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1890–1898 (2020)
- Sarmad, M., Lee, H.J., Kim, Y.M.: RL-gan-net: a reinforcement learning agent controlled gan network for real-time point cloud shape completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5898–5907 (2019)
- Scheres, S.: Processing of structurally heterogeneous Cryo-EM data in RELION. In: *Methods in Enzymology*. Elsevier, Academic Press, vol. 579, pp. 125–157 (2016)
- Shen, P.: The 2017 Nobel Prize in Chemistry: Cryo-EM comes of age. *Anal. Bioanal. Chem.* **410**(8), 2053–2057 (2018)
- Su, M., Zhang, H., Schawinski, K., Zhang, C., Cianfrocco, M.: Generative adversarial networks as a tool to recover structural information from cryo-electron microscopy data. *BioRxiv*, p. 256792 (2018)
- Tenenbaum, J., De Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
- Tran, L., Nguyen, S.M., Arai, M.: GAN-based noise model for denoising real images. In: *Proceedings of the Asian Conference on Computer Vision* (2020)
- Tripathi, S., Lipton, Z.C., Nguyen, T.Q.: Correction by projection: denoising images with generative adversarial networks (2018)
- Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: *Proceeding of the International Conference on Machine Learning*, pp. 1096–1103 (2008)
- Wang, J., Yin, C.C (2013) A zernike-moment-based non-local denoising filter for cryo-em images. *Sci. China Life Sci.* **56**(4), 384–390
- Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
- Wang, F., Gong, H., Liu, G., Li, M., Yan, C., Xia, T., Li, X., Zeng, J.: DeepPicker: a deep learning approach for fully automated particle picking in Cryo-EM. *J. Struct. Biol.* **195**(3), 325–336 (2016)

- Warren, B.E.: X-Ray Diffraction. Courier Corporation. Dover Publications; Reprint Edition (1990)
- Wei, D.Y., Yin, C.C.: An optimized locally adaptive non-local means denoising filter for cryo-electron microscopy data. *J. Struct. Biol.* **172**(3), 211–218 (2010)
- Wong, W., Bai, X.C., Brown, A., Fernandez, I., Hanssen, E., Condrón, M., Tan, Y.H., Baum, J., Scheres, S.: Cryo-EM structure of the Plasmodium falciparum 80s ribosome bound to the anti-protozoan drug emetine. *Elife* **3**, e03080 (2014)
- Wüthrich, K.: NMR with proteins and nucleic acids. *Europhys. News* **17**(1), 11–13 (1986)
- Xian, Y., Gu, H., Wang, W., Huang, X., Yao, Y., Wang, Y., Cai, J.F.: Data-driven tight frame for cryo-em image denoising and conformational classification. In: *Proceeding of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 544–548 (2018)
- Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: *Advances in Neural Information Processing Systems*, pp. 341–349 (2012)
- Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M., Zhang, Y., Sun, L., Wang, G.: Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE Trans. Med. Imaging* **37**(6), 1348–1357 (2018)
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
- Zhong, E., Bepler, T., Davis, J., Berger, B.: Reconstructing continuous distributions of 3D protein structure from Cryo-EM images. In: *International Conference on Learning Representation*, Addis Ababa (2020)



Variational Models and Their Combinations with Deep Learning in Medical Image Segmentation: A Survey

27

Luying Gui, Jun Ma, and Xiaoping Yang

Contents

Introduction	1002
Conventional Algorithms Based on Variational Methods	1003
The Data Term	1004
The Regularization Term	1006
Variational Models Meet Deep Learning in Medical Image Segmentation	1011
Variational Models Guided Deep Learning	1011
Deep Learning-Driven Variational Models	1015
Conclusion	1017
References	1017

Abstract

Image segmentation means to partition an image into separate meaningful regions. Segmentation in medical images can extract different organs, lesions, and other regions of interest, which helps in subsequent disease diagnosis, surgery planning, and efficacy assessment. However, medical images have many unavoidable interference factors, such as imaging noise, artificial artifacts, and mutual occlusion of organs, which make accurate segmentation highly difficult. Incorporating prior knowledge and image information into segmentation model based on variational methods has proven efficient for more accurate segmentation. In recent years, segmentation based on deep learning has been significantly developed, and the combination of classical variational method-

L. Gui · J. Ma

Department of Mathematics, Nanjing University of Science and Technology, Nanjing, China
e-mail: ly.gui@njust.edu.cn; junma@njust.edu.cn

X. Yang (✉)

Department of Mathematics, Nanjing University, Nanjing, China
e-mail: xpyang@nju.edu.cn

based models with deep learning is a hot topic. In this survey, we briefly review the segmentation methods based on a variational method making use of image information and regularity information. Subsequently, we clarify how the integration of variational methods into the deep learning framework leads to more precise segmentation results.

Keywords

Medical image segmentation · Variational models · Deep learning

Introduction

Medical image segmentation plays an important role in clinical practices, such as quantitative analysis of lesions, radiotherapy planning, pre-operative planning, intra-operative navigation, and post-operative evaluation. A large number of segmentation methods have been proposed in the past few decades such as graph cut-based (Boykov et al. 2001; Boykov and Funka-Lea 2006), atlas-based methods (Iglesias and Sabuncu 2015), etc. Variational model-based methods are one of the most widely used approaches in medical image segmentation.

The key idea behind the variational model is to make the contour reach the object boundary and minimize the energy functional, which is usually related to information such as intensity, gradient, and texture of the image itself, and also usually includes the desired properties in order to achieve a better segmentation result.

Variational model-based segmentation methods have many desired features, for example, they have transparent and explainable mathematical formulations. Customized constraints and priors can be easily and naturally incorporated into the energy functionals. Moreover, they do not rely on large training data. However, variational models still in general suffer from several shortages:

- The final segmentation results rely on good and reasonable initializations
- The hyperparameters need to be tuned for each testing case
- They lack the ability to learn efficient representations from labeled data

During the past 5 years, fully supervised deep learning methods have revolutionized medical image segmentation (Litjens et al. 2017), and many convolutional neural networks (CNNs) (Long et al. 2015; Shelhamer et al. 2017; Ronneberger et al. 2015; Isensee et al. 2021) have achieved unprecedented performance, such as liver segmentation (Bilic et al. 2019; Kavur et al. 2021), cardiac segmentation (Bernard et al. 2018), kidney segmentation (Heller et al. 2020), and so on. CNN-based segmentation methods directly build the end-to-end mapping between images and annotations by automatically learning object feature representations from a number of training data. The learned models can be directly applied to testing images without any hyperparameter tuning. However, these methods lack

interpretability and rely on the large training sets. In this paper, we mainly focus on fully supervised deep learning methods, while there are also weakly supervised methods (Cheplygina et al. 2019) for medical image segmentation.

Thanks to the complementary roles between classical variational models and modern deep learning approaches, a natural trend is to combine the advantages of the two types of approaches to design more accurate, data-efficient, and transparent segmentation methods.

This paper aims to present an overview of classical variational models and their extensions in deep learning era, especially in medical image segmentation. The remainder of this article is organized as follows. First, we introduce the conventional variational models with typical data terms and regularization terms. Then, we present the different combination mechanisms between variational models and deep learning: variational model-guided deep learning and deep learning-driven variational models. Finally, we draw a brief conclusion.

Conventional Algorithms Based on Variational Methods

In 1989 Mumford and Shah proposed a famous image segmentation model, named Mumford-Shah (MS) model (Mumford and Shah 1989), which assumes the image I as a piece-wise smooth function u with the following energy functional:

$$E_{MS}(C, u) = \underbrace{\int_{\Omega} |I - u|^2 dx}_{E_{\text{fidelity}}} + \underbrace{v \int_{\Omega \setminus C} |\nabla u|^2 dx + \gamma \mathcal{H}^1(C)}_{E_{\text{regularization}}}, \tag{1}$$

where C is a closed subset of image domain Ω and represents the boundary of the object, and \mathcal{H}^1 is the one-dimensional Hausdorff measure.

The solution of the functional (1) is formed by smooth regions R_i , which is represented by u and with sharp boundaries C . A reduced form of this problem is to simplify the restriction of E_{MS} to piecewise constant functions u , that is, $u = c_i$ on each R_i . The reduced case is proposed by Chan and Vese (2001); the energy functional of Chan-Vese (CV) model is as follows:

$$E_{CV}(C, c_1, c_2) = \underbrace{\lambda_1 \int_{\text{inside}(C)} |I - c_1|^2 dx + \lambda_2 \int_{\text{outside}(C)} |I - c_2|^2 dx}_{E_{\text{fidelity}}} + \underbrace{\mu |C|}_{E_{\text{regularization}}}, \tag{2}$$

where c_1 and c_2 are two constants, respectively.

The MS and CV models are based on the assumptions of the segmented regions. Differing from the above models, the “snakes” model focus on boundary detection,

and these kinds of models have been extensively studied since the original work of Kass et al. (1988). The main idea is based on deforming the initial contour so that it is oriented towards the boundary of the object to be detected. The classical snakes model relates the parametrized planar curve $C(q) : [0, 1] \rightarrow \mathbb{R}^2$ to an energy which is given by

$$E_{\text{snakes}}(C) = \underbrace{-\int_0^1 |\nabla I(C(q))|^2 dq}_{E_{\text{fidelity}}} + \underbrace{\int_0^1 \alpha |C'(q)|^2 + \beta |C''(q)|^2 dq}_{E_{\text{regularization}}}. \quad (3)$$

The above three most typical methods are based on regional information and boundary information, respectively. Many researchers also classify the variational model-based segmentation methods into two categories: region information-based and boundary information-based. This classification method is mainly according to the usage of different types of information in data terms. However, in the actual segmentation of medical images, there are inevitably disturbing factors such as imaging noise, artifacts, and occlusions, which can easily mislead the segmentation algorithm and lead to imprecise segmentation results. In this case, it has become a current inevitable trend to impose proper features or constraints on the segmentation models. The energy term to achieve this function is called the regularization term. The functional of the above three classical methods also consists of two types of energy, the fidelity term and the regularization term, as labeled in these energy functionals. One is the term driven by image information, which guarantees the correspondence between segmentation results and image data and is called the fidelity term. The other guarantees specific properties of the contour or region. This category is called the regularization term.

The Data Term

In image segmentation, the fidelity term is also called the data term for two main reasons. First, the energy of this term usually originates from the image itself, such as E_{fidelity} in the snakes model, which utilizes the gradient information of the image, and E_{fidelity} in the CV model, which utilizes the mean values of the intensity of the different regions of the image. In addition, segmentation models also usually make assumptions about the image, such as the MS model, in which a piecewise smooth function u is used to approximate the image. The fidelity term $\int_{\Omega} |I - u|^2 dx$ ensures that the function u does not deviate too far from the actual image I . According to the different types of image information utilized by the fidelity, we classify them into two categories, boundary information-based, and regional information-based.

The Boundary Information

Boundary and edge information usually includes important image features that are often used to delineate the object of interest in an image. In image segmentation, the

actual boundary of the object is usually considered to be where the pixel changes most dramatically, so the boundary information can be obtained by applying edge detectors, which typically involve first- or second-order spatial differential operators.

One of the most popular segmentation models using edge information is the GAC model (Caselles et al. 1997), which uses image gradient to construct a monotonically decreasing function as a stopping function to control the contour evolution. Since the object boundary is usually expressed as the maximum gradient in the image, this method enables the contour to stop at the desired object boundary.

Since segmentation algorithms aim to find the boundaries of the objects, the detection of boundaries and boundary-based segmentation algorithms is a very intuitive idea and has very accurate segmentation results on better-quality images. However, since interferences such as noise and pseudo-boundaries are often present on medical images and segmentation targets often show weak or missing boundaries, in these cases, boundary-dependent algorithms are often fragile. Therefore, some researchers have also emphasized the importance of integrating regional information for accurate segmentation (Haddon and Boyce 1990; Falah et al. 1994; Chan et al. 1996; Muñoz et al. 2003).

The Regional Information

Although boundaries of the objects provide a natural data-fitting target, it is commonly believed that region-based formulations exhibit less local minima than approaches that solely rely on gradient information of the objects (Cremers et al. 2007). In region-based methods, the intensity/gray value of the image is usually used, such as in CV model, where the gray values of the target and background are assumed to be close to two different constants, respectively. Under this circumstance, the intensity on each pixel is considered to be spatially independent. However, for textured images, the gray value of a pixel is considered to be correlated with its surroundings. The texture is a special attribute of an image for which there is no formal scientific definition (Tuceryan and Jain 1998), and local correlations of intensities usually characterize the textures. Although texture can be visually recognized, it is difficult to define one mathematically, so it is difficult to segment images with texture by general methods. The texture features have been proposed to capture these local correlations. Common representations of texture properties are gray-level co-occurrence matrices (Reska et al. 2015; Wu et al. 2015; Haddon and Boyce 1990; Boonnuk et al. 2015; Lu et al. 2017; Pons et al. 2008), Gabor filters (Gui et al. 2017c), local binary patterns (LBP) (Gui and Yang 2018), sparse texture dictionaries, variational image decompositions, and rapidly developing deep learning based on convolutional neural networks(CNN) in recent years. In addition, since the Gaussian mixture model (GMM) is theoretically capable of fitting any distribution of pixels, it is also commonly used as a regional term in the segmentation of medical images (Martinez-Uso et al. 2010; Balafar 2014; Ji et al. 2012).

However, in medical images, irregular intensity distributions are often presented. Intensity inhomogeneity on medical images is a common phenomenon that can be

caused by many factors, such as complex noise, unavoidable artifacts produced by the imaging equipment, and the nature of the imaging object itself. For different situations, researchers have proposed various schemes to solve these problems. For example, to suppress the effects of noise, researchers used a combination of a local denoising term and a local fidelity term to ensure segmentation accuracy (Ali et al. 2018); in Niu et al. (2017), researchers used a local similarity factor to resist the influence of noise. To fitting the unevenly distributed intensity, researchers Yu et al. (2019) generated an adaptive perturbation factor to integrate the external energy functional of the curve evolution. In Li et al. (2008), researchers investigated two fitting functions that locally approximate the image intensities on the two sides of the contour, respectively. In addition, different methods are proposed to correct the bias due to uneven illumination and imaging artifacts (Zhou et al. 2017; Li et al. 2009). Some researchers have also proposed a quantitative assessment of the degree of inhomogeneity of the regions themselves, so as to find the boundaries of different regions and thus segment the desired objects (Li et al. 2011, 2020b; Gui et al. 2017a).

In practical medical image segmentation, the boundary and region information are usually used in combination to achieve a better discriminative object description. For example, the corner detection detects the boundary points, but the critical points inside the object are extracted. On the other hand, the active shape model (Cootes et al. 2000) creates a position-dependent statistical model of all critical points at the object boundary and interior (Cootes et al. 1994). Based on this, statistical texture information is incorporated to form an active appearance model (Cootes et al. 2001; Beichel et al. 2005). Figure 1 shows the different segmentation results given by the two methods, which use intensity information (Chan and Vese 2001) of the image and intensity combined with texture information (Gui et al. 2017c), respectively. The differences between the two segmentation results can be observed by zooming in on the region.

The Regularization Term

In segmentation, the regularization term, also known as constraint, keeps the model from overfitting or imposing some restrictions so that the segmentation curve or segmented region has specific desired properties. Based on the purpose of these regularization terms, they can be divided into two categories: generic regularization terms, which are not related to the segment objects, and specific regularization terms, which are related to the segment objects. Furthermore, they constrain and guide the segmentation model according to some characteristics of the objects.

Generic Regularization

The constraints imposed on the curve are usually independent of the specific segmentation target, by which the smoothness or other characteristics of the curve are guaranteed.

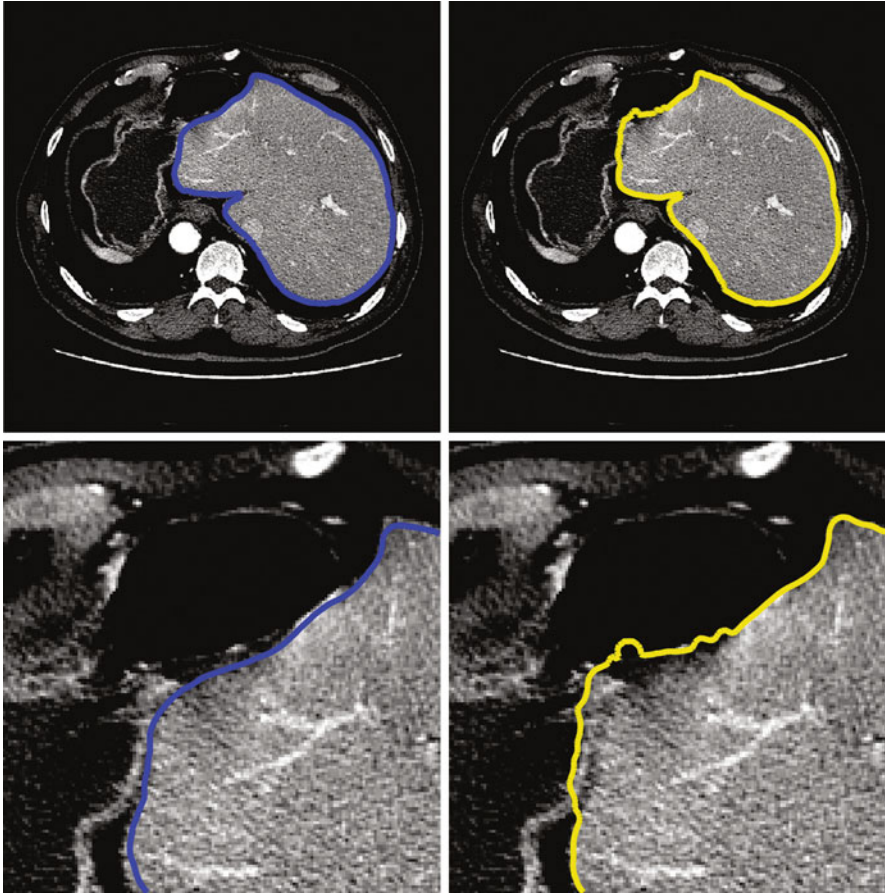


Fig. 1 The 1st row: liver segmentation results, from left to right: by CV model (Chan and Vese 2001) and method from Gui et al. (2017c); the 2nd row: zoomed regions of the segmentation results

The penalty for length is one of the most famous regularization terms in the segmentation model, such as the MS model (1) and CV model (2). Although the constraint on the length of the contour helps cope with problems such as a certain amount of noise in the image, it also brings a bias towards smaller-length contour lines, which leads to isotropic smooth segmentation curves, and small/shortened objects.

The total variation regularization can smooth only the tangent direction of each level line

$$R_{TV}(\phi) = \sup \left\{ \int_{\Omega} u \operatorname{div} \phi : \phi \in C'_c, \|\phi\| \leq \infty \right\} \quad (4)$$

and the H^1 regularization

$$R_{H^1}(\phi) = \mu \int_{\Omega} |\nabla\phi(x)|^2 dx, \quad (5)$$

applies a purely isotropic smoothing at every pixel x .

Curvature-based regularity is another valuable type of regularization. In psychophysical experiments on contour completion (Kanizsa 1974), the curvature is considered to be an important part of human perception. So curvature regularity (Osher and Sethian 1988) is often used to segment obscured targets (Esedoglu and March 2003) and some thin and elongated targets. Comparative experiments in Schoenemann and Cremers (2007) show that the length-based regularity term usually converges to a small curve enclosing a few pixels only due to intensity inhomogeneity, low contrast, initial positions, etc. In contrast, the curvature-based regularity term usually provides a more meaningful area, i.e., the region of the entire objects.

Another famous curvature-based regularization is the elasticity regularity (Tai et al. 2011); the standard Euler's elastic energy of the curve γ can be written as follows:

$$R_{\text{elastic}}(\gamma) = \int_{\gamma} (a + b\kappa^2) ds, \quad (6)$$

where κ is the curvature γ , and two parameters $a, b > 0$. The most remarkable feature of elastic regularity is that it promotes convex contours. It may therefore be used for some particular task of segmenting objects with a convex shape (Bae et al. 2017). And in the snake model (3), the regularization term then consists of two components, the bending energy and the elastic energy, where the bending energy is defined as the sum of the squared curvature of the curve, generating the bending force. In contrast, the elastic energy prevents the stretching of the curve by introducing tension.

In addition to restriction on the nature of the curve itself, regularization terms on the curve have also been proposed as a guarantee of stability and speed of evolution. For instance, Li et al. (2010) avoided re-initialization of the level set by imposing restriction on the gradient of the high-dimensional surface ϕ while ensuring evolutionary stability, making larger steps and faster speeds possible. Yu et al. (2019) performed a restriction on a small neighborhood of zero-level set functions by adding a perturbation factor, thus breaking the pseudo-balance due to heavy noise and then reaching the global optimum.

Targeted Regularization Terms Arising from Object Properties

This type of constraint is usually derived from the nature of the segmentation target itself and is therefore also commonly referred to as prior information. The shape,

geometric, and topological properties of the target are widely applied to promote segmentation efficiency.

The segmentation task in medical images is usually to segment out some organs, tissues, or lesions. Fortunately, some organs and tissues have generally similar morphological features. Although the images are subject to imaging errors and individual differences, the shape prior is a robust semantic descriptor for specifying targeted objects. In our categorization, shape prior can be modeled in two ways: building statistical templates and representing by analytical expressions.

Some simple shapes, such as circles or ellipses, can be expressed analytically, and by optimizing the parameters of these analytic expressions, the shape constraints of this analytic representation can be adapted to different variations of the segmented objects, including scale, rotation, and translation (Ray and Acton 2004).

For complex shapes that are difficult to express analytically, an alternative approach is to use a prior shape representation in the form of templates. Template-based shape priors are usually obtained by training on a set of similar shapes. Some researchers have studied the distribution of points on significant positions of the object, also called landmark points, to build a shape template for the object (Cootes et al. 1995), and some researchers employed boundary points as the shape templates (Grenander et al. 2012; Mardia et al. 1991). Subsequently, this kind of parametric point distribution shape prior was also extended into a hybrid segmentation model incorporating intensities (Grenander and Miller 1994) or both gradient and region-homogeneity information (Chakraborty et al. 1994). In the level-set-based approaches, shape constraint is represented as a zero level set of a higher-dimensional surface. Any deviation from the shape can be penalized (Leventon et al. 2002); a simple way to calculate the dissimilarity between them is given by $\int_{\Omega} (\phi_1 - \phi_2)^2 dx$, where ϕ_1 and ϕ_2 are shape constraint and segmented contour, respectively. Usually, to fit the unknown segmentation target, parameters of position, scale, orientation, and other information are also included in the shape energy term (Chen et al. 2002; Pluempitwiriyaew et al. 2005).

In addition to specific shapes, segmentation targets on medical images may have other more general morphological properties that allow researchers to add them as high-level information to the energy functional as effective constraints. For example, many objects have convex characteristics. As mentioned above, the curvature-based elastic energy term can maintain the convexity of the target. In addition, the limitation of the region can also provide the convexity of the segmentation target (Li et al. 2019; Yan et al. 2020; Luo et al. 2019). In medical images, the left ventricle segmentation is a representative example of the need to preserve the convexity of the object (Feng et al. 2016; Shi and Li 2021; Hajiaghayi et al. 2016). Segmentation of the left ventricle (LV) is critical for the diagnosis of cardiovascular disease. Accurate assessment of crucial clinical parameters such as ejection fraction, myocardial mass, and beat volume depends on the segmentation of the LV, that is, the precise segmentation of the endocardial border. According to the anatomy of the left ventricle, the left ventricle includes the cardiac chambers, trabeculae, and papillary muscles surrounded by the myocardium. Although there is good contrast

between myocardium and blood flow on MR images, there are still difficulties in segmentation. This problem is mainly due to the presence of papillary muscles and trabeculae (irregular walls) within the ventricles. They have the same intensity distribution as the surrounding myocardial tissue. Therefore, they can easily mislead the segmentation algorithm and prevent the walls from being clearly depicted, causing critical difficulties in endocardial segmentation.

In addition to the above geometric features, many other regularization terms proposed for segmented object characteristics can also facilitate segmentation. For example, some segmentation objects have a tendency to cluster together, which is defined as compactness. This characteristic can be used as constraint in segmentation organs, such as liver, prostate, as well as cysts and most hepatocellular carcinoma (Gui et al. 2017b). Considering that segmented objects in medical images may present deformation due to lesions, researchers used low-order moment as regularity to constrain the size/volume (Ayed et al. 2008) or location (Klodt and Cremers 2011) of the objects. Figure 2 shows the different segmentation results given by the two methods, one using the classical GAC method (Caselles et al. 1997) without any prior and the other using the intensity information of the image and the isoperimetric shape prior (Gui et al. 2017b). The differences between the two segmentation results can be observed by zooming in on the region.

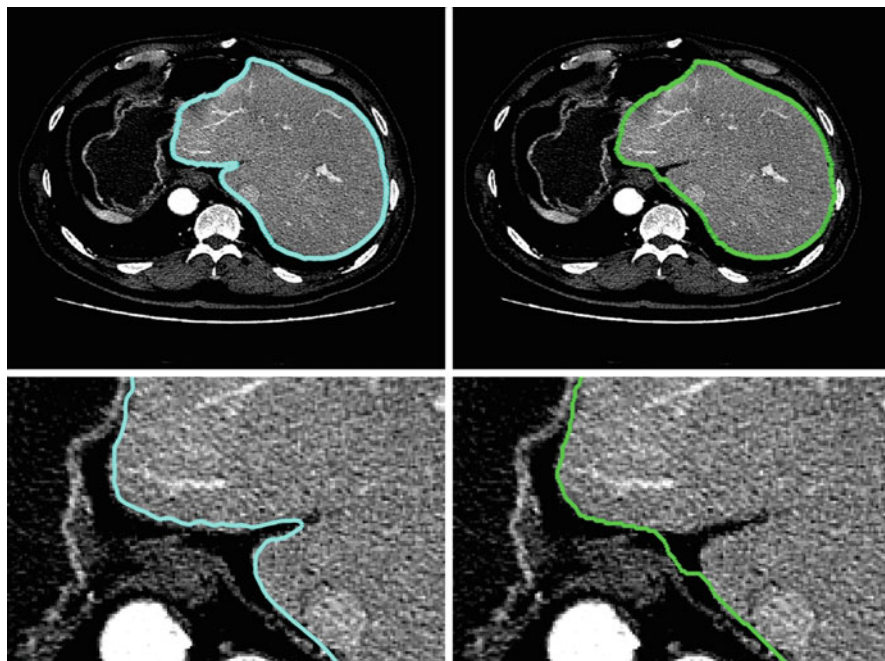


Fig. 2 The 1st row: liver segmentation results, from left to right: by geodesic active contours (GAC) (Caselles et al. 1997) and method from Gui et al. (2017b); the 2nd row: zoomed regions of the segmentation results

Variational Models Meet Deep Learning in Medical Image Segmentation

Since 2015, deep learning has gradually dominated medical image segmentation methods. A typical segmentation network is composed of an encoder network followed by a decoder network. The encoder network aims to extract and aggregate features from input images, and the decoder network is to project the features onto the pixel space to get dense predictions. In this way, the deep learning network can directly generate pixel-wise segmentation results with input images. Thus, a natural problem is that could one combine the advantages of deep learning networks and variational models. In this section, we will summarize the progress in this direction.

Variational Models Guided Deep Learning

Variational Model-Inspired Network Modules

Variational models lack learning ability that cannot obtain discrimination ability from the labeled dataset¹. On the other hand, deep learning methods have poor interpretabilities. In order to formulate the variational model in a learnable framework and increase the interpretability of deep learning, Le et al. (2018b) reformulated the level set (Chan and Vese 2001) evolution as a deep recurrent neural network (Cho et al. 2014) because both of them are time sequence process. In general, the level set function ϕ was updated by

$$\phi_{t+1} = \phi_t + \eta \frac{\partial \phi_t}{\partial t}, \quad (7)$$

where η is the step size (or the learning rate in deep learning). Then, sequence data $\{x_t\}$ for recurrent network input are generated based on the level set evolution:

$$x_{t+1} = \kappa(\phi_t) - F_\theta(I - c_1)^2 + B_\theta(I - c_2)^2, \quad (8)$$

where $\kappa(\phi) = -\mathbf{div}(\frac{\nabla \phi}{|\nabla \phi|})$ is the curvature, and F_θ and B_θ are the learnable parameters that control the force of foreground and background, respectively. This procedure corresponds to the minimization of Chan-Vese energy functional (Chan and Vese 2001) composed of the data fitting term and the contour length term. The final network layer output is computed from the hidden state ϕ_t followed by a Softmax layer to obtain foreground and background segmentation probability maps. The variational level set and this deep learning level set have the same input, including the image and the initial level set function. However, they have

¹The network module is a combination of several network layers, which is part of the network. For example, the well-known U-Net consists of multiple Convolution-Batch Normalization-ReLU modules.

different update rules and outputs. Specifically, the variational level set is updated by the gradient flow of the energy functional, and the output is still a level set function, while deep learning level set is updated by network layers with learnable hyperparameters, and the output is the Softmax probability map.

This network module can be directly connected to existing segmentation networks with convolutional layers and deconvolutional layers for medical image segmentation. For example, Le et al. proposed deep recurrent level set network for brain tumor segmentation (Le et al. 2018a), which achieved less computational time during inference and improved the Dice Similarity Coefficient (DSC) by 1–2%.

In addition to unrolling the level set evolution as network modules, regularizers or priors in classical variational models can also be incorporated into segmentation networks for end-to-end learning. The main challenge is to formulate the non-smooth constraints as differentiable network modules. Typical segmentation CNNs (Ronneberger et al. 2015; Çiçek et al. 2016; Shelhamer et al. 2017) predict each pixel independently and do not explicitly consider the dependency between pixels, which could lead to isolated or scattered small segmentation errors, especially when only few training data is available. To embed spatial regularity in segmentation CNNs, Jia et al. proposed total variation (TV) regularized segmentation CNNs (Jia et al. 2021) to add spatial regularization to the segmented networks, which can produce smooth edges and eliminate isolated segmentation errors. This approach was further applied to pancreas segmentation (Fan and Tai 2019) by unfolding the primal-dual block of TV regularizer and embedding in 2D U-Net (Ronneberger et al. 2015). This type of method has two main benefits. On the one hand, it can produce smooth segmentation edges and eliminate isolated segmentation errors. On the other hand, it is more efficient than the commonly used post-processing methods (Kamnitsas et al. 2017). In order to explicitly add non-local priors to CNNs, Jia et al. (2020) introduced graph total variation to the Softmax function by a primal-dual hybrid gradient method, which can capture long-range information.

Some common shape priors were embedded in segmentation CNNs by reformulating the Softmax layer. Liu et al. (2020b) proposed a Soft Threshold Dynamics framework to integrate many spatial priors of the classical variational models into segmentation CNNs, including spatial regularization, volume, and star-shape priors. The key idea to interpret the Softmax function s is to consider it as a solution of the following variational problem:

$$\min_s - \langle s, o \rangle + \langle s, \ln s \rangle, \quad (9)$$

where o is the network output in the last layer and $\sum_{i=1}^N s_i = 1$ (N is the number of classes). In this way, many spatial priors can be imposed on the Softmax results by adding corresponding terms on the energy functional (9). Furthermore, a Soft Threshold Dynamics algorithm was designed to solve the regularized variation problems, which enable stable and fast convergence during forward and backward propagation. Similarly, the convex shape prior (Liu et al. 2020a) and volume-preserving regularization (Li et al. 2020a) were also imposed on segmentation

CNNs. In addition, different priors can be used in combination. For example, using both special regularization and the convex prior can make the segmentation boundary simultaneously smooth and convex.

Variational Model-Inspired Loss Functions

The energy functional of variational models can be directly used as loss functions to guide the learning procedure of segmentation CNNs.

The Mumford-Shah model-inspired loss function (Kim and Ye 2019) This loss function is based on the observation that the characteristic function in the Mumford-Shah model has a striking similarity to the Softmax function in segmentation CNNs. Thus, Kim et al. proposed the following loss function by replacing the characteristic function with Softmax function:

$$L_{MS}(\Theta; I) = \sum_{i=1}^N \int_{\Omega} |I(\mathbf{x}) - c_i|^2 S_i(I(\mathbf{x}); \Theta) d\mathbf{x} + \lambda \sum_{i=1}^N \int_{\Omega} |\nabla S_i(I(\mathbf{x}); \Theta)| d\mathbf{x}, \quad (10)$$

where Θ is the trainable network parameters and

$$c_i = \frac{\int_{\Omega} I(\mathbf{x}) S_i(\mathbf{x}; \Theta) d\mathbf{x}}{\int_{\Omega} S_i(\mathbf{x}; \Theta) d\mathbf{x}} \quad (11)$$

is the average intensity value of the i -th class. This loss function enables semi-supervised and unsupervised segmentation, which only requires limited labeled data.

Chan-Vese model-inspired loss function Kim et al. introduced level set loss (Kim et al. 2019) by using the region term of Chan-Vese model, which is defined by

$$L_{LevelSet} = \int_{\Omega} |I_{GT} - c_1|^2 H_{\epsilon}(\phi_{\Theta}) d\mathbf{x} + \int_{\Omega} |I_{GT} - c_2|^2 (1 - H_{\epsilon}(\phi_{\Theta})) d\mathbf{x}, \quad (12)$$

where ϕ_{Θ} is the predicted level set function by the network with parameters Θ and $H_{\epsilon}(\phi_{\Theta}) = \frac{1}{2} (1 + \tanh(\frac{\phi_{\Theta}}{\epsilon}))$. c_1 and c_2 denote the average values of the interior and exterior of the contour, which are defined by

$$c_1 = \frac{\int_{\Omega} I_{GT} H_{\epsilon}(\phi_{\Theta}) d\mathbf{x}}{\int_{\Omega} H_{\epsilon}(\phi_{\Theta}) d\mathbf{x}}$$

and

$$c_2 = \frac{\int_{\Omega} I_{GT} (1 - H_{\epsilon}(\phi_{\Theta})) d\mathbf{x}}{\int_{\Omega} (1 - H_{\epsilon}(\phi_{\Theta})) d\mathbf{x}},$$

respectively.

Chen et al. proposed an active contour loss (Chen et al. 2019) to consider the area inside and outside objects as well as the size of boundaries during learning. In particular, it introduces total variation to approximate the boundary length and membership functions to compute the region area, which is defined by

$$\begin{aligned} L_{ActiveContour} &= Length + \lambda Region \\ &= \int_{\Omega} |\nabla S_{\theta}| d\mathbf{x} + \int_{\Omega} |I_{GT} - c_1|^2 S_{\theta} + |I_{GT} - c_2|^2 (1 - S_{\theta}) d\mathbf{x}, \end{aligned} \quad (13)$$

where S_{θ} is the predicted Softmax probability map.

Both level set loss (Kim et al. 2019) and active contour loss (Chen et al. 2019) were derived from the Chan-Vese model (Chan and Vese 2001). The main difference is that the mean intensity values of the interior and exterior of the contour are fixed to 1 (foreground) and 0 (background), respectively, in the active contour loss, while the values are iteratively updated in the level set loss.

In addition to fully supervised segmentation tasks, Gur et al. (2019) introduced a new loss term for unsupervised micro-vascular image segmentation. The loss term was based on the morphological optimization method of Chan-Vese model (Marquez-Neila et al. 2013), which is defined by

$$L_{morph-AC} = \|\nabla S_{\theta}\|_1 ((I - c_1)^2 - 2(I - c_2)^2), \quad (14)$$

where ∇S_{θ} is the intermediate segmentation derivative, computed by the central differences.

Geodesic active contour inspired loss (Ma et al. 2021b) To explicitly embed object global information in segmentation CNNs, Ma et al. proposed a level set regression network with the geodesic active contour loss function:

$$L_{GAC} = \int_{\Omega} g_I \delta_{\epsilon}(\phi_{\theta}) |\nabla \phi_{\theta}| d\mathbf{x}, \quad (15)$$

where $g_I = \frac{1}{1+|\nabla I|}$ is the edge indicator function. Different from the level set loss and active contour loss that only used the groundtruth information, the geodesic active contour loss explicitly introduced the image gradient information, which can guide the CNNs to capture detailed boundary information.

Figure 3 presents the visualized segmentation results of different methods on left atrial MRI and pancreas CT images (Fig. 3-a). Commonly used Dice loss (Milletari et al. 2016) (Fig. 3-b) may have obvious segmentation errors because it does not have any global constraint. Level set loss (Kim et al. 2019) (Fig. 3-c) and active contour loss (Chen et al. 2019) (Fig. 3-d) generate similar results that are better than the Dice loss. However, there are still some isolated outliers in the segmentation results. In contrast, the learning GAC (Ma et al. 2021b) (Fig. 3-e) significantly reduces the isolated segmentation masses, and the boundaries are closer to the

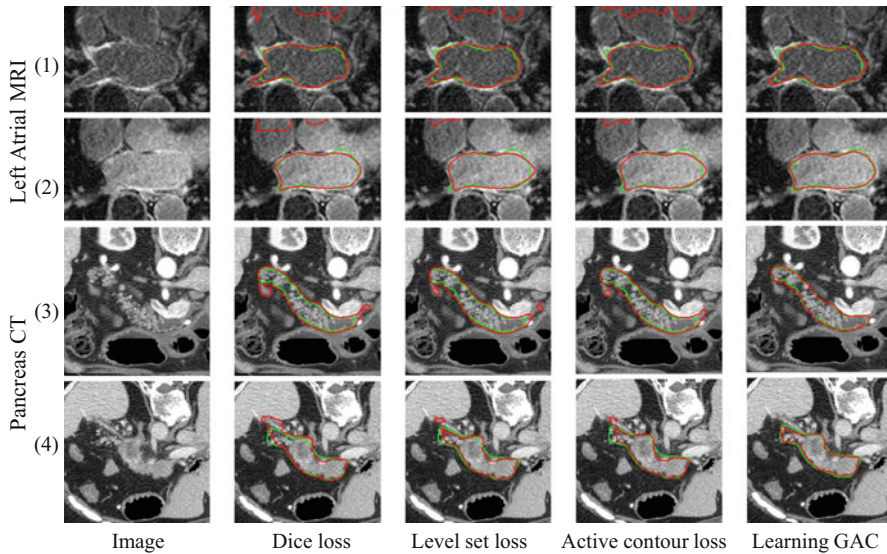


Fig. 3 Qualitative comparisons between commonly used Dice loss (Milletari et al. 2016), Chan-Vese model-inspired level set loss (Kim et al. 2019), active contour loss (Chen et al. 2019), and geodesic active contours inspired learning GAC method (Ma et al. 2021b). The green and red contours denote groundtruth and segmentation results, respectively. (a) Image. (b) Dice loss. (c) Level set loss. (d) Active contour loss. (e) Learning GAC

ground truth. This is because the learning GAC explicitly considers the image boundary information and geodesic geometry constraint, which can guide the network outputs to achieve a lower-energy state of the geodesic active contour model and then lead to more accurate results in boundary regions. In addition, it should be noted that the above variational model-inspired loss functions should be added to the Dice loss in a supervised learning framework.

Deep Learning-Driven Variational Models

Classical variational models are sensitive to initializations and hyperparameters settings. To address this limitation, many researchers use deep learning to directly generate initial segmentation contours and learn hyperparameters. On the other hand, variational models can help deep learning methods to obtain more accurate boundaries. The learning paradigm can be classified into two categories: two-stage framework and end-to-end framework.

Learning Hyperparameters in Two-Stage Framework

Hoogi et al. (2017) used CNN to estimate the hyperparameters of the mean separation model (Yezzi et al. 2002), and the energy functional was defined by

$$\min_{\phi, c_1, c_2} \int_{\Omega} \delta(\phi) |\nabla \phi| d\mathbf{x} + \lambda_1 \int_{\Omega} \frac{(I - c_1)^2}{A_1} H(\phi) d\mathbf{x} + \lambda_2 \int_{\Omega} \frac{(I - c_2)^2}{A_2} (1 - H(\phi)) d\mathbf{x} \quad (16)$$

where $A_1 = \int_{\Omega} H(\phi) d\mathbf{x}$ and $A_2 = \int_{\Omega} (1 - H(\phi)) d\mathbf{x}$ are the area of the local interior and exterior regions surrounding the contour. To adaptively estimate the region term weights λ_1 and λ_2 separately for each case during contour evolution, a CNN was employed to predict the location of the zero level set contour relative to the segmentation target (e.g., lesions), and the output was a probability for each of three classes: inside the lesion and far from its boundaries (p_1), close to the boundaries of the lesion (p_2), or outside the lesion and far away from its boundaries (p_3). The weight parameters were set as follows:

$$\lambda_1 = \exp\left(\frac{1 + p_2 + p_3}{1 + p_1 + p_2}\right), \quad \lambda_2 = \exp\left(\frac{1 + p_1 + p_2}{1 + p_2 + p_3}\right). \quad (17)$$

If $p_1 > p_3$, then $\lambda_2 > \lambda_1$ and the contour will expand. Conversely, if $p_3 > p_1$, then $\lambda_1 > \lambda_2$ and the contour tend to shrink. In this way, the contour can be adaptively expanded or shrunk towards the object boundary without any manual tuning.

Instead of predicting the contour location, Hatamizadeh et al. (2019) used an encoder-decoder network to predict the segmentation probability map S_{θ} . The weights was set as follows:

$$\lambda_1 = \exp\left(\frac{2 - S_{\theta}}{1 + S_{\theta}}\right), \quad \lambda_2 = \exp\left(\frac{1 + S_{\theta}}{2 - S_{\theta}}\right). \quad (18)$$

Experiments on various lesion segmentation tasks (e.g., brain lesion, liver lesion, lung lesion) and image modalities (CT and MR) show that the proposed method can produce more accurate and detailed boundaries compared with only using CNNs.

Learning Hyperparameters in End-to-End Framework

In order to avoid manual hyperparameter tuning, Zhang et al. (2020) proposed a deep active contour network (DACN) by integrating the convexified Chan-Vese model (Chan et al. 2006) into the DenseUNet (Huang et al. 2017; Ronneberger et al. 2015). The original Chan-Vese model is reduced to a convex minimization problem:

$$\min_{0 \leq u \leq 1} |\nabla u|_1 + \lambda(u, (I - c_1)^2 - (I - c_2)^2). \quad (19)$$

This minimization problem can be solved by the split Bregman algorithm (Goldstein et al. 2010). In the forward propagation, the DenseU-Net generated initial contours and pixel-wise hyperparameter maps of Eq. (19). Then, the contours, maps, and input images were transmitted to the active contour model that was solved by the split Bregman algorithm (Goldstein et al. 2010). The whole network was trained by comparing the final output to the ground truth with cross-entropy loss function.

Ali et al. proposed Trainable Deep Active Contours (TDACs) (Hatamizadeh et al. 2020) based on a standard encoder-decoder CNN and localized Chan-Vese model (Lankton and Tannenbaum 2008), which can explicitly capture local image information. The network also directly predicted pixel-wise hyperparameter maps and the initialization map that were used by the localized Chan-Vese model to update the segmentation results. The network and active contour modules of TDAC was simultaneously trained in an end-to-end manner. Both the initialization map and the active contour model output were passed to a Sigmoid function to generate final segmentation predictions. The loss function is the combination between cross entropy and Dice loss (Milletari et al. 2016) because the compound loss has been proved to be robust in segmentation tasks (Ma et al. 2021a).

Conclusion

In this paper, we have introduced the typical variational models and their combinations with modern deep learning methods, which have many applications in medical image segmentation. We have witnessed several different strategies to fuse the merits of variational models and deep learning methods. However, there is still a lack of the public segmentation benchmark to evaluate and compare these methods in a common and fair platform. We hope this survey can reach broad audiences with diverse backgrounds and inspire more inter-crossing researches between variational models and deep learning.

References

- Ali, H., Rada, L., Badshah, N.: Image segmentation for intensity inhomogeneity in presence of high noise. *IEEE Trans. Image Process.* **27**(8), 3729–3738 (2018)
- Ayed, I.B., Li, S., Islam, A., Garvin, G., Chhem, R.: Area prior constrained level set evolution for medical image segmentation. In: *Medical Imaging 2008: Image Processing*, vol. 6914, p. 691402. International Society for Optics and Photonics (2008)
- Bae, E., Tai, X.C., Wei, Z.: Augmented lagrangian method for an Euler’s elastica based segmentation model that promotes convex contours (2017)
- Balafar, M.: Gaussian mixture model based segmentation methods for brain MRI images. *Artif. Intell. Rev.* **41**(3), 429–439 (2014)
- Beichel, R., Bischof, H., Leberl, F., Sonka, M.: Robust active appearance models and their application to medical image analysis. *IEEE Trans. Med. Imaging* **24**(9), 1151–1169 (2005)
- Bernard, O., Lalonde, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G., et al.: Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE Trans. Med. Imaging* **37**(11), 2514–2525 (2018)
- Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.W., Han, X., Heng, P.A., Hesser, J., et al.: The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056* (2019)
- Boonnuak, T., Srisuk, S., Sripramong, T.: Texture segmentation using active contour model with edge flow vector. *Int. J. Inf. Electron. Eng.* **5**(2), 107 (2015)

- Boykov, Y., Funka-Lea, G.: Graph cuts and efficient ND image segmentation. *Int. J. Comput. Vis.* **70**(2), 109–131 (2006)
- Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001)
- Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vis.* **22**(1), 61–79 (1997)
- Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432 (2016)
- Chakraborty, A., Staib, L.H., Duncan, J.S.: An integrated approach to boundary finding in medical images. In: *Proceedings of IEEE Workshop on Biomedical Image Analysis*, pp. 13–22. IEEE (1994)
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chan, F., Lam, F., Poon, P., Zhu, H., Chan, K.: Object boundary location by region and contour deformation. *IEE Proc.-Vis. Image Sig. Process.* **143**(6), 353–360 (1996)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006)
- Chen, Y., Tagare, H.D., Thiruvankadam, S., Huang, F., Wilson, D., Gopinath, K.S., Briggs, R.W., Geiser, E.A.: Using prior shapes in geometric active contours in a variational framework. *Int. J. Comput. Vis.* **50**(3), 315–328 (2002)
- Chen, X., Williams, B.M., Vallabhaneni, S.R., Czanner, G., Williams, R., Zheng, Y.: Learning active contour models for medical image segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11632–11640 (2019)
- Cheplygina, V., de Bruijne, M., Pluim, J.P.: Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Med. Image Anal.* **54**, 280–296 (2019)
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: *Empirical Methods in Natural Language Processing (EMNLP)* (2014)
- Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: Use of active shape models for locating structures in medical images. *Image Vis. Comput.* **12**(6), 355–365 (1994)
- Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. *Comput. Vis. Image Underst.* **61**(1), 38–59 (1995)
- Cootes, T., Baldock, E., Graham, J.: An introduction to active shape models. *Image Process. Anal.* **328**, 223–248 (2000)
- Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 681–685 (2001)
- Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *Int. J. Comput. Vis.* **72**(2), 195–215 (2007)
- Esedoglu, S., March, R.: Segmentation with depth but without detecting junctions. *J. Math. Imaging Vis.* **18**(1), 7–15 (2003)
- Falah, R.K., Bolon, P., Cocqueruz, J.P.: A region-region and region-edge cooperative approach of image segmentation. In: *Proceedings of 1st International Conference on Image Processing*, vol. 3, pp. 470–474. IEEE (1994)
- Fan, J., Tai, X.c.: Regularized unet for automated pancreas segmentation. In: *Proceedings of the Third International Symposium on Image Computing and Digital Medicine*, pp. 113–117 (2019)
- Feng, C., Zhang, S., Zhao, D., Li, C.: Simultaneous extraction of endocardial and epicardial contours of the left ventricle by distance regularized level sets. *Med. Phys.* **43**(6Part1), 2741–2755 (2016)
- Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split bregman method: segmentation and surface reconstruction. *J. Sci. Comput.* **45**(1), 272–293 (2010)
- Grenander, U., Miller, M.I.: Representations of knowledge in complex systems. *J. R. Stat. Soc.: Ser. B (Methodological)* **56**(4), 549–581 (1994)

- Grenander, U., Chow, Y.-S., Keenan, D.M.: Hands: A pattern theoretic study of biological shapes, vol. 2. Springer Science & Business Media, New York (2012)
- Gui, L., Yang, X.: Automatic renal lesion segmentation in ultrasound images based on saliency features, improved lbp, and an edge indicator under level set framework. *Med. Phys.* **45**(1), 223–235 (2018)
- Gui, L., He, J., Qiu, Y., Yang, X.: Integrating compact constraint and distance regularization with level set for hepatocellular carcinoma (HCC) segmentation on computed tomography (CT) images. *Sens. Imaging* **18**(1), 4 (2017a)
- Gui, L., Li, C., Yang, X.P.: Medical image segmentation based on level set and isoperimetric constraint. *Phys. Med.* **42**, 162–173 (2017b)
- Gui, L., Yang, X., Cremers, A.B., Chen, Y.: Dempster-shafer evidence theory-based CV model for renal lesion segmentation of medical ultrasound images. *J. Med. Imaging Health Inform.* **7**(3), 595–606 (2017c)
- Gur, S., Wolf, L., Golgher, L., Blinder, P.: Unsupervised microvascular image segmentation using an active contours mimicking neural network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 10722–10731 (2019)
- Haddon, J.F., Boyce, J.F.: Image segmentation by unifying region and boundary information. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 929–948 (1990)
- Hajiaghayi, M., Groves, E.M., Jafarkhani, H., Kheradvar, A.: A 3-D active contour method for automated segmentation of the left ventricle from magnetic resonance images. *IEEE Trans. Biomed. Eng.* **64**(1), 134–144 (2016)
- Hatamizadeh, A., Hoogi, A., Sengupta, D., Lu, W., Wilcox, B., Rubin, D., Terzopoulos, D.: Deep active lesion segmentation. In: International Workshop on Machine Learning in Medical Imaging, pp. 98–105 (2019)
- Hatamizadeh, A., Sengupta, D., Terzopoulos, D.: End-to-end trainable deep active contour models for automated image segmentation: delineating buildings in aerial imagery. In: European Conference on Computer Vision, pp. 730–746 (2020)
- Heller, N., Isensee, F., Maier-Hein, K.H., Hou, X., Xie, C., Li, F., Nan, Y., Mu, G., Lin, Z., Han, M., Yao, G., Gao, Y., Zhang, Y., Wang, Y., Hou, F., Yang, J., Xiong, G., Tian, J., Zhong, C., Ma, J., Rickman, J., Dean, J., Stai, B., Tejpaul, R., Oestreich, M., Blake, P., Kaluzniak, H., Raza, S., Rosenberg, J., Moore, K., Walczak, E., Rengel, Z., Edgerton, Z., Vasdev, R., Peterson, M., McSweeney, S., Peterson, S., Kalapara, A., Sathianathan, N., Papanikolopoulos, N., Weight, C.: The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: results of the kits19 challenge. *Med. Image Anal.* **67**, 101821 (2020)
- Hoogi, A., Subramaniam, A., Veerapaneni, R., Rubin, D.L.: Adaptive estimation of active contour parameters using convolutional neural networks and texture analysis. *IEEE Trans. Med. Imaging* **36**(3), 781–791 (2017)
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
- Iglesias, J.E., Sabuncu, M.R.: Multi-atlas segmentation of biomedical images: a survey. *Med. Image Anal.* **24**(1), 205–219 (2015)
- Isensee, F., Jäger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* **18**(2), 203–211 (2021)
- Jia, F., Tai, X.C., Liu, J.: Nonlocal regularized cnn for image segmentation. *Inverse Probl. Imaging* **14**(5), 891 (2020)
- Jia, F., Liu, J., Tai, X.C.: A regularized convolutional neural network for semantic image segmentation. *Anal. Appl.* **19**(01), 147–165 (2021)
- Ji, Z., Xia, Y., Sun, Q., Chen, Q., Xia, D., Feng, D.D.: Fuzzy local Gaussian mixture model for brain MR image segmentation. *IEEE Trans. Inf. Technol. Biomed.* **16**(3), 339–347 (2012)
- Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected crf for accurate brain lesion segmentation. *Med. Image Anal.* **36**, 61–78 (2017)

- Kanizsa, G.: Contours without gradients or cognitive contours? *Giornale Italiano di Psicologia* (1974)
- Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comput. Vis.* **1**(4), 321–331 (1988)
- Kavur, A.E., Gezer, N.S., Barış, M., Aslan, S., Conze, P.H., Groza, V., Pham, D.D., Chatterjee, S., Ernst, P., Özkan, S., Baydar, B., Lachinov, D., Han, S., Pauli, J., Isensee, F., Perkonig, M., Sathish, R., Rajan, R., Sheet, D., Dovletov, G., Speck, O., Nürnberger, A., Maier-Hein, K.H., Bozdağı Akar, G., Ünal, G., Dicle, O., Selver, M.A.: Chaos challenge – combined (CT-MR) healthy abdominal organ segmentation. *Med. Image Anal.* **69**, 101950 (2021)
- Kim, B., Ye, J.C.: Mumford–Shah loss functional for image segmentation with deep learning. *IEEE Trans. Image Process.* **29**, 1856–1866 (2019)
- Kim, Y., Kim, S., Kim, T., Kim, C.: CNN-based semantic segmentation using level set loss. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1752–1760 (2019)
- Klodt, M., Cremers, D.: A convex framework for image segmentation with moment constraints. In: 2011 International Conference on Computer Vision, pp. 2236–2243. IEEE (2011)
- Lankton, S., Tannenbaum, A.: Localizing region-based active contours. *IEEE Trans. Image Process.* **17**(11), 2029–2039 (2008)
- Le, T.H.N., Gummadi, R., Savvides, M.: Deep recurrent level set for segmenting brain tumors. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 646–653 (2018a)
- Le, T.H.N., Quach, K.G., Luu, K., Duong, C.N., Savvides, M.: Reformulating level sets as deep recurrent neural network approach to semantic segmentation. *IEEE Trans. Image Process.* **27**(5), 2393–2407 (2018b)
- Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: 5th IEEE EMBS International Summer School on Biomedical Imaging, 2002, p. 8. IEEE (2002)
- Li, C., Kao, C.Y., Gore, J.C., Ding, Z.: Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.* **17**(10), 1940–1949 (2008)
- Li, C., Xu, C., Anderson, A.W., Gore, J.C.: MRI tissue classification and bias field estimation based on coherent local intensity clustering: a unified energy minimization framework. In: International Conference on Information Processing in Medical Imaging, pp. 288–299. Springer (2009)
- Li, C., Xu, C., Gui, C., Fox, M.D.: Distance regularized level set evolution and its application to image segmentation. *IEEE Trans. Image Process.* **19**(12), 3243–3254 (2010)
- Li, C., Huang, R., Ding, Z., Gatenby, J.C., Metaxas, D.N., Gore, J.C.: A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. *IEEE Trans. Image Process.* **20**(7), 2007–2016 (2011)
- Li, L., Luo, S., Tai, X.C., Yang, J.: Convex hull algorithms based on some variational models. arXiv preprint arXiv:1908.03323 (2019)
- Li, H., Liu, J., Cui, L., Huang, H., Tai, X.C.: Volume preserving image segmentation with entropy regularized optimal transport and its applications in deep learning. *J. Vis. Commun. Image Rep.* **71**, 102845 (2020a)
- Li, X., Yang, X., Zeng, T.: A three-stage variational image segmentation framework incorporating intensity inhomogeneity information. *SIAM J. Imaging Sci.* **13**(3), 1692–1715 (2020b)
- Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghahoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017)
- Liu, J., Tai, X.C., Luo, S.: Convex shape prior for deep neural convolution network based eye fundus images segmentation. arXiv preprint arXiv:2005.07476 (2020a)
- Liu, J., Wang, X., Tai, X.C.: Deep convolutional neural networks with spatial regularization, volume and star-shape prior for image segmentation. arXiv preprint arXiv:2002.03989 (2020b)

- Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
- Lu, J., Wang, G., Pan, Z.: Nonlocal active contour model for texture segmentation. *Multimedia Tools Appl.* **76**(8), 10991–11001 (2017)
- Luo, S., Tai, X.C., Huo, L., Wang, Y., Glowinski, R.: Convex shape prior for multi-object segmentation using a single level set function. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 613–621 (2019)
- Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.: Loss odyssey in medical image segmentation. *Med. Image Anal.* **71**, 102035 (2021a)
- Ma, J., He, J., Yang, X.: Learning geodesic active contours for embedding object global information in segmentation CNNs. *IEEE Trans. Med. Imaging* **40**(1), 93–104 (2021b)
- Mardia, K., Kent, J., Walder, A.: Statistical shape models in image analysis. In: Proceedings of the 23rd Symposium on the Interface, Seattle, pp. 550–557 (1991)
- Marquez-Neila, P., Baumela, L., Alvarez, L.: A morphological approach to curvature-based evolution of curves and surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), 2–17 (2013)
- Martinez-Uso, A., Pla, F., Sotoca, J.M.: A semi-supervised Gaussian mixture model for image segmentation. In: 2010 20th International Conference on Pattern Recognition, pp. 2941–2944. IEEE (2010)
- Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571 (2016)
- Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
- Muñoz, X., Freixenet, J., Cufi, X., Martí, J.: Strategies for image segmentation combining region and boundary information. *Pattern Recogn. Lett.* **24**(1–3), 375–392 (2003)
- Niu, S., Chen, Q., De Sisternes, L., Ji, Z., Zhou, Z., Rubin, D.L.: Robust noise region-based active contour model via local similarity factor for image segmentation. *Pattern Recogn.* **61**, 104–119 (2017)
- Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
- Pluempitwiriyawej, C., Moura, J.M., Wu, Y.J.L., Ho, C.: Stacs: new active contour scheme for cardiac MR image segmentation. *IEEE Trans. Med. Imaging* **24**(5), 593–603 (2005)
- Pons, S.V., Rodríguez, J.L.G., Pérez, O.L.V.: Active contour algorithm for texture segmentation using a texture feature set. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4. IEEE (2008)
- Ray, N., Acton, S.T.: Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Trans. Med. Imaging* **23**(12), 1466–1478 (2004)
- Reska, D., Boldak, C., Kretowski, M.: A texture-based energy for active contour image segmentation. In: *Image Processing & Communications Challenges*, vol. 6, pp. 187–194. Springer (2015)
- Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241 (2015)
- Schoenemann, T., Cremers, D.: Introducing curvature into globally optimal image segmentation: minimum ratio cycles on product graphs. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–6. IEEE (2007)
- Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017)
- Shi, X., Li, C.: Convexity preserving level set for left ventricle segmentation. *Magn. Reson. Imaging* **78**, 109–118 (2021)
- Tai, X.C., Hahn, J., Chung, G.J.: A fast algorithm for Euler’s elastica model using augmented lagrangian method. *SIAM J. Imaging Sci.* **4**(1), 313–344 (2011)

- Tuceryan, M., Jain, A.K.: Texture analysis. In: Chen, CH, Pau, LF, Wang, PSP (eds) *The Handbook of Pattern Recognition and Computer Vision*, 2nd Edn., pp. 207–248. World Scientific (1998)
- Wu, Q., Gan, Y., Lin, B., Zhang, Q., Chang, H.: An active contour model based on fused texture features for image segmentation. *Neurocomputing* **151**, 1133–1141 (2015)
- Yan, S., Tai, X.C., Liu, J., Huang, H.Y.: Convexity shape prior for level set-based image segmentation method. *IEEE Trans. Image Process.* **29**, 7141–7152 (2020)
- Yezzi Jr, A., Tsai, A., Willsky, A.: A fully global approach to image segmentation via coupled curve evolution equations. *J. Vis. Commun. Image Rep.* **13**(1–2), 195–216 (2002)
- Yu, H., He, F., Pan, Y.: A novel segmentation model for medical images with intensity inhomogeneity based on adaptive perturbation. *Multimedia Tools Appl.* **78**(9), 11779–11798 (2019)
- Zhang, M., Dong, B., Li, Q.: Deep active contour network for medical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 321–331 (2020)
- Zhou, S., Wang, J., Zhang, M., Cai, Q., Gong, Y.: Correntropy-based level set method for medical image segmentation and bias correction. *Neurocomputing* **234**, 216–229 (2017)



Michal Haindl

Contents

Introduction	1025
Visual Texture	1027
Bidirectional Texture Function	1027
BTF Measurement	1029
Compound Markov Model	1030
Principal Markov Model	1031
Principal Single Model Markov Random Field	1032
Non-parametric Markov Random Field	1032
Non-parametric Markov Random Field with Iterative Synthesis	1033
Non-parametric Markov Random Field with Fast Iterative Synthesis	1035
Potts Markov Random Field	1037
Potts-Voronoi Markov Random Field	1038
Bernoulli Distribution Mixture Model	1040
Gaussian Mixture Model	1041
Local Markov and Mixture Models	1042
3D Causal Simultaneous Autoregressive Model	1042
3D Moving Average Model	1046
Spatial 3D Gaussian Mixture Model	1047
Applications	1049
Texture Synthesis and Enlargement	1050
Texture Compression	1053
Texture Editing	1053
Illumination Invariants	1053
(Un)supervised Image Recognition	1054
Multispectral/Multi-channel Image Restoration	1056
Conclusion	1057
References	1058

M. Haindl (✉)

Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czechia
e-mail: haindl@utia.cas.cz

Abstract

An authentic material's surface reflectance function is a complex function of over 16 physical variables, which are unfeasible both to measure and to mathematically model. The best simplified measurable material texture representation and approximation of this general surface reflectance function is the seven-dimensional bidirectional texture function (BTF). BTF can be simultaneously measured and modeled using state-of-the-art measurement devices and computers and the most advanced mathematical models of visual data. However, such an enormous amount of visual BTF data, measured on the single material sample, inevitably requires state-of-the-art storage, compression, modeling, visualization, and quality verification. Storage technology is still the weak part of computer technology, which lags behind recent data sensing technologies; thus, even for virtual reality correct materials modeling, it is infeasible to use BTF measurements directly. Hence, for visual texture synthesis or analysis applications, efficient mathematical BTF models cannot be avoided. The probabilistic BTF models allow unlimited seamless material texture enlargement, texture restoration, tremendous unbeatable appearance data compression (up to 1:1000 000), and even editing or creating new material appearance data. Simultaneously, they require neither storing actual measurements nor any pixel-wise parametric representation. Unfortunately, there is no single universal BTF model applicable for physically correct modeling of visual properties of all possible BTF textures. Every presented model is better suited for some subspace of possible BTF textures, either natural or artificial. In this contribution, we intend to survey existing mathematical BTF models which allow physically correct modeling and enlargement measured texture under any illumination and viewing conditions while simultaneously offering huge compression ratio relative to natural surface materials optical measurements. Exceptional 3D Markovian or mixture models, which can be either solved analytically or iteratively and quickly synthesized, are presented. Illumination invariants can be derived from some of its recursive statistics and exploited in content-based image retrieval, supervised or unsupervised image recognition. Although our primary goal is physically correct texture synthesis of any unlimited size, the presented models are equally helpful for various texture analytical applications. Their modeling efficiency is demonstrated in several analytical and modeling image applications, in particular, on a (un)supervised image segmentation, bidirectional texture function (BTF) synthesis and compression, and adaptive multispectral and multi-channel image and video restoration.

Keywords

Bidirectional texture function · Texture modeling · Markov random fields · Discrete distribution mixtures · Expectation-Maximization algorithm

Introduction

Multidimensional data modeling or understanding (or set of spatially related objects) is more accurate and efficient if we respect all interdependencies between single objects. Objects to be processed, for example, multispectral pixels, in a digitized image are often mutually dependent (e.g., correlated) with a dependency degree related to a distance between two objects in their corresponding data space. These relations can be incorporated into a pattern recognition or visualization process through an appropriate multidimensional data model. If such a model is probabilistic, we can benefit from a consistent Bayesian framework for solving many related visual or pattern recognition tasks.

Features derived from multidimensional data models are information preserving in the sense that they can be used to synthesize data spaces closely resembling original measurement data space as can be illustrated on the recent best visual representation of real material surfaces in the form of seven-dimensional bidirectional texture function (Haindl and Filip 2007; Filip and Haindl 2009). Virtual or augmented reality systems require object surfaces covered with physically correct nature-like color textures to enhance realism in visual scenes applied in computer games, CAD systems, or other computer graphics applications. Surface material appearance modeling thus aims to generate and enlarge a synthetic texture visually indiscernible from the visual properties of measured material, whatever the observation conditions might be.

While simple color textures can be either digitized measured natural textures or textures synthesized from an appropriate mathematical model, realistic 7D BTF textures require mathematical modeling. Measured BTF textures are far less convenient alternative, because of extreme virtual system memory demands, limited size measurements, visible discontinuities (if we apply some usual computer graphics sampling approach for texture enlargement (De Bonet 1997; Efros and Freeman 2001; Praun et al. 2000; Xu et al. 2000; Wei and Levoy 2000, 2001; Liang et al. 2001; Soler et al. 2002; Dong and Chantler 2002; Zelinka and Garland 2002; Haindl and Hatka 2005a,b; Ngan and Durand 2006)), or several other drawbacks (Haindl 1991). Some of these methods are based on per-pixel sampling (Wei and Levoy 2001; Tong et al. 2002; Zelinka and Garland 2003; Zhang et al. 2003) while other are patch-based sampling methods (Praun et al. 2000; Xu et al. 2000; Efros and Freeman 2001; Liang et al. 2001; Soler et al. 2002; Kwatra et al. 2003; Dong et al. 2010). Texture synthesis algorithms (Heeger and Bergen 1995; Liu and Picard 1996; Efros and Leung 1999; Portilla and Simoncelli 2000) view surface texture as a stochastic process and aim to produce new realizations that resemble an input exemplar by either copying pixels (non-parametric methods) or matching image statistics (parametric techniques). Some of these simple gray scale/color texture modeling methods, which also allow texture enlargement, could be formally applied independently for each BTF material space. However, this is infeasible for all about a thousand measurements for a single BTF material due to their enormous

computing time and memory constraints. Furthermore, for example, a car interior usually has about 20 different materials to synthesize.

Principle component analysis (PCA)-based BTF approximation (Müller et al. 2003; Sattler et al. 2003; Ruiters et al. 2013) allows BTF lossy compression but not enlargement. Furthermore, projecting the measured data onto a linear space constructed by statistical analysis such as PCA results in low-quality data compression. Another compression method (Tsai and Shih 2012) is based on K-clustered tensor approximation or the polynomial wavelet tree (Baril et al. 2008).

BTF data can be approximated using separate texel models, i.e., spatially varying bidirectional reflectance distribution function (SVBRDF) models that combine texture mapping and BRDF models but sacrifice some spatial dependency information. A linear combination of multivariate spherical radial basis functions is used to model BTF as a set of texelwise BRDFs (SVBRDF) in Tsai et al. (2011). Another SVBRDF method (Wu et al. 2011) uses a parametric mixture model with a basis analytical BRDF function for texel modeling. Several SVBRDF models use multilayer perceptron neural networks (Aittala et al. 2016; Deschaintre et al. 2018; Rainer et al. 2020). A deep convolutional neural network VGG-19 is used in Aittala et al. (2016), while the convolutional neural network recovers SVBRDF from estimated normal, diffuse albedo, specular albedo, and specular roughness from a single image lit by a handheld flash in Deschaintre et al. (2018). A learned SVBRDF decoder in a multilayer perceptron neural model approximates BRDF values in Rainer et al. (2020). The SVBRDF methods approximate BTF quality, are computationally expensive due to the nonlinear optimization, allow only moderate compression ratio, require several manually tuned parameters, and do not allow BTF space enlargement.

Mathematical multidimensional data models are useful for describing many of the multidimensional data types provided that we can assume some data homogeneity, so some data characteristics are a translation invariant. While the 1D models like time series (Anderson 1971; Broemeling 1985) are relatively well researched, and they have a rich application history in control theory, econometrics, medicine, meteorology, and many other data mining or machine learning applications, multidimensional models are much less known (e.g., more than three-dimensional MRF), and their applications are still limited. The reason is not only unsolved theory difficulties but mainly their vast computing power demands, which prevented their more extensive use until recently.

Visual data models need nonstandard multidimensional (three-dimensional for static color textures, four-dimensional for videos, or even seven-dimensional for static BTFs) models. However, if such a nD data space can be factorized, then these data can also be approximated using a set of lower-dimensional probabilistic models. Although full visual nD models allow unrestricted spatial-spectral-temporal-angular correlation modeling, their main drawback is many parameters to be estimated, which require a correspondingly large learning set. In some models (e.g., Markov models), the necessity is to estimate all these parameters simultaneously.

We introduced (Haindl and Havlíček 1998, 2000, 2010, 2016, 2017b, 2018a,b; Haindl et al. 2012, 2015b), several efficient fast multiresolution Markov random field (MRF)-based models which exploit BTF space factorization. Our methods avoid the time-consuming Markov chain Monte Carlo simulation (MCMC) so typical for Markov models applications with one exception of the Potts MRF. Our models avoid some problems of alternative options (see Haindl 1991 for details), but they are also easy to analyze as well as to synthesize, and last but not least, they are still flexible enough to correctly imitate a broad set of natural and artificial textures or other spatial data.

We can categorize the model's applications into synthesis and analysis. Analytical applications include static or dynamic data un-/semi-/supervised recognition, scene understanding, data space analysis, motion detection, and numerous others. Typical synthesis applications are missing data reconstruction, restoration, image compression, and static or dynamic texture synthesis.

Visual Texture

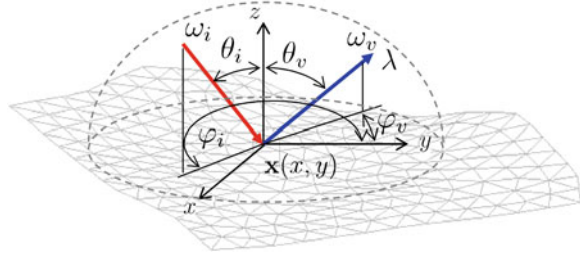
The visual texture notion is closely tied to the human semantic meaning of surface material appearance, and texture analysis is an essential and frequently published area of image processing. However, there is still no mathematically rigorous definition of the texture that would be accepted throughout the computer vision community.

We understand a textured image or the *visual texture* (Haindl and Filip 2013) to be a realization of a random field, and our effort is to find its parameterizations in such a way that the real texture representing the specific material appearance measurements will be visually indiscernible from the corresponding random field's realization, whatever the observation conditions might be. Some work distinguishes between texture and color. We regard such separation between spatial structure and spectral information to be artificial and principally wrong because there is no bijective mapping between gray scale and multispectral textures. Thus, our random field model is always multispectral.

Bidirectional Texture Function

A natural material's surface general reflectance function (GRF), representing physically correct visual properties of surface materials and their variations under any observation conditions, is a complex function of 16 physical variables. It is currently unfeasible to measure or to model such a function mathematically. Practical applications thus require significant simplification, namely, using additional assumptions. These approximative assumptions neglect the most less significant variables to achieve a solvable problem, with the solution still far more realistic

Fig. 1 BTF reflectance model



than the traditional three-dimensional static color texture representation. BTF can model complex lighting effects such as self-shadows, masking, foreshortening, interreflections, and multiple subsurface light scattering due to material surface microgeometry.

The seven-dimensional bidirectional texture function (BTF) reflectance model Fig. 1 is the best recent visual texture representation, which can still be simultaneously measured and modeled using state-of-the-art measurement devices and computers as well as the most advanced mathematical models of visual data. Thus, it is the most important representation for the high-end and physically correct surface materials appearance modeling. Nevertheless, BTF requires the most advanced modeling as well as high-end hardware support. The BTF reflectance model

$$Y_r^{BTF} = BTF(\lambda, x, y, \theta_i, \phi_i, \theta_v, \phi_v), \quad (1)$$

where Y_r^{BTF} is a random spectral reflectance vector at location r , r is a multiindex, and Y_r^{BTF} accepts six simplifying assumptions from GRF – light transport in material takes zero time ($t_i = t_v$ (incident time is equal to the reflection time) and $t_v = \emptyset$), reflectance behavior of the surface is time invariant ($t_v = t_i = const.$, $t_v = t_i = \emptyset$); interaction with the material does not change wavelength ($\lambda_i = \lambda_v$), i.e., $\lambda_v = \emptyset$), constant radiance along light rays ($z_i = z_v = \emptyset$), no transmittance ($\theta_t = \phi_t = \emptyset$), and incident light leaves at the same point.

Multispectral BTF is a seven-dimensional random function, which considers measurement dependency on color spectrum and planar material position, as well as its dependence on illumination incident light (lower index i) and viewing reflection light (lower index v) angles $BTF(r, \theta_i, \phi_i, \theta_v, \phi_v)$, where the multiindex $r = [r_1, r_2, r_3]$ specifies planar horizontal and vertical position in material sample image, r_3 is the spectral index, and θ, ϕ are elevation and azimuthal angles of the illumination and view direction vectors. The BTF measurements comprise a whole the hemisphere of light and camera positions in observed material sample coordinates according to selected quantization steps, and this is the main difference compared to the standard three-dimensional static color texture. This difference significantly improves the visual quality and realism of BTF representation and simultaneously complicates its measurement and modeling.

BTF Measurement

Accurate and reliable BTF acquisition is not a trivial task; only a few BTF measurement systems currently exist (for details see Haindl and Filip 2013; Schwartz et al. 2014; Dana et al. 1997; Koudelka et al. 2003; Sattler et al. 2003; Han and Perlin 2003; Müller et al. 2004; Wang and Dana 2006; Ngan and Durand 2006; Debevec et al. 2000; Marschner et al. 2005; Holroyd et al. 2010; Ren et al. 2011; Aittala et al. 2013, 2015). However, their number increases every year in response to the growing demand for photorealistic virtual representations of real-world materials. These systems are (similar to bidirectional reflectance distribution function (BRDF) measurement systems) based on the light source, video/still camera, and material sample. The main difference between individual BTF measurement systems is in the type of measurement setup allowing four degrees of freedom for camera/light, the type of measurement sensor (CCD, video, and some other), and light.

In some systems, the camera is moving, and the light is fixed (Dana et al. 1997; Sattler et al. 2003; Neubeck et al. 2005), while in others, e.g., Koudelka et al. (2003), it is just the opposite. There are also systems where both camera and light source remain fixed (Han and Perlin 2003; Müller et al. 2004).

The UTIA gonioreflectometer setup Fig. 2 consists of independently controlled arms with a camera and light. Its parameters, such as angular precision 0.03 degree, spatial resolution 1000 DPI, or selective spatial measurement, classify this

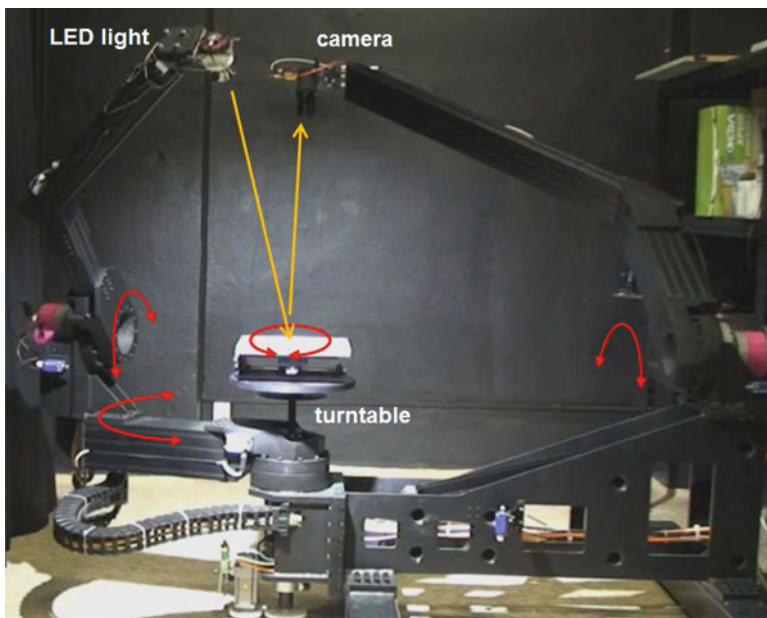


Fig. 2 UTIA gonioreflectometer

gonioreflectometer to the state-of-the-art devices. The typical resolution of the area of interest is around 2000×2000 pixels, sample size 7×7 [cm], and sensor distance ≈ 2 [m] with a field of view angle of 8.25° , and each of them is represented using at least 16-bit floating-point value for a reasonable representation of high-dynamic-range visual information. Illumination source is 11 LED arrays, each having a flux of 280 lm at 0.7 A, spectral wavelength 450 – 700 [nm], and its optics. The memory requirements for storage of a single material sample amount to 360 gigabytes per color channel but can be much more for a more precise spectral measurement.

We measure each material sample mostly in 81 viewing positions n_v and 81 illumination positions n_i , resulting in 6561 images per sample (4 terabytes of data).

Compound Markov Model

BTF data space is seven-dimensional, and thus it also requires seven-dimensional probabilistic models for physically correct BTF modeling, data compression, and enlargement with all related problems needed for robust estimation of all their numerous parameters. A practical alternative is to factorize a seven-dimensional problem into a set of lower-dimensional models with fewer parameters dedicated to model subparts of a BTF texture combined into a compound BTF model.

We exploit the compound Markov model for physically correct BTF modeling for either synthesis or analytical applications. Let us denote a multiindex $r = (r_1, r_2)$, $r \in I$, where I is a discrete two-dimensional rectangular lattice and r_1 is the row and r_2 the column index, respectively. The principal field pixel $X_r \in \mathcal{K}$ where \mathcal{K} is the index set of K distinguished sub-models, i.e., $X_r \in \{1, 2, \dots, K\}$ is a random variable with natural number value (a positive integer). Y_r is the multispectral pixel at location r and $Y_{r,j} \in \mathcal{R}$ is its j -th spectral plane component. Both random fields (X, Y) are indexed on the same $M \times N$ lattice I .

Let us assume that each multispectral observed texture \tilde{Y} (composed of d spectral planes, e.g., $d = 3$ for color textures) and indexed on the $\tilde{M} \times \tilde{N}$ lattice \tilde{I} (usually $\tilde{I} \subseteq I$ and \tilde{M}, \tilde{N} are number of rows and columns of the measured BTF texture) can be modeled by a compound Markov random field model (CMRF), where the principal Markov random field (MRF) X controls switching to a regional local MRF model ${}^i Y$ where $Y = \bigcup_{i=1}^K {}^i Y$. Single K regional random field sub-models ${}^i Y$ are defined on their corresponding lattice subsets ${}^i I$, ${}^i I \cap {}^j I = \emptyset \quad \forall i \neq j$, $I = \bigcup_{i=1}^K {}^i I$ ($X_r = X_s \quad \forall r, s \in {}^i I$) and they are of the same MRF type. These models differ only in their contextual support set ${}^i I_r$ and corresponding parameter sets ${}^i \theta$ (a set of all i -th local random field parameters). The same type of sub-models are assumed only for simplicity and can be omitted without any problems if needed. The BTF-CMRF model has a posterior probability

$$P(X, Y | \tilde{Y}) = P(Y | X, \tilde{Y})P(X | \tilde{Y}) \quad (2)$$

and the corresponding optimal maximum a posteriori (MAP) solution is

$$(\hat{X}, \hat{Y}) = \arg \max_{X \in \Omega_X, Y \in \Omega_Y} P(Y | X, \tilde{Y}) P(X | \tilde{Y}),$$

where Ω_X, Ω_Y are the corresponding configuration spaces for both random fields (X, Y) . To avoid an iterative MCMC MAP solution for parameter estimation, we proposed the following two-step approximation \check{X}, \check{Y} (Haindl and Havlíček 2010):

$$(\check{X}) = \arg \max_{X \in \Omega_X} P(X | \tilde{Y}), \quad (3)$$

$$(\check{Y}) = \arg \max_{Y \in \Omega_Y} P(Y | \check{X}, \tilde{Y}). \quad (4)$$

This approximation significantly simplifies the BTF-CMRF estimation without compromising random sampling for its synthesis because it allows us to take advantage of the possible analytical estimation of all regional MRF models $^i Y$ in (4). We randomly sample the required enlarged texture in the same order, i.e., at first (3) and, consequently, based on this principal random field realization, the local random fields (4). Furthermore, there is no need to have a unique solution of the (3), (4) approximation because the aim is to obtain a visually indiscernible result or results from the target observation. The subsequent Markovian/mixture compound models use the notation BTF-CMRF^{principal_model local_model} where the upper indices indicate the principal as well as the local model families.

Principal Markov Model

The principal part (X) of the BTF compound Markov models (BTF-CMRF) is assumed to be independent on illumination and observation angles, i.e., it is identical for all possible combinations $\phi_i, \phi_v, \theta_i, \theta_v$ azimuthal and elevation illumination/viewing angles, respectively. This assumption does not compromise the resulting BTF space quality because it influences only a material texture macrostructure independent of these angles for static BTF textures.

The principal random field \check{X} is estimated using simple K-means clustering of \tilde{Y} in the RGB color space into a predefined number of K classes, where cluster indices are $\check{X}_r \quad \forall r \in I$ estimates. We further use for simplicity the RGB color space, but any other color space can be used as well. The number of classes K can be estimated using the Kullback-Leibler divergence and considering a sufficient amount of data necessary to estimate all local Markovian models reliably. If the BTF texture contains subparts with distinct texture but similar colors, any more sophisticated texture segmenter (e.g., Haindl and Mikeš 2007; Haindl et al. 2009a,b, 2015a) can be used.

Principal Single Model Markov Random Field

The simplest principal model is a constant field that contains only one model BTF-CMRF^{c...} $P(X | \tilde{Y}) = \text{const.}$, i.e., $P(X_r | \tilde{Y}) = P(X_s | \tilde{Y}) \quad \forall r, s$. Then there is no need to use the MAP approximation (3), (4), and the compound Markov model simplifies into a single random field BTF-MRF model, and the BTF-MRF model can be any of the following local MRF models.

Non-parametric Markov Random Field

If we do not assume any specific principal control field parametric model, but rather we seamlessly and directly enlarge its realization from measured data (Fig. 3), we get several non-parametric principal control field approaches. The non-parametric principal field BTF-CMRF^{N Prol...} (NProl... – a non-parametric roller-based principal field with any local random fields denoted as ...; see Figs. 3, 4, 16) can be modeled using the roller method (Haindl and Havlíček 2010) for optimal \tilde{X} compression and speedy enlargement to any required field size. The roller method (Haindl and Hatka 2005a,b) principle is the overlapping tiling and subsequent minimum error boundary cut. One or several optimal double toroidal data patches are seamlessly and randomly repeated during the synthesis step. This fully automatic method starts with minimal tile size detection, which is limited by the size of the principal field, the number of toroidal tiles we are looking for, and the sample spatial frequency content.

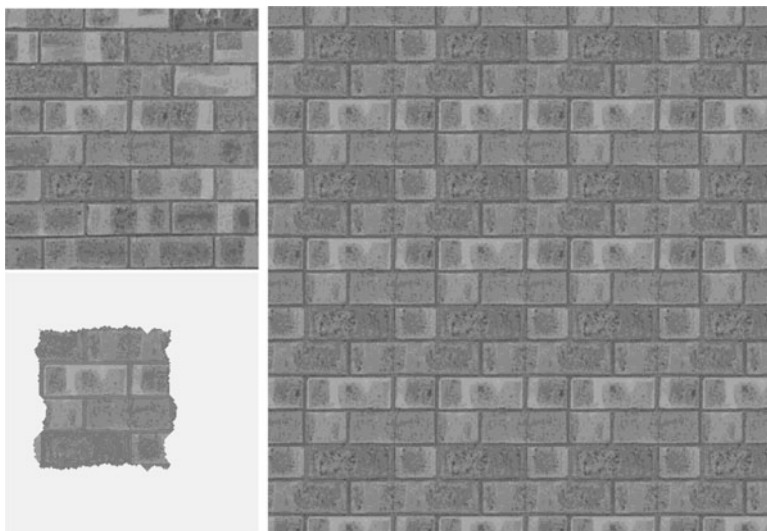


Fig. 3 Measured brick principal field (upper left), its optimal double toroidal patch (bottom left), and enlarged synthetic principal field (right, $K = 8$)



Fig. 4 Synthetic BTF-CMRF^{NPro3DCAR} enlarged color bark (right) estimated from their natural measurements (left)

Non-parametric Markov Random Field with Iterative Synthesis

The non-parametric principal random field \check{X} is estimated using simple K-means clustering of \check{Y} in the RGB color space into a predefined number of K classes, where cluster indices ω_i are $\check{X}_r \forall r \in I$ estimates. The clustering resulting thematic map is used to compute region size histograms \check{h}_i for all $i = 1, \dots, K$ classes. Let us order classes according to the decreasing number of pixels \check{n}_i belonging to each class, i.e., $\check{n}_1 \geq \check{n}_2 \geq \dots \geq \check{n}_K$. Histograms \check{h}_i are the only parameters required to store for the principal field.

Iterative Principal Field Synthesis

The iterative algorithm (Haindl and Havlíček 2018b) (Figs. 5 and 6) uses a data structure that describes membership in the region for each pixel. This data structure for each region additionally contains the class membership, size of the region and the requested number of regions of its size, all border pixels from both sides of the border, possibility to decrease or increase the region, and, for all classes, the histogram and regions, which can be increased or decreased. After any change in a pixel class assignment, this structure has to be updated.

0. The synthesized $M \times N$ required principal field is initialized to the largest class, and all histograms cells are rescaled using the scaling factor $\frac{MN}{\tilde{M}\tilde{N}}$, where $\tilde{M} \times \tilde{N}$ is the target (measured) texture size, i.e., $X_r^{(0)} = \omega_1 \forall r \in I$ and $\check{h}_i \rightarrow h_i$ for $i = 1, \dots, K$. A lattice multiindex r is randomly generated starting from the second-largest class ω_2 till the smallest size class ω_K . Class index X_r is changed to new value $X_r = \omega_i$ only if its previous value was $X_r = \omega_1$ and the total number of principal field pixels with class indicator ω_i is smaller than its final value n_i . After this initialization step, all classes have their correct required number of pixels but not yet their correct region size histograms.

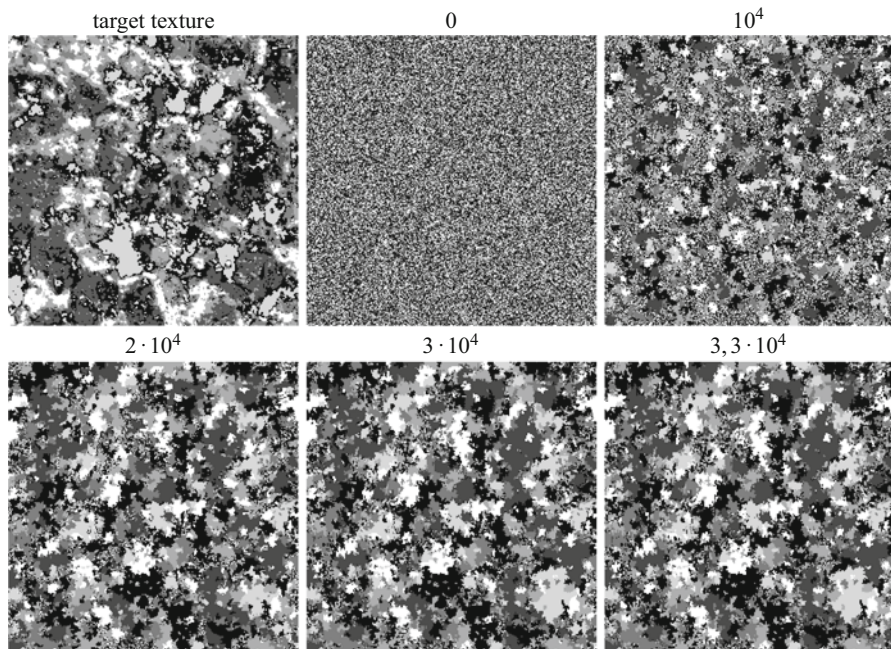


Fig. 5 The granite (Fig. 6) principal field synthesis. The target texture principal field, initialization, and selected iteration steps rightwards

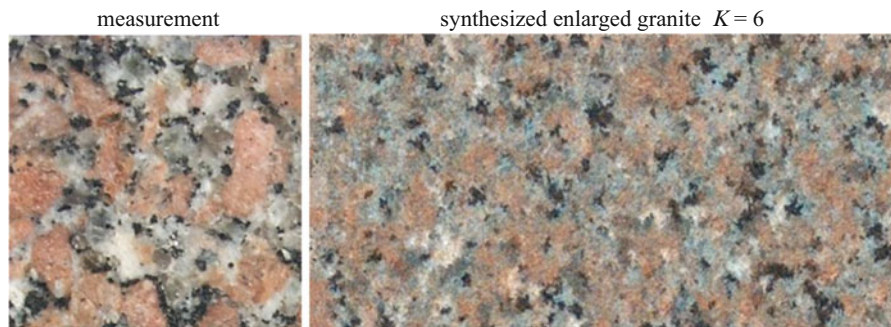


Fig. 6 The granite measurement and its synthetic enlargement (BTF-CMRF^{NPI3AR})

1. Pixels r and s are randomly selected with the following properties: The pixel r from the class ω_i is on the border between region $\downarrow \omega_i^A$ (a region A which can be decreased) and region $\uparrow \omega_j^B$ (a region B which can be increased). The pixel s from the class ω_j is on the border between region $\downarrow \omega_j^C$ (a region C which can be decreased) and region $\uparrow \omega_i^D$ (a region D which can be increased). These regions have to be distinct, i.e., $A \cap D = \emptyset$ and $B \cap C = \emptyset$. If such pixels r, s exist, go to step 5. If not repeat this step once more.

2. Gradually check all class couples starting from $\omega_1, \omega_2, \dots, \omega_K$ to find pixels r, s which meet conditions in step 1. All regions corresponding to the chosen classes, ω_i and ω_j , are selected randomly. If such pixels r, s exist, go to step 5.
3. Randomly select a region from class ω_i , which has two neighboring regions of class ω_j such as one can be decreased and another increased. If there exist two border pixels r, s in the region ω_j , where r is a border pixel with a region to be increased and s with a region to be decreased, go to step 5.
4. Gradually check all classes with incorrect histogram, starting from $\omega_1, \omega_2, \dots, \omega_K$; for every class ω_i gradually check all its regions $\uparrow \omega_i^A$ which can be increased; for each region $\uparrow \omega_i^A$, check every region neighboring border pixel r from class ω_j and region $\downarrow \omega_j^B$ (a region B which can be decreased), and find pixel s with the following properties: pixel s is from the class ω_i and region $\downarrow \omega_i^C$ (a region C which can be decreased), and pixel s is on the boarder of the region $\uparrow \omega_j^D$ from class ω_j (a region which can be increased). These regions have to be distinct, i.e., $A \cap C = \emptyset$ and $B \cap D = \emptyset$. If such pixels do not exist, go to step 7.
5. $X_r = \omega_j, X_s = \omega_i$ update the data structure.
6. If the number of iterations is less than a selected limit, go to 1.
7. Store the resulting principal field and stop.

Steps 1 and 2 allow simultaneous improvement of four regions, while step 3 improves two regions only. The algorithm converges to the correct class histograms $h_i \ i = 1, \dots, K$.

Non-parametric Markov Random Field with Fast Iterative Synthesis

The non-parametric principal field (Haindl and Havlíček 2018a) BTF-CMRF^{NPfi...} is estimated as in the previous section, and its synthesis is modified to be significantly faster at the cost of slightly compromised principal field variability. The fast algorithm compromise is its preference for convex regions instead of their general shapes but profits with faster convergence.

The median speed up between this method and the approach for the non-parametric principal field synthesis in section “[Non-parametric Markov Random Field with Iterative Synthesis](#)” is one-fifth of the required cycles to converge. Some textures (e.g., granite; Fig. 7) have sufficiently similar statistics of the synthesized regions with the principal target field already in the initialization step. Hence, the principal field synthesis even does not need any iterations. The lichen Fig. 8 principal target field (512×512) requires 29 137 iterations, while the previous iterative method needs nearly 5 times more (140 146) iterations to converge.

Iterative Principal Field Synthesis

The iterative algorithm is based on a similar data structure, which describes membership in the region for each pixel, as in the previous section. Both iterative algorithms differ only in their initialization steps.

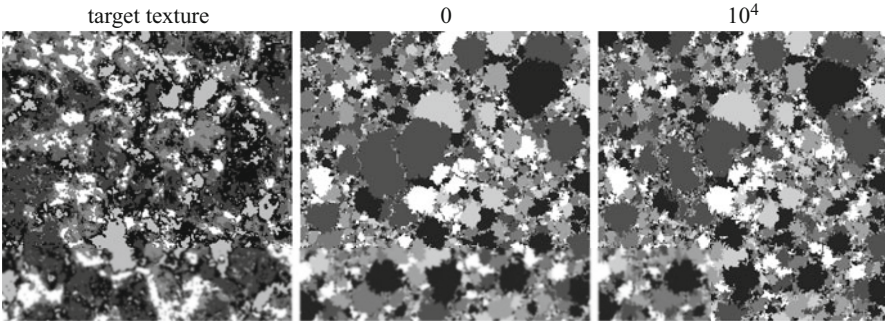


Fig. 7 The granite principal field synthesis. The target texture principal field, initialization, and a similar 10^4 -th iteration step result

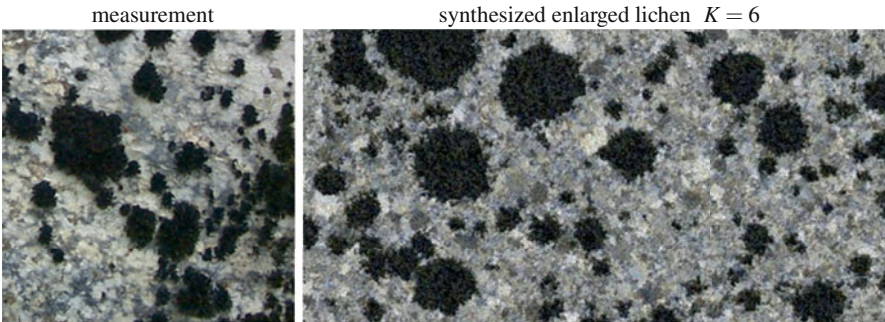


Fig. 8 The lichen measurement and its synthetic enlargement (BTF-CMRF^{NPfi3DCAR})

0. The synthesized $M \times N$ required principal field is initialized to the value ω_0 it means that pixel was not assigned to any class ω_i for $i = 1, \dots, K$. All histogram cells are rescaled using the scaling factor $\frac{MN}{\tilde{M}\tilde{N}}$, i.e., $X_r^{(0)} = \omega_1 \forall r \in I$ and $\tilde{h}_i \rightarrow h_i$ for $i = 1, \dots, K$. All regions from all classes $i = 1, \dots, K$ are sorted by region size. Starting from the biggest region A_1 till the smallest region A_M , where M the is number of all regions, a lattice multiindex r is randomly generated. The first pixel X_r of the region A_j where $j = 1, \dots, M$ and class ω_i is randomly selected and is changed to new value $X_r = \omega_i$ only if its previous value was $X_r = \omega_0$. All neighbors X_s of the pixel X_r which fulfil conditions $X_s = \omega_0$ and pixel X_s that has no neighbor from the class ω_i are added to the queue Q . Till the size of region A_j is higher than the number of actually added pixels, the next pixel X_r is randomly selected from the queue Q , the values are changed to $X_r = \omega_i$ and its neighbors are added to the queue Q if they meet the mentioned conditions. If the queue Q is empty and the size of the region A_j is higher than the number of actually assigned pixels, the rest of the pixels is randomly assigned to the class ω_i after the initialization of the last region A_M . After this initialization step,

all classes have their correct required number of pixels but not their correct region size histograms.

- 1.–7. Identical with the corresponding items in section “[Iterative Principal Field Synthesis](#)”.

Steps 1 and 2 allow simultaneous improvement of four regions, while step 3 improves two regions only. The algorithm converges to the correct class histograms $h_i \quad i = 1, \dots, K$.

Potts Markov Random Field

The resulting thematic principal map \check{X} BTF-CMRF^{2P...} is represented by the hierarchical two-scale Potts model (Haindl et al. 2012)

$$\check{X}^{(a)} = \frac{1}{Z^{(a)}} \exp \left\{ -\beta^{(a)} \sum_{s \in I_r} \delta_{X_r^{(a)} X_s^{(a)}} \right\} \quad (5)$$

where Z is the appropriate normalizing constant and $\delta()$ is the Kronecker delta function. The rough-scale-upper-level Potts model ($a = 1$) regions are further elaborated with the detailed fine-scale-level ($a = 2$) Potts model which models the corresponding subregions in each upper-level region. The parameter $\beta^{(a)}$ for both level models is estimated using an iterative estimator which starts from the upper β limit (β_{\max}) and adjusts (decreases or increases) its value until the Potts model regions have similar parameters (average inscribed squared region size and/or the region’s perimeter) with the target texture switching field. This iterative estimator gives more resembling results with the target texture than the alternative maximum pseudo-likelihood method (Levada et al. 2008). The corresponding Potts models are synthesized (Fig. 9 – middle) using the fast Swendsen-Wang sampling method (Swendsen and Wang 1987).

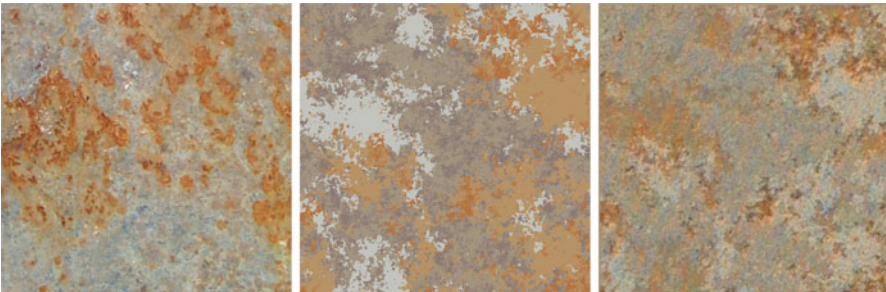


Fig. 9 The rusty plate texture measurement, its principal synthetic field, and the final synthetic CMRF^{P3AR} model texture

Potts-Voronoi Markov Random Field

The principal field (X) of the CMRF BTF-CMRF^{PV...} model (Haindl et al. 2015b) is a mosaic represented as a Voronoi diagram (Aurenhammer 1991), and the distribution of the particular colors (texture classes) of the mosaic is modeled as a Potts random field which is built on top of the adjacency graph (G) of the mosaic. Figure 10 illustrates this model applied to the floor mosaic, while Fig. 11 shows this model applied to a glass mosaic synthesis in St. Vitus Cathedral in Prague Castle. The algorithm requires input in the form of a segmented mosaic with distinguishable regions of the same texture type.

After that follows the identification of the mosaic field centers and the estimation of the parameters of the 2D discrete point process, which samples the control points of the newly synthesized Voronoi mosaic. This sampling is done using a 2D histogram, which has shown to be sufficient for the good quality estimate. The only other parameter is the number of points to be sampled, which grows linearly with the required area of the synthetic image in the case of texture enlargement applications.

With the control points for the Voronoi mosaic cells having been sampled, we compute the Voronoi diagram, and optionally mark the delimiting edges between adjacent cells. The assignment of a regional texture model to each mosaic cell (the principal MRF ($P(X | \check{Y})$)) is then mapped by the flexible K -state Potts random field (Potts and Domb 1952; Wu 1982).

Let us denote $G = (V, E)$ the adjacency graph of the mosaic areas and

$$N_u = \{v \in V : (u, v) \in E\}, \quad u \in V \quad (6)$$

the 1st-order neighborhood, where V, E are the vertex and edge sets. Vertexes correspond to the particular areas in the mosaic, and there is an edge between two vertexes if their corresponding areas are directly next to each other.

The resulting thematic principal map \check{X} is represented by the Potts model for a general graph

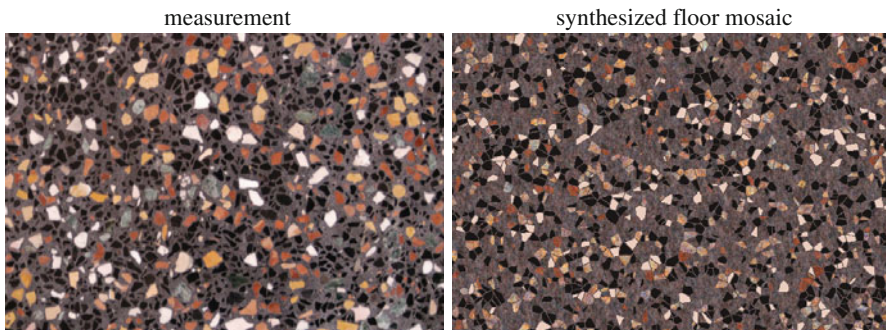


Fig. 10 The floor mosaic measurement and its synthesis (BTF-CMRF^{PV3DCAR})



Fig. 11 An example of St. Vitus Cathedral in Prague Castle stained glass window with two original panels (yellow arrows) replaced with synthetic images (BTF-CMRF^{PV3DCAR})

$$p(\check{X}|\beta) = \frac{1}{Z} \exp \left\{ -\beta \sum_{u \in V, v \in N_u} \delta(X_u, X_v) \right\} \tag{7}$$

where Z is the appropriate normalizing constant and $\delta()$ is the Kronecker delta function. The parameter β is estimated from the K-means clustered input mosaic using the maximum pseudo-likelihood method described by Levada et al. (2008). The local density of the Potts field can be expressed as

$$p(X_u = q | X_{v \in N_u}, \beta) = \frac{\exp \left\{ \beta \sum_{s \in N_u} \delta(q, X_s) \right\}}{\sum_{k=1}^K \exp \left\{ \beta \sum_{v \in N_u} \delta(k, X_v) \right\}} \tag{8}$$

for which the pseudo-likelihood approximation is

$$PL(\beta) = \prod_{u \in V} p(X_u = q | X_{v \in N_u}, \beta). \tag{9}$$

Calculating the logarithm, differentiating, and setting the result equal to 0, we get the maximum pseudo-likelihood equation (10) for the β estimate:

$$\Psi(\beta) = - \sum_{u \in V} \frac{\sum_{k=1}^K \left(\sum_{v \in N_u} \delta(X_u, X_v) \right) \exp \left\{ \beta \sum_{v \in N_u} \delta(k, X_v) \right\}}{\sum_{k=1}^K \exp \left\{ \beta \sum_{v \in N_u} \delta(k, X_v) \right\}} + \sum_{u \in V} \sum_{v \in N_u} \delta(X_u, X_v) = 0. \quad (10)$$

The corresponding Potts models are synthesized using the fast Swendsen-Wang sampling method (Swendsen and Wang 1987), although for smaller fields, which the mosaics undoubtedly are, other sampling MCMC methods such as the Gibbs sampler (Geman and Geman 1984) can be used. Alternatively, the Metropolis algorithm (Metropolis et al. 1953) should also work sufficiently fast enough.

Bernoulli Distribution Mixture Model

The distribution $P(X_{\{r\}})$ is assumed to be multivariable Bernoulli mixture (BM) (Haindl and Havlíček 2017b). The mixture distribution $P(X_{\{r\}})$ has the form

$$P(X_{\{r\}}) = \sum_{m \in \mathcal{M}} P(X_{\{r\}} | m) p(m) = \sum_{m \in \mathcal{M}} \prod_{s \in I_r} p_s(Y_s | m) p(m), \quad (11)$$

where \mathcal{M} is set of all mixture components, m a mixture component index, $\{r\}$ is a set of indices from I_r , and the principal field BTF-CMRF^{BM...} is further decomposed into separate binary bit planes of binary variables $\xi \in \mathcal{B}$, $\mathcal{B} = \{0, 1\}$ which are separately modeled and can be learned from much smaller training texture than a multi-level discrete mixture model (see examples in Fig. 14). We suppose that a bit factor of a principal field can be fully characterized by a marginal probability distribution of binary levels on pixels within the scope of a window centered around the location r and specified by the index set $I_r \subset I$, i. e., $X_{\{r\}} \in \mathcal{B}^{|I_r|}$ and $P(X_{\{r\}})$ is the corresponding marginal distribution of $P(X | \tilde{Y})$. The component distributions $P(\cdot | m)$ are factorizable, and multivariable Bernoulli

$$P(X_{\{r\}} | m) = \prod_{s \in I_r} \dot{\theta}_{m,s}^{X_s} (1 - \dot{\theta}_{m,s})^{1-X_s} \quad X_s \in X_{\{r\}}. \quad (12)$$

The mixture model parameters (11), (12) include component weights $p(m)$ and the univariate discrete distributions of binary levels. They are defined by one parameter $\dot{\theta}_{m,s}$ as a vector of probabilities:

$$p_s(\cdot | m) = (\dot{\theta}_{m,s}, 1 - \dot{\theta}_{m,s}). \quad (13)$$

The EM solution is (14), (15):

$$q^{(t)}(m | X_{\{r\}}) = \frac{p^{(t)}(m) P^{(t)}(X_{\{r\}} | m)}{\sum_{j \in \mathcal{M}} p^{(t)}(j) P^{(t)}(X_{\{r\}} | j)}, \tag{14}$$

$$p^{(t+1)}(m) = \frac{1}{|\mathcal{S}|} \sum_{X_{\{r\}} \in \mathcal{S}} q^{(t)}(m | X_{\{r\}}), \tag{15}$$

and

$$p_s^{(t+1)}(\xi | m) = \frac{1}{|\mathcal{S}| p^{(t+1)}(m)} \sum_{X_{\{r\}} \in \mathcal{S}} \delta(\xi, X_s) q^{(t)}(m | X_{\{r\}}), \quad \xi \in \mathcal{B}. \tag{16}$$

The total number of mixture (11), (13) parameters is thus $\dot{M}(1 + \eta) \dot{M} \in \mathcal{M}$ – confined to the appropriate norming conditions. The advantage of the multivariable Bernoulli model (13) is a simple switchover to any marginal distribution by deleting superfluous terms in the products $P(X_{\{r\}} | m)$.

Gaussian Mixture Model

The discrete principal field can be alternatively modeled (Haindl and Havlíček 2017b) by a continuous RF BTF-CMRF^{GM...} if we map single indices into continuous random variables with uniformly separated mean values and small variance. The synthesis results are subsequently inversely mapped back into a corresponding synthetic discrete principal field. We assume the joint probability distribution $P(X_{\{r\}}), X_{\{r\}} \in \mathcal{X}^\eta$ in the form of a normal mixture, and the mixture components are defined as products of univariate Gaussian densities

$$P(X_{\{r\}} | \mu_m, \sigma_m) = \prod_{s \in I_r} p_s(X_s | \mu_{ms}, \sigma_{ms}), \tag{17}$$

$$p_s(X_s | \mu_{ms}, \sigma_{ms}) = \frac{1}{\sqrt{2\pi} \sigma_{ms}} \exp \left\{ -\frac{(X_s - \mu_{ms})^2}{2\sigma_{ms}^2} \right\},$$

i. e., the components are multivariate Gaussian densities with diagonal covariance matrices. The maximum-likelihood estimates of the parameters $p(m), \mu_{ms}, \sigma_{ms}$ can be computed by the expectation-maximization (EM) algorithm (Dempster et al. 1977; Grim and Haindl 2003). Anew we use a data set \mathcal{S} obtained by pixel-wise shifting the observation window within the original texture image $\mathcal{S} = \{X_{\{r\}}^{(1)}, \dots, X_{\{r\}}^{(K)}\}, X_{\{r\}}^{(k)} \subset X$. The corresponding log-likelihood function is maximized by the EM algorithm ($m \in \mathcal{M}, n \in \mathcal{N}, X_{\{r\}} \in \mathcal{S}$), and the iterations are (14), (15) and

$$\mu_{m,n}^{(t+1)} = \frac{1}{\sum_{X_{\{r\}} \in \mathcal{S}} q^{(t)}(m | X_{\{r\}})} \sum_{X_{\{r\}} \in \mathcal{S}} X_n q(m | X_{\{r\}}), \quad (18)$$

$$(\sigma_{m,n}^{(t+1)})^2 = -(\mu_{m,n}^{(t+1)})^2 + \frac{\sum_{X_{\{r\}} \in \mathcal{S}} X_n^2 q^{(t)}(m | X_{\{r\}})}{\sum_{X_{\{r\}} \in \mathcal{S}} q(m | X_{\{r\}})}. \quad (19)$$

Local Markov and Mixture Models

While the principal models control the overall large-scale low-frequency textural structure, the local models synthesize the detail, regional and fine-granularity spatial-spectral BTF information. Once we have synthesized the required size's principal random field, using some of the previously described models, we use it to synthesize the local random part (3) of the BTF compound random model Y . This local model is a mosaic of K random field sub-models. These sub-models are assumed to be of the same type, but they differ in parameters and contextual support sets. This assumption is for simplicity only and is not restrictive because every sub-model is estimated and synthesized independently; thus, the Y mosaic can be easily composed of different types of random field models.

Local i -th texture region (not necessarily continuous) models are view and illumination dependent; thus, they need to be ideally represented by models which can be analytically estimated as well as easily non-iteratively synthesized (BTF-CMRF^{*N*ProI3DCAR} (Haindl and Havlíček 2010), BTF-CMRF^{*2P3DCAR*} (Haindl et al. 2012), BTF-CMRF^{*PV3DCAR*} (Haindl et al. 2015b), BTF-CMRF^{*c3DGM*} (Haindl and Havlíček 2016), BTF-CMRF^{*BM3DCAR*} (Haindl and Havlíček 2017b), BTF-CMRF^{*GM3DCAR*}, BTF-CMRF^{*N*ProI3DMA} (Haindl and Havlíček 2017a), BTF-CMRF^{*N*Pi3DCAR} (Haindl and Havlíček 2018b), BTF-CMRF^{*N*Pfi3DCAR} (Haindl and Havlíček 2018a)).

3D Causal Simultaneous Autoregressive Model

The 3D causal simultaneous autoregressive model (3DCAR) is an exceptional model because all its statistics can be solved analytically, and it can be utilized to build much more complex nD data models. For example, the 7D BTF models illustrated in Fig. 4 are composed from up to one hundred 3DCARs.

A digitized image Y is assumed to be defined on a finite rectangular $N \times M \times d$ lattice I , and $r = (r_1, r_2, r_3) \in I$ denotes a pixel multiindex with the row, columns, and spectral indices, respectively. The notation $I_r^c \subset I$ is a causal or unilateral neighborhood of pixel r , i.e.,

$$I_r^c \subset I_r^c = \{s : 1 \leq s_1 \leq r_1, 1 \leq s_2 \leq r_2, s \neq r\}.$$

The 3D causal simultaneous autoregressive model (3DCAR) is the wide-sense Markov model that can be written in the following regression equation form:

$$\tilde{Y}_r = \sum_{s \in I_r^c} A_s \tilde{Y}_{r-s} + e_r \quad \forall r \in I \tag{20}$$

where A_s are matrices (21) and the zero mean white Gaussian noise vector e_r has uncorrelated components with data indexed from I_r^c but noise vector components can be mutually correlated with a constant covariance matrix Σ .

$$A_{s_1, s_2} = \begin{pmatrix} a_{1,1}^{s_1, s_2}, \dots, a_{1,d}^{s_1, s_2} \\ \vdots, \ddots, \vdots \\ a_{d,1}^{s_1, s_2}, \dots, a_{d,d}^{s_1, s_2} \end{pmatrix} \tag{21}$$

where $d \times d$ are parameter matrices. The model can be expressed in the matrix form

$$Y_r = \gamma Z_r + e_r, \tag{22}$$

where

$$Z_r = [\tilde{Y}_{r-s}^T : \forall s \in I_r^c], \tag{23}$$

Z_r is a $d\eta \times 1$ vector, $\eta = \text{card}(I_r^c)$ and γ

$$\gamma = [A_1, \dots, A_\eta] \tag{24}$$

is a $d \times d\eta$ parameter matrix. To simplify notation the multiindexes r, s, \dots have only two components further on in this section.

An optimal support can be selected as the most probable model given past data

$$Y^{(r-1)} = \{Y_{r-1}, Y_{r-2}, \dots, Y_1, Z_r, Z_{r-1}, \dots, Z_1\},$$

i.e., $\max_j \{p(\mathcal{M}_j | Y^{(r-1)})\}$. Simultaneous conditional density can be evaluated analytically from

$$p(Y^{(r-1)} | \mathcal{M}_j) = \int \int p(Y^{(r-1)} | \gamma, \Sigma^{-1}) p(\gamma, \Sigma^{-1} | \mathcal{M}_j) d\gamma d\Sigma^{-1} \tag{25}$$

, and for the implemented uniform priors start, we get a decision rule (Haindl and Šimberová 1992):

The most probable AR model given past data $Y^{(r-1)}$, the normal-Wishart parameter prior and the uniform model prior is the model \mathcal{M}_i (Haindl 1983) for which

$$i = \arg \max_j \{D_j\}$$

$$D_j = -\frac{d}{2} \ln |V_{x(r-1)}| - \frac{\beta(r) - d\eta + d + 1}{2} \ln |\lambda_{(r-1)}| + \frac{d^2\eta}{2} \ln \pi \tag{26}$$

$$+ \sum_{i=1}^d \left[\ln \Gamma \left(\frac{\beta(r) - d\eta + d + 2 - i}{2} \right) - \ln \Gamma \left(\frac{\beta(0) - d\eta + d + 2 - i}{2} \right) \right]$$

where $V_{z(r-1)} = \tilde{V}_{z(r-1)} + V_{z(0)}$ with $\tilde{V}_{z(r-1)}$ defined in (31), $V_{z(0)}$ is an appropriate part of V_0 (31), $\beta(r)$ is defined in (27), (28) and $\lambda_{(r-1)}$ is (29).

The statistics (26) uses the following notation (27), (28), (29), (30) and (31):

$$\beta(r) = \beta(0) + r - 1 = \beta(r - 1) + 1, \tag{27}$$

$$\beta(0) > \eta - 2, \tag{28}$$

and

$$\lambda_{(r)} = V_{y(r)} - V_{zy(r)}^T V_{z(r)}^{-1} V_{zy(r)}. \tag{29}$$

$$V_{r-1} = \tilde{V}_{r-1} + V_0, \tag{30}$$

$$\tilde{V}_{r-1} = \begin{pmatrix} \sum_{k=1}^{r-1} \tilde{Y}_k \tilde{Y}_k^T & \sum_{k=1}^{r-1} \tilde{Y}_k \tilde{Z}_k^T \\ \sum_{k=1}^{r-1} \tilde{Z}_k \tilde{Y}_k^T & \sum_{k=1}^{r-1} \tilde{Z}_k \tilde{Z}_k^T \end{pmatrix} = \begin{pmatrix} \tilde{V}_{y(r-1)} & \tilde{V}_{zy(r-1)}^T \\ \tilde{V}_{zy(r-1)} & \tilde{V}_{z(r-1)} \end{pmatrix}. \tag{31}$$

Marginal densities $p(\gamma | Y^{(r-1)})$ and $p(\Sigma^{-1} | Y^{(r-1)})$ can be evaluated from (32), (33), respectively.

$$p(\gamma | Y^{(r-1)}) = \int p(\gamma, \Sigma^{-1} | Y^{(r-1)}) d\Sigma^{-1} \tag{32}$$

$$p(\Sigma^{-1} | Y^{(r-1)}) = \int p(\gamma, \Sigma^{-1} | Y^{(r-1)}) d\gamma \tag{33}$$

The marginal density $p(\Sigma^{-1} | Y^{(r-1)})$ is the Wishart distribution density (Haindl 1983)

$$p(\Sigma^{-1} | Y^{(r-1)}) = \frac{\pi^{\frac{d(1-d)}{4}} |\Sigma^{-1}|^{\frac{\beta(r)-d\eta}{2}}}{2^{\frac{d(\beta(r)-d\eta+d+1)}{2}} \prod_{i=1}^d \Gamma(\frac{\beta(r)-d\eta+2+d-i}{2})} |\lambda_{(r-1)}|^{\frac{\beta(r)-d\eta+d+1}{2}}$$

$$\exp \left\{ -\frac{1}{2} tr \{ \Sigma^{-1} \lambda_{(r-1)} \} \right\} \tag{34}$$

with

$$E \left\{ \Sigma^{-1} \mid Y^{(r-1)} \right\} = (\beta(r) - d\eta + d + 1) \lambda_{(r-1)}^{-1} \tag{35}$$

$$E \left\{ (\Sigma^{-1} - E\{\Sigma^{-1} \mid Y^{(r-1)}\})^T (\Sigma^{-1} - E\{\Sigma^{-1} \mid Y^{(r-1)}\}) \mid Y^{(r-1)} \right\} = \frac{2(\beta(r) - d\eta + 1)}{\lambda_{(r-1)} \lambda_{(r-1)}^T} \tag{36}$$

The marginal density $p(\gamma \mid Y^{(r-1)})$ is matrix t distribution density (Haindl 1983)

$$p(\gamma \mid Y^{(r-1)}) = \frac{\prod_{i=1}^d \Gamma(\frac{\beta(r)+d+2-i}{2})}{\prod_{i=1}^d \Gamma(\frac{\beta(r)-d\eta+d+2-i}{2})} \pi^{-\frac{d^2\eta}{2}} |\lambda_{(r-1)}|^{-\frac{d\eta}{2}} |V_{z(r-1)}|^{\frac{d}{2}} \left| I + \lambda_{(r-1)}^{-1} (\gamma - \hat{\gamma}_{r-1}) V_{z(r-1)} (\gamma - \hat{\gamma}_{r-1})^T \right|^{-\frac{\beta(r)+d+1}{2}} \tag{37}$$

with the mean value

$$E \left\{ \gamma \mid Y^{(r-1)} \right\} = \hat{\gamma}_{r-1} \tag{38}$$

and covariance matrix

$$E \left\{ (\gamma - \hat{\gamma}_{r-1})^T (\gamma - \hat{\gamma}_{r-1}) \mid Y^{(r-1)} \right\} = \frac{V_{z(r-1)}^{-1} \lambda_{(r-1)}}{\beta(r) - d\eta} \tag{39}$$

Similar statistics can be easily derived (Haindl 1983) for the alternative Jeffreys non-informative parameter prior. Similar to other model statistics, also the predictive density can be analytically derived.

The one-step-ahead predictive posterior density for the normal-Wishart parameter prior has the form of d-dimensional Student’s probability density (40) (Haindl 1983)

$$p(Y_r \mid Y^{(r-1)}) = \frac{\Gamma(\frac{\beta(r)-d\eta+d+2}{2})}{\Gamma(\frac{\beta(r)-d\eta+2}{2}) \pi^{\frac{d}{2}} (1 + Z_r^T V_{z(r-1)}^{-1} Z_r)^{\frac{d}{2}} |\lambda_{(r-1)}|^{\frac{1}{2}}} \left(1 + \frac{(Y_r - \hat{\gamma}_{r-1} Z_r)^T \lambda_{(r-1)}^{-1} (Y_r - \hat{\gamma}_{r-1} Z_r)}{1 + Z_r^T V_{z(r-1)}^{-1} Z_r} \right)^{-\frac{\beta(r)-d\eta+d+2}{2}}, \tag{40}$$

with $\beta(r) - d\eta + 2$ degrees of freedom; if $\beta(r) > d\eta$ then the conditional mean value is

$$E \left\{ Y_r \mid Y^{(r-1)} \right\} = \hat{\gamma}_{r-1} Z_r, \tag{41}$$

and

$$E \left\{ (Y_r - \hat{\gamma}_{r-1} Z_r)(Y_r - \hat{\gamma}_{r-1} Z_r)^T \mid Y^{(r-1)} \right\} = \frac{1 + Z_r V_{z^{(r-1)}}^{-1} Z_r^T}{(\beta(r) - d\eta)} \lambda_{(r-1)}. \tag{42}$$

The 3DCAR model can be made adaptive if we modify its recursive statistics using an exponential forgetting factor, i.e., a constant $\varphi \approx 0.99$. This forgetting factor smaller than 1 is used to weigh the influence of older data. The numerical stability of 3DCAR can be guaranteed if all its recursive statistics use the square root factor updating applying either the Cholesky or LDL^T decomposition (Haindl 2000), respectively.

The 3DCAR (analogously also the 2DCAR model) model has advantages in analytical solutions (Bayes, ML, or LS estimates) for $I_r, \hat{\gamma}, \hat{\sigma}^2, \hat{Y}_r$ statistics. It allows straightforward, fast synthesis, adaptivity, and building efficient recursive application algorithms.

3D Moving Average Model

Single multispectral texture factors Y are modeled using the extended version (3D MA) of the moving average model (Li et al. 1992; Haindl and Havlíček 2017a). A stochastic multispectral texture can be considered to be a sample from a 3D random field defined on an infinite 2D lattice. The model assumes that each factor is the output of an underlying system, which completely characterizes it in response to a 3D uncorrelated random input. This system can be represented by the impulse response of a linear 3D filter. The intensity values of the most significant pixels, together with their neighbors, are collected and averaged. The resultant 3D kernel is used as an estimate of the impulse response of the underlying system. A synthetic mono-spectral factor can be generated by convolving an uncorrelated 3D random field with this estimate. Suppose a stochastic multispectral texture denoted by Y is the response of an underlying linear system that completely characterizes the texture in response to a 3D uncorrelated random input E_r ; then, Y_r is determined by the difference equation

$$Y_r = \sum_{s \in I_r} B_s E_{r-s} \tag{43}$$

where B_s are constant matrix coefficients and $I_r \subset I$.

Hence, Y_r can be represented as $Y_r = h(r) * E_r$ where the convolution filter $h(r)$ contains all parameters B_s . In this equation, the underlying system behaves as a 3D filter, where we restrict the system impulse response to have significant values only

within a finite region. The geometry of I_r determines the causality or non-causality of the model.

The parameter estimation can be based on the modified random decrement technique (RDT) (Cole Jr 1973; Asmussen 1997). RDT assumes that the input is an uncorrelated random field. If every pixel component is higher than its corresponding threshold vector component and simultaneously at least one of its four neighbors is less than this threshold, the pixel is saved in the data accumulator. The procedure begins by selecting thresholds usually chosen as some percentage of the standard deviation of each spectral plane's intensities separately. In addition to that, a 3D MA model also requires to estimate the noise spectral correlation, i.e.,

$$\begin{aligned} E\{E_r E_s\} &= 0 & \forall r_1 \neq s_1 \vee r_2 \neq s_2, \\ E\{E_{r_1, r_2, r_3} E_{\bar{r}_1, \bar{r}_2, \bar{r}_3}\} &\neq 0 & \forall r_3 \neq \bar{r}_3. \end{aligned}$$

The synthetic factor can be generated simply by convolving an uncorrelated 3D RF E with the estimate of B according to (43). All generated factors form a new Gaussian pyramid. Fine resolution synthetic smooth texture is obtained by the collapse of the pyramid, i.e., an inverse procedure of that one creating the pyramid. This model can be used for materials which consist of several types of relatively small regions with fine-granular inner structure such as sand, grit, cork, lichen, or plaster. Figure 12 illustrates the visual quality of this simple model if the regional textures violate this fine-granularity assumption.

Spatial 3D Gaussian Mixture Model

A static homogeneous three-dimensional textural factor Y is assumed to be defined on a finite rectangular $M \times N \times d$ lattice I , $r = (r_1, r_2) \in I$ denotes a pixel multiindex with the row, columns, and indices, respectively. Let us suppose that Y represents a realization of a random vector with a probability distribution $P(Y)$. The statistical properties of interior pixels of the moving window on Y are translation invariant due to assumed textural homogeneity. They can be represented by a joint probability distribution, and the properties of the texture can be fully characterized

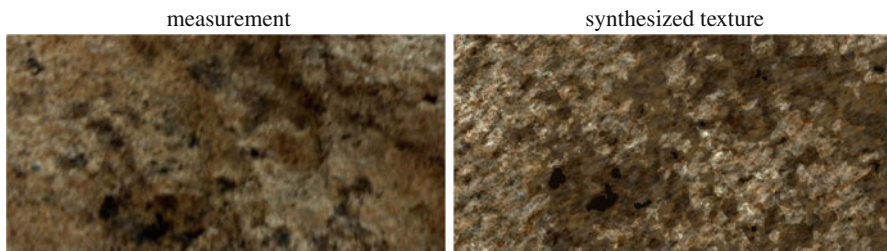


Fig. 12 The stone measurement and its synthesis (BTF-CMRF^{NP3DMA})

by statistical dependencies on a sub-field, i.e., by a marginal probability distribution of spectral levels on pixels within the scope of a window centered around the location r and specified by the index set:

$$I_r = \{r + s : |r_1 - s_1| \leq \alpha \wedge |r_2 - s_2| \leq \beta\} \subset I. \tag{44}$$

The index set I_r depends on modeled visual data and can have any other than this rectangular shape. $Y_{\{r\}}$ denotes the corresponding matrix containing all $d \times 1$ vectors Y_s in some fixed order arrangement such that $s \in I_r, Y_{\{r\}} = [Y_s \ \forall s \in I_r], Y_{\{r\}} \subset Y, \eta = \text{cardinality}\{I_r\}$, and $P(Y_{\{r\}})$ is the corresponding marginal distribution of $P(Y)$.

If we assume the joint probability distribution $P(Y_{\{r\}})$, in the form of a normal mixture (Haindl and Havlíček 2016)

$$\begin{aligned} P(Y_{\{r\}}) &= \sum_{m \in \mathcal{M}} p(m) P(Y_{\{r\}} | \mu_m, \Sigma_m) \quad Y_{\{r\}} \subset Y, \\ &= \sum_{m \in \mathcal{M}} p(m) \prod_{s \in I_r} p_s(Y_s | \mu_{m,s}, \Sigma_{m,s}) \end{aligned} \tag{45}$$

where $Y_{\{r\}} \in \mathfrak{R}^{d \times \eta}$ is $d \times \eta$ matrix, μ_m is $d \times \eta$ mean matrix, Σ_m is $d \times d \times \eta$ a covariance tensor, and $p(m)$ are probability weights and the mixture components are defined as products of multivariate Gaussian densities

$$P(Y_{\{r\}} | \mu_m, \Sigma_m) = \prod_{s \in I_{\{r\}}} p_s(Y_s | \mu_{ms}, \Sigma_{ms}), \tag{46}$$

$$p_s(Y_s | \mu_{ms}, \Sigma_{ms}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{m,s}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (Y_r - \mu_{m,s})^T \Sigma_{m,s}^{-1} (Y_r - \mu_{m,s}) \right\}, \tag{47}$$

i. e., the components are multivariate Gaussian densities with covariance matrices (53).

The underlying structural model of conditional independence is estimated from a data set \mathcal{S} obtained by the step-wise shifting of the contextual window I_r within the original textural image, i. e., for each location r one realization of $Y_{\{r\}}$.

$$\mathcal{S} = \{Y_{\{r\}} \ \forall r \in I, I_r \subset I\} \quad Y_{\{r\}} \in \mathfrak{R}^{d \times \eta}. \tag{48}$$

Parameter Estimation

The unknown parameters of the approximating mixture can be estimated using the iterative EM algorithm (Dempster et al. 1977). In order to estimate the unknown distributions $p_s(\cdot | m)$ and the component weights $p(m)$ we maximize the likelihood function (49) corresponding to the training set (48):

$$L = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} P(Y_{\{r\}} | \mu_m, \Sigma_m) p(m) \right]. \quad (49)$$

The likelihood is maximized using the iterative EM algorithm (with non-diagonal covariance matrices):

E:

$$q^{(t)}(m | Y_{\{r\}}) = \frac{\tilde{P}^{(t)}(Y_{\{r\}} | \mu_m, \Sigma_m) p^{(t)}(m)}{\sum_{j \in \mathcal{M}} P^{(t)}(Y_{\{r\}} | \mu_j, \Sigma_j) p^{(t)}(j)}, \quad (50)$$

M:

$$p^{(t+1)}(m) = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}}), \quad (51)$$

$$\begin{aligned} \mu_{m,s}^{(t+1)} &= \frac{1}{\sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})} \\ &\quad \sum_{Y_{\{r\}} \in \mathcal{S}} Y_s q^{(t)}(m | Y_{\{r\}}). \end{aligned} \quad (52)$$

The covariance matrices are

$$\begin{aligned} \Sigma_{m,s}^{(t+1)} &= \frac{\sum_{Y_{\{r\}} \in \mathcal{S}, Y_s \in Y_{\{r\}}} q^{(t)}(m | Y_{\{r\}}) (Y_s - \mu_{m,s}^{(t+1)})(Y_s - \mu_{m,s}^{(t+1)})^T}{\sum_{Y_r \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})} \quad (53) \\ &= \frac{\sum_{Y_{\{r\}} \in \mathcal{S}, Y_s \in Y_{\{r\}}} q^{(t)}(m | Y_{\{r\}}) Y_s Y_s^T}{\sum_{Y_r \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})} - \frac{p^{(t+1)}(m) |\mathcal{S}| \mu_{m,s}^{(t+1)} \left(\mu_{m,s}^{(t+1)} \right)^T}{\sum_{Y_r \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})}. \end{aligned}$$

The iteration process stops when the criterion increments are sufficiently small. The EM algorithm iteration scheme has the monotonic property $L^{(t+1)} \geq L^{(t)}$, $t = 0, 1, 2, \dots$ which implies the convergence of the sequence $\{L^{(t)}\}_0^\infty$ to a stationary point of the EM algorithm (local maximum or a saddle point of L). Figure 13 illustrates the usefulness of the BTF-CMRF^{3DGM} model for textile material modeling, while Fig. 18 shows this model applied to scratch restoration.

Applications

Numerous modeling applications can exploit the BTF models. The synthesis is beneficial not only for physically correct appearance modeling of surface materials under realistic and variable observation conditions (Figs. 15 and 17, upper row)

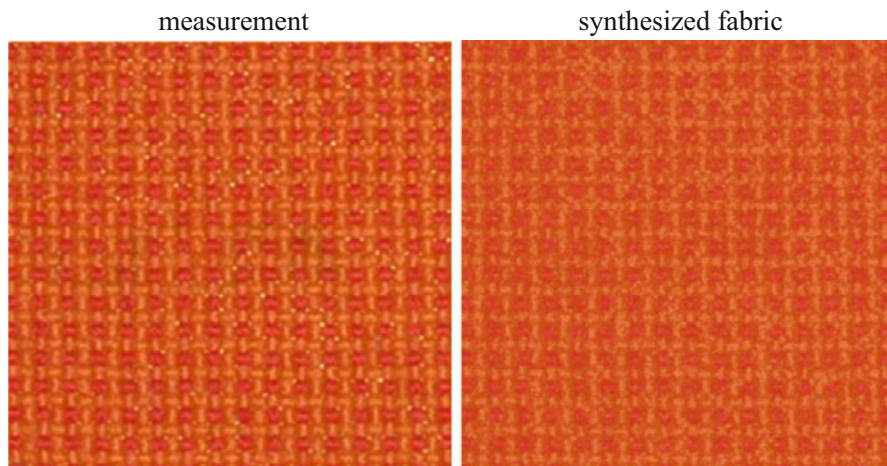


Fig. 13 The fabric measurement and its synthesis (BTF-CMRF^{3DGM})

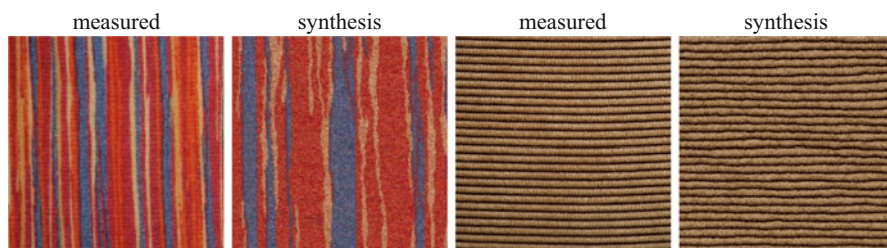


Fig. 14 Measured original cloth and corduroy materials and their synthesis using the $CRF^{BM-3CAR}$ model

but also for texture editing (Fig. 16), texture compression, or texture inpainting and restoration (Fig. 18). Various state-of-the-art unsupervised, semi-supervised, or supervised visual scene classification and understanding under variable observation conditions is the primary application for BTF analysis.

Texture Synthesis and Enlargement

Texture synthesis methods may be divided primarily into intelligent sampling and model-based methods (Fig. 14). They differ in need to store (sampling) or not (modeling) some actual texture measurements for new texture synthesis. Thus, even some methods which view texture as a stochastic process (Heeger and Bergen 1995; Efros and Leung 1999) still require to store an input exemplar. Sampling approaches De Bonet (1997), Efros and Leung (1999), Efros and Freeman (2001), Heeger and Bergen (1995), Xu et al. (2000), Dong and Chantler (2002), and Zelinka and Garland

(2002) rely on sophisticated sampling from real texture measurements, while the model-based techniques (Kashyap 1981; Haindl 1991; Haindl and Havlíček 1998, 2000; Bennett and Khotanzad 1998, 1999; Gimelfarb 1999; Paget and Longstaff 1998; Zhu et al. 2000) describe texture data using multidimensional mathematical models, and their synthesis is based on the estimated model parameters only. The mathematical model-based synthesis has an advantage in the possibility of seamless texture enlargement to any size (e.g., Fig. 6). The enlargement of a restricted texture measurement is always required in any application but cannot be achieved with sampling approaches without visible seams or repetitions.

The BTF modeling's ultimate aim is to create a visual impression of the same material without a pixel-wise correspondence to the finding condition model of the original measurements. Figure 15 shows the finding condition model of the beautiful gothic style relief (around 1370) of the Christ in Gethsemane (Prague) in the right and restored condition to a possible original appearance in the left.

The cornerstone of our BTF compression and modeling methods is the replacement of a vast number of original BTF measurements by their efficient parametric estimates derived from an underlying set of spatial probabilistic models and thus to allow a huge BTF compression ratio unattainable by any alternative sampling-based BTF synthesis method. Simultaneously these models can be used to reconstruct missing parts of the BTF measurement space or the controlled BTF space editing (Haindl and Havlíček 2009, 2012; Haindl et al. 2015b) by changing some of the model's parameters.

Textures without significant low frequencies such as Fig. 14-corduroy or Fig. 13-fabric can be modeled using simple local models only, either Markovian or mixtures such as 3DCAR, 3DMA, 3DBM, 3DGM, etc. Textures with substantial low frequencies (Figs. 4, 9, 14-cloth) will benefit from a compound version of the BTF model. Non-BTF textures can approximate low frequencies using a multiscale version of these models, e.g., pyramidal model (Haindl and Filip 2013).

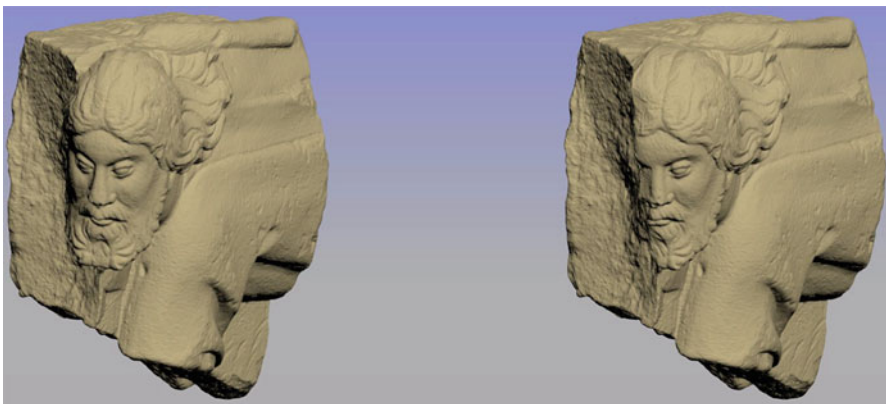
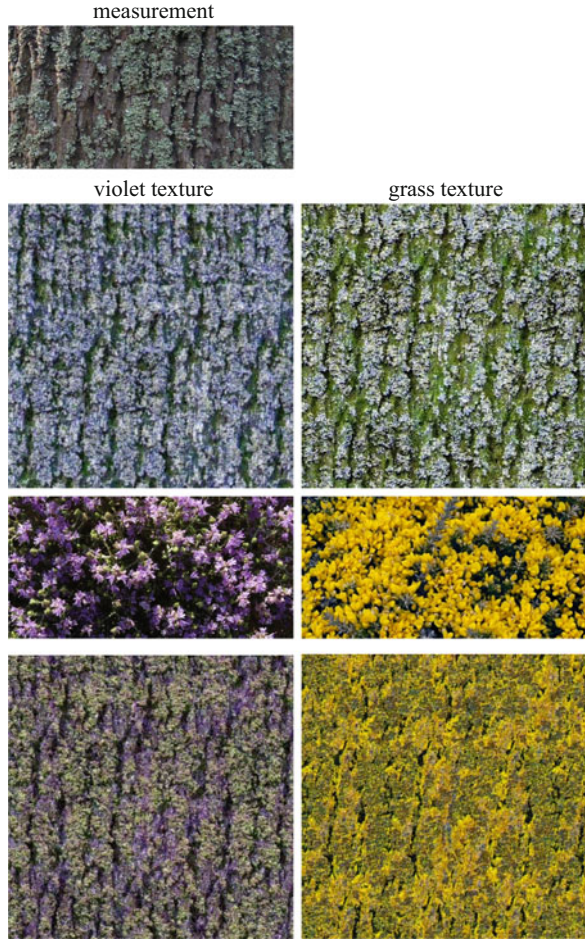


Fig. 15 3D model of the beautiful gothic style relief of the Christ in Gethsemane, Prague (finding condition model right, restored condition to a possible original appearance left) mapped with the BTF synthetic plaener using the $CMRF^{3CAR}$ model

Fig. 16 Synthetic BTF-CMRF^{NProI3DCAR} edited and enlarged maple bark texture (second and fourth rows) with single sub-models estimated from their natural measurements (maple bark first and flowers third row)



The 3DCAR model is synthesized directly from its predictor (41) and Gaussian noise generator (22), (39). The advantage of a mixture model is its simple synthesis based on the marginals:

$$p_{n|\rho}(Y_n | Y_{\{\rho\}}) = \sum_{m=1}^{\dot{M}} W_m(Y_{\{\rho\}}) p_n(Y_n | m), \tag{54}$$

where $W_m(Y_{\{\rho\}})$ are the a posterior component weights corresponding to the given sub-matrix $Y_{\{\rho\}} \subset Y_{\{r\}}$:

$$W_m(Y_{\{\rho\}}) = \frac{p(m)P_\rho(Y_{\{\rho\}} | m)}{\sum_{j=1}^{\dot{M}} p(j)P_\rho(Y_{\{\rho\}} | j)}, \tag{55}$$

$$P_{\rho}(Y_{\{\rho\}} | m) = \prod_{n \in \rho} p_n(Y_n | m). \quad (56)$$

There are several alternatives for the 3DGM model synthesis (Haindl et al. 2011) (Fig. 13). The unknown multivariate vector-levels Y_n can be synthesized by random sampling from the conditional density (54), or the mixture RF can be approximated using the GM mixture prediction.

Texture Compression

BTF – the best current measurable representation of a material appearance – requires tens of thousands of images using a sophisticated high-precision automatic measuring device. Such measurements result in a massive amount of data that can easily reach tens of terabytes for a single measured material. Nevertheless, these data have still insufficient spatial extent for any real virtual reality applications and have to be further enlarged using advanced modeling techniques. The resulting BTF size excludes its direct rendering in graphical applications, and compression of these huge BTF data spaces is inevitable. The usual car interior model requires more than 20 of such demanding BTF material measurements, and a similar problem holds for other applications of the physically correct appearance modeling such as computer games or film animations. A related problem is measurement data storage because storage technology is still the weak link, lagging behind recent developments in data sensing technologies. The apparent solution is mathematical modeling which allows replacing massive measured data with few thousand parameters and thus to reach tremendous unbeatable appearance data compression apart from unlimited seamless material texture enlargement. For example, the compression ratio relative to our BTF measurements is up to 1 : 1000000.

Texture Editing

Material-appearance editing is a practical approach with vast potential for significant speedup and cost reduction in industrial virtual prototyping or various design applications. An editing process can simulate materials for which no direct measurements are available or not existing in Nature (Fig. 16). Another example of the edited texture is two panels with the artificial but fitting glass mosaic synthesis in St. Vitus Cathedral in Prague Castle stained glass window on Fig. 11. Such edited artifacts allow an artist to test several possible design alternatives or model defunct monuments.

Illumination Invariants

Textures are essential clues to specify objects present in a visual scene. However, the appearance of natural textures is highly illumination and view angle-dependent. As a

consequence, the most recent realistic texture-based classification or segmentation methods require multiple training images (Varma and Zisserman 2005) captured under all possible illumination and viewing conditions for each class. Such learning is clumsy, probably expensive, and very often even impossible if required measurements are not available.

If we assume fixed positions of viewpoint and illumination sources, uniform illumination sources, and Lambertian surface reflectance, then two images \tilde{Y}, Y acquired with different illumination spectra can be linearly transformed to each other:

$$\tilde{Y}_r = B Y_r \quad \forall r. \quad (57)$$

It is possible to show (Vacha and Haindl 2007) that assuming (57) the following 3DCAR model-derived features are illumination invariant:

1. trace: $\text{trace } A_m, m = 1, \dots, \eta K$
2. eigenvalues: $v_{m,j}$ of $A_m, m = 1, \dots, \eta K, j = 1, \dots, C$
3. $1 + X_r^T V_x^{-1} X_r,$
4. $\sqrt{\sum_r (Y_r - \hat{\gamma} X_r)^T \lambda^{-1} (Y_r - \hat{\gamma} X_r)},$
5. $\sqrt{\sum_r (Y_r - \mu)^T \lambda^{-1} (Y_r - \mu)},$

where μ is the mean value of the vector Y_r .

Above textural features derived from the 3DCAR model are robust to illumination direction changes, invariant to illumination brightness and spectrum changes, and simultaneously also robust to Gaussian noise degradation. We extensively verified this property on the BTF texture measurements, where illumination sources are spanned over 75% of possible illumination half-sphere. Figure 17 illustrates the application of 3DCAR model-derived features are illumination invariants to the unsupervised wood mosaic segmentation.

(Un)supervised Image Recognition

Unsupervised or supervised texture segmentation is the prerequisite for successful content-based image retrieval, scene analysis, automatic acquisition of virtual models, quality control, security, medical applications, and many others.

Similarly, robust surface material recognition requires the BTF data learning set. We classified 65 wood species measured in the BTF representation in the study Mikeš and Haindl (2019) using the state-of-the-art convolutional neural network (TensorFlow library (Google 2019; Krizhevsky 2009; Krizhevsky et al. 2012; Pattanayak 2017)). We documented (Mikeš and Haindl 2019) sharp classification accuracy decrease when using standard texture recognition approach, i.e., small

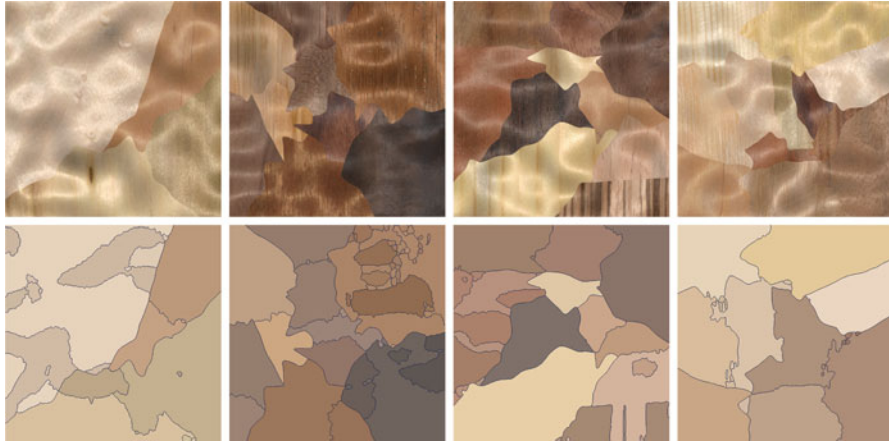


Fig. 17 BTF wood mosaic and the MW3-AR⁸_i model-based (Haindl et al. 2015a) unsupervised segmentation results

learning set size and the vertical viewing and illumination angle, which is a very inadequate representation of the enormous material appearance variability.

Although plentiful different methods were already published (Zhang 1997), the image recognition problem is still far from being solved. This situation is among others due to missing reliable performance comparison between different techniques. Only limited results were published (Martin et al. 2001; Sharma and Singh 2001; Ojala et al. 2002; Haindl and Mikeš 2008) on suitable quantitative measures that allow us to evaluate and compare the quality of segmentation algorithms.

Spatial interaction models and especially Markov random field-based models are increasingly popular for texture representation (Kashyap 1986; Reed and du Buf 1993; Haindl 1991), etc. Several researchers dealt with the difficult problem of unsupervised segmentation using these models, see for example Panjwani and Healey (1995), Manjunath and Chellapa (1991), Andrey and Tarroux (1998), Haindl (1999), and Matuszak and Schreiber (2009).

Our unsupervised segmenters (Haindl and Mikeš 2004, 2005, 2006; Haindl et al. 2015a) assume the multispectral or multi-channel textures to be locally represented by the parameters (Θ_r) of the multidimensional random field models possibly recursively evaluated for each pixel and several scales. The segmentation part of the algorithm is then based on the underlying Gaussian mixture model ($p(\Theta_r) = \sum_{i=1}^K p_i p(\Theta_r | v_i, \Sigma_i)$) representing the Markovian parametric space and starts with an over-segmented initial estimation, which is adaptively modified until the optimal number of homogeneous mammogram segments is reached. The corresponding mixture model equations ($p(\Theta_r)$, $p(\Theta_r | v_i, \Sigma_i)$) are solved using a modified EM algorithm (Haindl and Mikeš 2007).

The concept of decision fusion for high-performance pattern recognition is well known and widely accepted in the area of supervised classification, where (often very diverse) classification technologies, each providing complementary sources of information about class membership, can be integrated to provide more accurate, robust, and reliable classification decisions than single-classifier applications. Our method (Haindl and Mikeš 2007) circumvents the problem of multiple unsupervised segmenter combination by fusing multiple-processed measurements into a single segmenter feature vector.

Multispectral/Multi-channel Image Restoration

Physical imaging, processing or transmission systems, and a recording medium are imperfect, and thus a recorded image represents a degraded version of the original scene.

The image restoration task is to recover an unobservable image given the observed corrupted image \check{Y} with respect to some statistical criterion. Image restoration is a busy research area for already several decades, and many restoration algorithms have been proposed (Andrews and Hunt 1977; Geman and Geman 1984; Acton and Bovik 1999; Loubes and Rochet 2009; Felsberg 2009; Burgeth et al. 2009; Polzehl and Tabelow 2009).

The image degradation is often supposed to be approximated by the linear degradation model:

$$\check{Y}_r = \sum_{s \in I_r} f_s Y_{r-s} + e_r \quad (58)$$

where f is a discrete representation of the unknown point-spread function. The point-spread function can be non-homogeneous, but we assume its slow changes relative to the size of an image. I_r is some contextual support set, and the degradation noise e is uncorrelated with the unobservable image. The point-spread function is unknown but such that we can assume the unobservable image Y to be reasonably well approximated by the expectation of the corrupted image

$$\hat{Y} = E\{\check{Y}\} \quad (59)$$

in regions with gradual pixel value changes.

Let us approximate after having observed $\check{Y}^{(j-1)} = \{\check{Y}_{j-1}, \dots, \check{Y}_1\}$ the mean value $\hat{Y}_j = E\{\check{Y}_j\}$ by the $E\{\check{Y}_j | \check{Y}^{(j-1)} = \check{y}^{(j-1)}\}$ where $\check{y}^{(j-1)}$ are known past realization for j . Thus, we suppose that all other possible realizations $\check{y}^{(j-1)}$ than the true past pixel values have negligible probabilities. This assumption implies conditional expectations approximately equal to unconditional ones, i.e.,

$$E\{\check{Y}_j\} \approx E\{\check{Y}_j | \check{Y}^{(j-1)}\}, \quad (60)$$

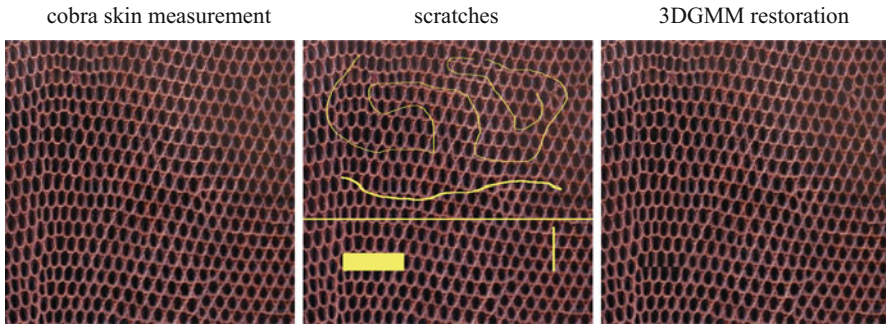


Fig. 18 Cobra skin scratch restoration using the spatial 3D Gaussian mixture model

and assuming the noisy image \check{Y} can be represented by a 3DCAR model, then the restoration model as well as the local estimation of the point-spread function leads to a fast analytical solution (Haindl 2002). A similar restoration approach can also be derived for a multi-channel (Haindl and Šimberová 2002) or multitemporal (Haindl and Šimberová 2005) image restoration problems typically caused by random fluctuations originating mostly in the Earth's atmosphere during ground-based telescope observations.

A difficult restoration problem is to restore missing parts of an image or a spatially correlated data field. For example, every movie deteriorates with usage and time irrespective of any care it gets. Movies (on both optical and magnetic materials) suffer from blotches, dirt, sparkles, noise, scratches (Fig. 18), missing or heavily corrupted frames, mold, flickering, jittering, image vibrations, and other problems. For each kind of defect, usually a different kind of restoration algorithm is needed. The scratch notion means every coherent region with missing data (simultaneously in all spectral bands) in a color movie frame (Haindl and Filip 2002), static image, range map, radio-spectrograph (Haindl and Šimberová 1996), radar observation, color textures (Haindl and Havlíček 2015), etc. These missing data restoration methods (inpainting) exploit correlations in the spatial/spectral/temporal data space and benefit from the discussed Markovian or mixture (Fig. 18) random field models.

Conclusion

There is no single universal BTF model applicable for physically correct modeling of visual properties of all possible BTF textures. Every presented model is better suited for some subspace of possible BTF textures, either natural or artificial. Their selection depends primarily on their spectral and spatial frequency content as well as on available learning data. We present exceptional adaptive 3D Markovian or mixture models, either solved analytically or iteratively and quickly synthesized.

The presented compound Markovian models are rare exceptions in the Markovian model family that allow deriving extraordinarily efficient and fast data process-

ing algorithms. All their statistics can be either evaluated recursively, and they either do not need any Monte Carlo sampling typical for other Markovian models or can use a fast form of such sampling (Potts random field). The 3DCAR models have an advantage over non-causal (3DAR) in their analytical treatment. It is possible to find the analytical solution of model parameters, optimal model support, model predictor, etc. Similarly, the 3DCAR model synthesis is straightforward, and this model can be directly generated from the model equation.

The mixture models are capable of reducing additive noise and restore missing textural parts simultaneously. They produce high-quality results, especially of regular or near-regular color textures. Their typical drawback the extensive learning data set requirement is lessened by the ample available BTF measurement space using a transfer learning approach.

The BTF-CMRF models offer a large data compression ratio (only tens of parameters per BTF), easy simulation, and fast, seamless synthesis of any required texture size. The methods have no restriction to the number of spectral channels; thus, they can be easily applied to hyperspectral BTFs. The methods can be easily generalized for color or BTF texture editing by estimating some local models from different target materials or for image restoration or inpainting.

The Markovian models can be used for image enhancement, e.g., utterly automatic mammogram enhancement, multispectral and multiresolution texture qualitative measures development, or image or video segmentation. Some of these models also allow robust textural features for texture classification when learning and classified textures differ in scale. The classifiers based on Markovian features can exploit illumination or geometric invariance properties and often outperform the state-of-the-art alternative methods on tested public databases (e.g., eye, bark, needles, textures).

Acknowledgments The Czech Science Foundation Project GAČR 19-12340S supported this research.

References

- Acton, S., Bovik, A.: Piecewise and local image models for regularized image restoration using cross-validation. *IEEE Trans. Image Process.* **8**(5), 652–665 (1999)
- Aittala, M., Weyrich, T., Lehtinen, J.: Practical SVBRDF capture in the frequency domain. *ACM Trans. Graph. (Proc. SIGGRAPH)* **32**(4), 110:1–110:13 (2013)
- Aittala, M., Weyrich, T., Lehtinen, J.: Two-shot SVBRDF capture for stationary materials. *ACM Trans. Graph.* **34**(4), 110:1–110:13 (2015). <https://doi.org/10.1145/2766967>
- Aittala, M., Aila, T., Lehtinen, J.: Reflectance modeling by neural texture synthesis. *ACM Trans. Graph.* **35**(4), 65:1–65:13 (2016). <https://doi.org/10.1145/2897824.2925917>
- Anderson, T.W.: *The Statistical Analysis of Time Series*. Wiley, New York (1971)
- Andrews, H.C., Hunt, B.: *Digital Image Restoration*. Prentice-Hall, Englewood Cliffs (1977)
- Andrey, P., Tarroux, P.: Unsupervised segmentation of markov random field modeled textured images using selectionist relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3), 252–262 (1998)

- Asmussen, J.C.: Modal analysis based on the random decrement technique: application to civil engineering structures. PhD thesis, University of Aalborg (1997)
- Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv. (CSUR)* **23**(3), 345–405 (1991)
- Baril, J., Boubekeur, T., Gioia, P., Schlick, C.: Polynomial wavelet trees for bidirectional texture functions. In: *SIGGRAPH'08: ACM SIGGRAPH 2008 talks*, p. 1. ACM, New York (2008). <https://doi.org/10.1145/1401032.1401072>
- Bennett, J., Khotanzad, A.: Multispectral random field models for synthesis and analysis of color images. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3), 327–332 (1998)
- Bennett, J., Khotanzad, A.: Maximum likelihood estimation methods for multispectral random field image models. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(6), 537–543 (1999)
- Broemeling, L.D.: *Bayesian Analysis of Linear Models*. Marcel Dekker, New York (1985)
- Burgeth, B., Pizarro, L., Didas, S., Weickert, J.: Coherence-enhancing diffusion filtering for matrix fields. In: *Locally Adaptive Filtering in Signal and Image Processing*. Springer, Berlin (2009)
- Cole Jr, H.A.: On-line failure detection and damping measurement of aerospace structures by random decrement signatures. Technical Report TMX-62.041, NASA (1973)
- Dana, K.J., Nayar, S.K., van Ginneken, B., Koenderink, J.J.: Reflectance and texture of real-world surfaces. In: *CVPR*, pp. 151–157. IEEE Computer Society (1997)
- De Bonet, J.: Multiresolution sampling procedure for analysis and synthesis of textured images. In: *ACM SIGGRAPH 97*, pp. 361–368. ACM Press (1997)
- Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: *Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*, pp. 145–156 (2000)
- Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. B* **39**(1), 1–38 (1977)
- Deschaintre, V., Aittala, M., Durand, F., Drettakis, G., Bousseau, A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Trans. Graph.* **37**(4), 1–15 (2018). <https://doi.org/10.1145/3197517.3201378>
- Dong, J., Chantler, M.: Capture and synthesis of 3D surface texture. In: *Texture 2002*, vol. 1, pp. 41–45. Heriot-Watt University (2002)
- Dong, J., Wang, R., Dong, X.: Texture synthesis based on multiple seed-blocks and support vector machines. In: *2010 3rd International Congress on Image and Signal Processing (CISP)*, vol. 6, pp. 2861–2864 (2010). <https://doi.org/10.1109/CISP.2010.5646815>
- Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Fiume, E. (ed.) *ACM SIGGRAPH 2001*, pp. 341–346. ACM Press (2001). citeseer.nj.nec.com/efros01image.html
- Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: *Proceedings of International Conference on Computer Vision (2)*, Corfu, pp. 1033–1038 (1999). citeseer.nj.nec.com/efros99texture.html
- Felsberg, M.: Adaptive filtering using channel representations. In: *Locally Adaptive Filtering in Signal and Image Processing*. Springer, Berlin (2009)
- Filip, J., Haindl, M.: Bidirectional texture function modeling: a state of the art survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(11), 1921–1940 (2009). <https://doi.org/10.1109/TPAMI.2008.246>
- Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions and bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intel.* **6**(11), 721–741 (1984)
- Gimelfarb, G.: *Image Textures and Gibbs Random Fields*. Kluwer Academic Publishers, Dordrecht (1999)
- Google (2019) Tensorflow. Technical report, Google AI, <http://www.tensorflow.org/>
- Grim, J., Haindl, M.: Texture modelling by discrete distribution mixtures. *Comput. Stat. Data Anal.* **41**(3–4), 603–615 (2003)
- Haindl, M.: Identification of the stochastic differential equation of the type arma. PhD thesis, ÚTIA Czechoslovak Academy of Sciences, Prague (1983)
- Haindl, M.: Texture synthesis. *CWI Q.* **4**(4), 305–331 (1991)

- Haindl, M.: Texture segmentation using recursive Markov random field parameter estimation. In: Bjarne, K.E., Peter, J. (eds.) Proceedings of the 11th Scandinavian Conference on Image Analysis, Pattern Recognition Society of Denmark, Lyngby, pp. 771–776 (1999). <http://citeseer.ist.psu.edu/305262.html>; <http://www.ee.surrey.ac.uk/Research/VSSP/3DVision/virtuous/Publications/Haindl-SCIA99.ps.gz>
- Haindl, M.: Recursive square-root filters. In: Sanfeliu, A., Villanueva, J., Vanrell, M., Alquezar, R., Jain, A., Kittler, J. (eds.) Proceedings of the 15th IAPR International Conference on Pattern Recognition, vol. II, pp. 1018–1021. IEEE Press, Los Alamitos (2000). <https://doi.org/10.1109/ICPR.2000.906246>
- Haindl, M.: Recursive model-based colour image restoration. Lect. Notes Comput. Sci. (2396), 617–626 (2002)
- Haindl, M., Filip, J.: Fast restoration of colour movie scratches. In: Kasturi, R., Laurendeau, D., Suen, C. (eds.) Proceedings of the 16th International Conference on Pattern Recognition, vol. III, pp. 269–272. IEEE Computer Society, Los Alamitos (2002). <https://doi.org/10.1109/ICPR.2002.1047846>
- Haindl, M., Filip, J.: Extreme compression and modeling of bidirectional texture function. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1859–1865 (2007). <https://doi.org/10.1109/TPAMI.2007.1139>
- Haindl, M., Filip, J.: Visual Texture. Advances in Computer Vision and Pattern Recognition. Springer, London (2013). <https://doi.org/10.1007/978-1-4471-4902-6>
- Haindl, M., Hatka, M.: BTF Roller. In: Chantler, M., Drbohlav, O. (eds.) Texture 2005. Proceedings of the 4th International Workshop on Texture Analysis, pp. 89–94. IEEE, Los Alamitos (2005a)
- Haindl, M., Hatka, M.: A roller – fast sampling-based texture synthesis algorithm. In: Skala, V. (ed.) Proceedings of the 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 93–96. UNION Agency – Science Press, Plzen (2005b)
- Haindl, M., Havlíček, V.: Multiresolution colour texture synthesis. In: Dobrovodský, K. (ed.) Proceedings of the 7th International Workshop on Robotics in Alpe-Adria-Danube Region, pp. 297–302. ASCO Art, Bratislava (1998)
- Haindl, M., Havlíček, V.: A multiresolution causal colour texture model. Lect. Notes Comput. Sci. (1876), 114–122 (2000)
- Haindl, M., Havlíček, V.: Texture editing using frequency swap strategy. In: Jiang, X., Petkov, N. (eds.) Computer Analysis of Images and Patterns. Lecture Notes in Computer Science, vol. 5702, pp. 1146–1153. Springer (2009). https://doi.org/10.1007/978-3-642-03767-2_139
- Haindl, M., Havlíček, V.: A compound MRF texture model. In: Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010, pp. 1792–1795. IEEE Computer Society CPS, Los Alamitos (2010). <https://doi.org/10.1109/ICPR.2010.442>
- Haindl, M., Havlíček, V.: A plausible texture enlargement and editing compound markovian model. In: Salerno, E., Cetin, A., Salvetti, O. (eds.) Computational Intelligence for Multimedia Understanding. Lecture Notes in Computer Science, vol. 7252, pp. 138–148. Springer, Berlin/Heidelberg (2012). https://doi.org/10.1007/978-3-642-32436-9_12, <http://www.springerlink.com/content/047124j43073m202/>
- Haindl, M., Havlíček, V.: Color Texture Restoration, pp. 13–18. IEEE, Piscataway (2015). <https://doi.org/10.1109/ICCIS.2015.7274540>
- Haindl, M., Havlíček, V.: Three-dimensional gaussian mixture texture model. In: The 23rd International Conference on Pattern Recognition (ICPR), pp. 2026–2031. IEEE (2016). <https://doi.org/978-1-5090-4846-5/16/protect/TI/textdollar31.0>, <http://www.icpr2016.org/site/>
- Haindl, M., Havlíček, M.: A compound moving average bidirectional texture function model. In: Zgrzynowa, A., Choros, K., Sieminski, A. (eds.) Multimedia and Network Information Systems, Advances in Intelligent Systems and Computing, vol. 506, pp. 89–98. Springer International Publishing (2017a). https://doi.org/10.1007/978-3-319-43982-2_8
- Haindl, M., Havlíček, V.: Two compound random field texture models. In: Beltrán-Castañón, C., Nyström, I., Famili, F. (eds.) 2016 the 21st IberoAmerican Congress on Pattern Recognition (CIARP 2016). Lecture Notes in Computer Science, vol. 10125, pp. 44–51. Springer International Publishing AG, Cham (2017b). https://doi.org/10.1007/978-3-319-52277-7_6

- Haindl, M., Havlíček, V.: BTF compound texture model with fast iterative non-parametric control field synthesis. In: di Baja, G.S., Gallo, L., Yetongnon, K., Dipanda, A., Castrillon-Santana, M., Chbeir, R. (eds.) Proceedings of the 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS 2018), pp. 98–105. IEEE Computer Society CPS, Los Alamitos (2018a). <https://doi.org/10.1109/SITIS.2018.00025>
- Haindl, M., Havlíček, V.: BTF compound texture model with non-parametric control field. In: The 24th International Conference on Pattern Recognition (ICPR 2018), pp. 1151–1156. IEEE (2018b). <http://www.icpr2018.org/>
- Haindl, M., Mikeš, S.: Model-based texture segmentation. *Lect. Notes Comput. Sci.* (3212), 306–313 (2004)
- Haindl, M., Mikeš, S.: Colour texture segmentation using modelling approach. *Lect. Notes Comput. Sci.* (3687), 484–491 (2005)
- Haindl, M., Mikeš, S.: Unsupervised texture segmentation using multispectral modelling approach. In: Tang, Y., Wang, S., Yeung, D., Yan, H., Lorette, G. (eds.) Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, vol. II, pp. 203–206. IEEE Computer Society, Los Alamitos (2006). <https://doi.org/10.1109/ICPR.2006.1148>
- Haindl, M., Mikeš, S.: Unsupervised texture segmentation using multiple segmenters strategy. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. Lecture Notes in Computer Science, vol. 4472, pp. 210–219. Springer (2007). https://doi.org/10.1007/978-3-540-72523-7_22
- Haindl, M., Mikeš, S.: Texture segmentation benchmark. In: Lovell, B., Laurendeau, D., Duin, R. (eds.) Proceedings of the 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4. IEEE Computer Society, Los Alamitos (2008). <https://doi.org/10.1109/ICPR.2008.4761118>
- Haindl, M., Šimberová, S.: A multispectral image line reconstruction method. In: Theory & Applications of Image Analysis. Series in Machine Perception and Artificial Intelligence, pp. 306–315. World Scientific, Singapore (1992). https://doi.org/10.1142/9789812797896_0028
- Haindl, M., Šimberová, S.: A high – resolution radiospectrograph image reconstruction method. *Astron. Astrophys.* **115**(1), 189–193 (1996)
- Haindl, M., Šimberová, S.: Model-based restoration of short-exposure solar images. In: Abraham, A., Ruiz-del Solar, J., Koppen, M. (eds.) Soft Computing Systems Design, Management and Applications, pp. 697–706. IOS Press, Amsterdam (2002)
- Haindl, M., Šimberová, S.: Restoration of multitemporal short-exposure astronomical images. *Lect. Notes Comput. Sci.* (3540), 1037–1046 (2005)
- Haindl, M., Mikeš, S., Pudil, P.: Unsupervised hierarchical weighted multi-segmenter. In: Benediktsson, J., Kittler, J., Roli, F. (eds.) Lecture Notes in Computer Science. MCS 2009, vol. 5519, pp. 272–282. Springer (2009a). https://doi.org/10.1007/978-3-642-02326-2_28
- Haindl, M., Mikeš, S., Vácha, P.: Illumination invariant unsupervised segmenter. In: Bayoumi, M. (ed.) IEEE 16th International Conference on Image Processing – ICIP 2009, pp. 4025–4028. IEEE (2009b). <https://doi.org/10.1109/ICIP.2009.5413753>
- Haindl, M., Havlíček, V., Grim, J.: Probabilistic mixture-based image modelling. *Kybernetika* **46**(3), 482–500 (2011). <http://www.kybernetika.cz/content/2011/3/482/paper.pdf>
- Haindl, M., Remeš, V., Havlíček, V.: Potts compound markovian texture model. In: Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, pp. 29–32. IEEE Computer Society CPS, Los Alamitos (2012)
- Haindl, M., Mikeš, S., Kudo, M.: Unsupervised surface reflectance field multi-segmenter. In: Azzopardi, G., Petkov, N. (eds.) Computer Analysis of Images and Patterns. Lecture Notes in Computer Science, vol. 9256, pp. 261–273. Springer International Publishing (2015a). https://doi.org/10.1007/978-3-319-23192-1_22
- Haindl, M., Remeš, V., Havlíček, V.: BTF Potts Compound Texture Model, vol. 9398, pp. 939807–1–939807–11. SPIE, Bellingham (2015b). <https://doi.org/10.1117/12.2077481>
- Han, J.Y., Perlin, K.: Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Trans. Graph.* **22**(3), 741–748 (2003)
- Heeger, D., Bergen, J.: Pyramid based texture analysis/synthesis. In: ACM SIGGRAPH 95, pp. 229–238. ACM Press (1995)

- Holroyd, M., Lawrence, J., Zickler, T.: A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Trans. Graph. (Proc. SIGGRAPH 2010)* (2010). <http://www.cs.virginia.edu/~mjh7v/Holroyd10.php>
- Kashyap, R.: Analysis and synthesis of image patterns by spatial interaction models. In: Kanal, L., Rosenfeld, A. (eds.) *Progress in Pattern Recognition 1*. Elsevier, North-Holland (1981)
- Kashyap, R.: Image models. In: Young, T.Y., Fu, K.S. (eds.) *Handbook of Pattern Recognition and Image Processing*. Academic, New York (1986)
- Koudelka, M.L., Magda, S., Belhumeur, P.N., Kriegman, D.J.: Acquisition, compression, and synthesis of bidirectional texture functions. In: *Texture 2003: Third International Workshop on Texture Analysis and Synthesis, Nice*, pp. 59–64 (2003)
- Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, University of Toronto (2009)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
- Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.* **22**(3), 277–286 (2003)
- Levada, A., Mascarenhas, N., Tannus, A.: Pseudolikelihood equations for potts mrf model parameter estimation on higher order neighborhood systems. *Geosci. Remote Sens. Lett. IEEE* **5**(3), 522–526 (2008). <https://doi.org/10.1109/LGRS.2008.920909>
- Li, X., Cadzow, J., Wilkes, D., Peters, R., Bodruzzaman II, M.: An efficient two dimensional moving average model for texture analysis and synthesis. In: *Proceedings IEEE Southeastcon'92*, vol. 1, pp. 392–395. IEEE (1992)
- Liang, L., Liu, C., Xu, Y.Q., Guo, B., Shum, H.Y.: Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph. (TOG)* **20**(3), 127–150 (2001)
- Liu, F., Picard, R.: Periodicity, directionality, and randomness: wold features for image modeling and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(7), 722–733 (1996). <https://doi.org/10.1109/34.506794>
- Loubes, J., Rochet, P.: Regularization with approximated L^2 maximum entropy method. In: *Locally Adaptive Filtering in Signal and Image Processing*. Springer, Berlin (2009)
- Manjunath, B., Chellapa, R.: Unsupervised texture segmentation using Markov random field models. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 478–482 (1991)
- Marschner, S.R., Westin, S.H., Arbre, A., Moon, J.T.: Measuring and modeling the appearance of finished wood. *ACM Trans. Graph.* **24**(3), 727–734 (2005)
- Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of 8th International Conference on Computer Vision*, vol. 2, pp. 416–423 (2001). <http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>
- Matuszak, M., Schreiber, T.: Locally specified polygonal Markov fields for image segmentation. In: *Locally Adaptive Filtering in Signal and Image Processing*. Springer, Berlin (2009)
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953)
- Mikeš, S., Haindl, M.: View dependent surface material recognition. In: *Bebis, G., Boyle, R., Parvin, B., Koračin, D., Ushizima, D., Chai, S., Sueda, S., Lin, X., Lu, A., Thalmann, D., Wang, C., Xu, P. (eds.) 14th International Symposium on Visual Computing (ISVC 2019)*. Lecture Notes in Computer Science, vol. 11844, pp. 156–167. Springer Nature Switzerland AG (2019). https://doi.org/10.1007/978-3-030-33720-9_12, <https://www.isvc.net/>
- Müller, G., Meseth, J., Klein, R.: Compression and real-time rendering of measured BTFs using local PCA. In: *Vision, Modeling and Visualisation 2003*, pp. 271–280 (2003)
- Müller, G., Meseth, J., Sattler, M., Sarlette, R., Klein, R.: Acquisition, synthesis and rendering of bidirectional texture functions. In: *Eurographics 2004, STAR – State of The Art Report*, Eurographics Association, pp. 69–94 (2004)
- Neubeck, A., Zalesny, A., Gool, L.: 3D texture reconstruction from extensive BTF data. In: *Chantler, M., Drbohlav, O. (eds.) Texture 2005*. Heriot-Watt University, Edinburgh (2005)

- Ngan, A., Durand, F.: Statistical acquisition of texture appearance. In: Eurographics Symposium on Rendering, Eurographics (2006)
- Ojala, T., Maenpaa, T., Pietikainen, M., Viertola, J., Kyllonen, J., Huovinen, S.: Outex: new framework for empirical evaluation of texture analysis algorithms. In: International Conference on Pattern Recognition, pp. 1:701–706 (2002)
- Paget, R., Longstaff, I.D.: Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Trans. Image Process.* **7**(8), 925–932 (1998)
- Panjwani, D., Healey, G.: Markov random field models for unsupervised segmentation of textured color images. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(10), 939–954 (1995)
- Pattanayak, S.: Pro Deep Learning with TensorFlow. Apress (2017). <https://doi.org/10.1007/978-1-4842-3096-1>
- Polzehl, J., Tabelow, K.: Structural adaptive smoothing: principles and applications in imaging. In: Locally Adaptive Filtering in Signal and Image Processing. Springer, Berlin (2009)
- Portilla, J., Simoncelli, E.: A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vis.* **40**(1), 49–71 (2000)
- Potts, R., Domb, C.: Some generalized order-disorder transformations. In: Proceedings of the Cambridge Philosophical Society, vol. 48, pp. 106–109 (1952)
- Praun, E., Finkelstein, A., Hoppe, H.: Lapped textures. In: ACM SIGGRAPH 2000, pp. 465–470 (2000)
- Rainer, G., Ghosh, A., Jakob, W., Weyrich, T.: Unified neural encoding of BTFs. In: Computer Graphics Forum, vol. 39, pp. 167–178. Wiley Online Library (2020)
- Reed, T.R., du Buf, J.M.H.: A review of recent texture segmentation and feature extraction techniques. *CVGIP–Image Underst.* **57**(3), 359–372 (1993)
- Ren, P., Wang, J., Snyder, J., Tong, X., Guo, B.: Pocket reflectometry. *ACM Trans. Graph. (Proc. SIGGRAPH)* **30**(4) (2011). <https://doi.org/10.1145/2010324.1964940>
- Ruiters, R., Schwartz, C., Klein, R.: Example-based interpolation and synthesis of bidirectional texture functions. In: Computer Graphics Forum, vol. 32, pp. 361–370. Wiley Online Library (2013)
- Sattler, M., Sarlette, R., Klein, R.: Efficient and realistic visualization of cloth. In: Eurographics Symposium on Rendering (2003)
- Schwartz, C., Sarlette, R., Weinmann, M., Rump, M., Klein, R.: Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the university of bonn. *Sensors* **14**(5), 7753–7819 (2014). <https://doi.org/10.3390/s140507753>. <http://www.mdpi.com/1424-8220/14/5/7753>
- Sharma, M., Singh, S.: Minerva scene analysis benchmark. In: Seventh Australian and New Zealand Intelligent Information Systems Conference, pp. 231–235. IEEE (2001)
- Soler, C., Cani, M., Angelidis, A.: Hierarchical pattern mapping. *ACM Trans. Graph.* **21**(3), 673–680 (2002)
- Swendsen, R.H., Wang, J.S.: Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* **58**(2), 86–88 (1987). <https://doi.org/10.1103/PhysRevLett.58.86>
- Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Trans. Graph. (TOG)* **21**(3), 665–672 (2002)
- Tsai, Y.T., Shih, Z.C.: K-clustered tensor approximation: a sparse multilinear model for real-time rendering. *ACM Trans. Graph.* **31**(3), 19:1–19:17 (2012). <https://doi.org/10.1145/2167076.2167077>
- Tsai, Y.T., Fang, K.L., Lin, W.C., Shih, Z.C.: Modeling bidirectional texture functions with multivariate spherical radial basis functions. *Pattern Anal. Mach. Intell. IEEE Trans.* **33**(7), 1356–1369 (2011). <https://doi.org/10.1109/TPAMI.2010.211>
- Vacha, P., Haindl, M.: Image retrieval measures based on illumination invariant textural mrf features. In: CIVR'07: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 448–454. ACM Press, New York (2007). <https://doi.org/10.1145/1282280.1282346>
- Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. *Int. J. Comput. Vis.* **62**(1–2), 61–81 (2005)

- Wang, J., Dana, K.: Relief texture from specularities. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(3), 446–457 (2006)
- Wei, L., Levoy, M.: Texture synthesis using tree-structure vector quantization. In: *ACM SIGGRAPH 2000*, pp. 479–488. ACM Press/Addison Wesley/Longman (2000). citeseer.nj.nec.com/wei01texture.html
- Wei, L., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: *SIGGRAPH 2001*, pp. 355–360. ACM (2001)
- Wu, F.: (1982) The Potts model. *Rev. Modern Phys.* **54**(1), 235–268
- Wu, H., Dorsey, J., Rushmeier, H.: A sparse parametric mixture model for BTF compression, editing and rendering. *Comput. Graph. Forum* **30**(2), 465–473 (2011)
- Xu, Y., Guo, B., Shum, H.: Chaos mosaic: fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-32, Redmont (2000)
- Zelinka, S., Garland, M.: Towards real-time texture synthesis with the jump map. In: *13th European Workshop on Rendering*, p. 99104 (2002)
- Zelinka, S., Garland, M.: Interactive texture synthesis on surfaces using jump maps. In: Christensen, P., Cohen-Or, D. (eds.) *14th European Workshop on Rendering, Eurographics (2003)*
- Zhang, Y.J.: Evaluation and comparison of different segmentation algorithms. *Pattern Recogn. Lett.* **18**, 963–974 (1997)
- Zhang, J.D., Zhou, K., Velho ea, L.: Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Trans. Graph.* **22**(3), 295–302 (2003)
- Zhu, S., Liu, X., Wu, Y.: Exploring texture ensembles by efficient Markov Chain Monte Carlo – toward a “trichromacy” theory of texture. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(6), 554–569 (2000)



Regularization of Inverse Problems by Neural Networks

29

Markus Haltmeier and Linh Nguyen

Contents

Introduction	1066
Ill-Posedness	1067
Data-Driven Reconstruction	1068
Outline	1069
Preliminaries	1069
Right Inverses	1070
Regularization Methods	1072
Deep Learning	1074
Regularizing Networks	1075
Null-Space Networks	1076
Convergence Analysis	1078
Extensions	1079
The NETT Approach	1080
Learned Regularization Functionals	1080
Convergence Analysis	1082
Related Methods	1088
Conclusion and Outlook	1090
References	1091

Abstract

Inverse problems arise in a variety of imaging applications, including computed tomography, non-destructive testing, and remote sensing. Characteristic features of inverse problems are the non-uniqueness and instability of their solutions.

M. Haltmeier (✉)

Department of Mathematics, University of Innsbruck, Innsbruck, Austria
e-mail: markus.haltmeier@uibk.ac.at

L. Nguyen

Department of Mathematics, University of Idaho, Moscow, ID, USA
e-mail: lnguyen@uidaho.edu

© Springer Nature Switzerland AG 2023

K. Chen et al. (eds.), *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, https://doi.org/10.1007/978-3-030-98661-2_81

1065

Therefore, any reasonable solution method requires the use of regularization tools that select specific solutions and, at the same time, stabilize the inversion process. Recently, data-driven methods using deep learning techniques and neural networks showed to significantly outperform classical solution methods for inverse problems. In this chapter, we give an overview of inverse problems and demonstrate the necessity of regularization concepts for their solution. We show that neural networks can be used for the data-driven solution of inverse problems and review existing deep learning methods for inverse problems. In particular, we view these deep learning methods from the perspective of regularization theory, the mathematical foundation of stable solution methods for inverse problems. This chapter is more than just a review as many of the presented theoretical results extend existing ones.

Keywords

Inverse problems · Deep learning · Neural networks · Regularization theory · Ill-posedness · Stability · Theoretical foundation

Introduction

The solution of inverse problems arises in a variety of practically important applications, including medical imaging, computer vision, geophysics, as well as many other branches of pure and applied sciences. Inverse problems are most efficiently formulated as an estimation problem of the form

$$\text{recover } \mathbf{x}^* \in \mathbb{X} \text{ from data } \mathbf{y} = \mathbf{A}(\mathbf{x}^*) + \xi \in \mathbb{Y}. \quad (1)$$

Here, $\mathbf{A} : \mathbb{X} \rightarrow \mathbb{Y}$ is a mapping between normed spaces, $\mathbf{x}^* \in \mathbb{X}$ is the true unknown solution, \mathbf{y} represents the given data, and ξ is an unknown data perturbation. In this context, the application of the operator \mathbf{A} is referred to as the forward operator or forward problem, and solving (1) is the corresponding inverse problem. In the absence of noise where $\xi = 0$, we refer to $\mathbf{y} = \mathbf{A}(\mathbf{x}^*)$ as exact data, and in the case where $\xi \neq 0$, we refer to \mathbf{y} as noisy data.

One of the prime examples of inverse problems are image reconstruction problems, where the forward operator describes the data generation process depending on the image reconstruction modality. For example, in X-ray computed tomography (CT), the forward operator is the sampled Radon transform, whereas in magnetic resonance imaging (MRI), the forward operator is the sampled Fourier transform. Reconstructing the diagnostic image from experimentally collected data leads to solving an inverse problem of the form (1). In these and other applications, the underlying forward operator is naturally formulated on infinite-dimensional spaces, because the object to be reconstructed is a function of a continuous spatial variable. Even though the numerical solution is performed in a finite dimensional

discretization, the mathematical properties of the continuous formulation are crucial for understanding and improving image formation algorithms.

III-Posedness

The inherent character of inverse problems is their ill-posedness. This means that even in the case of exact data, the solution of (1) is either not unique, not existent, or does not stably depend on the given data. More formally, for an inverse problem, at least one of the following three unfavorable properties holds:

- (I1) NON-UNIQUENESS: For some $\mathbf{x}_1^* \neq \mathbf{x}_2^* \in \mathbb{X}$, we have $\mathbf{A}(\mathbf{x}_1^*) = \mathbf{A}(\mathbf{x}_2^*)$.
- (I2) NON-EXISTENCE: For some $\mathbf{y} \in \mathbb{Y}$, the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has no solution.
- (I3) INSTABILITY: Smallness of $\|\mathbf{A}(\mathbf{x}_1^*) - \mathbf{A}(\mathbf{x}_2^*)\|$ does not imply smallness of $\|\mathbf{x}_1^* - \mathbf{x}_2^*\|$.

These conditions imply that the forward operator does not have a continuous inverse, which could be used to directly solve (1). Instead, regularization methods have to be applied, which result in stable methods for solving inverse problem.

Regularization methods approach the ill-posedness by two steps. First, to address non-uniqueness and non-existence issues (I1), (I2), one restricts the image and pre-image space of the forward operator to sets $\mathbb{M} \subseteq \mathbb{X}$ and $\text{ran}(\mathbf{A}) \subseteq \mathbb{Y}$, such that the restricted forward operator $\mathbf{A}_{\text{res}}: \mathbb{M} \rightarrow \text{ran}(\mathbf{A})$ becomes bijective. For any exact data, the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ then has a unique solution in \mathbb{M} , which is given by the inverse of the restricted forward operator applied to \mathbf{y} . Second, in order to address the instability issue (I3), in a second step, one considers a family of continuous operators $\mathbf{B}_\alpha: \mathbb{Y} \rightarrow \mathbb{X}$ for $\alpha > 0$ that converge to $\mathbf{A}_{\text{res}}^{-1}$ in a suitable sense; see section “Preliminaries” for precise definitions.

Note that the choice of the set \mathbb{M} is crucial as it represents the class of desired reconstructions and acts as selection criteria for picking a particular solution of the given inverse problem. The main challenge is that this class is actually unknown or at least it cannot be described properly. For example, in CT for medical imaging, the set of desired solutions represents the set of all functions corresponding to spatially attenuation inside patients, a function class that is clearly challenging, if not impossible, to describe in simple mathematical terms.

Variational regularization and variants (Scherzer et al. 2009) have been the most successful class of regularization methods for solving inverse problems. Here, \mathbb{M} is defined as solutions having a small value of a certain regularization functional that can be interpreted as a measure for the deviation from the desired solutions. Various regularization functionals have been analyzed for inverse problems, including Hilbert space norms (Engl et al. 1996), total variation (Acar and Vogel 1994), and sparse ℓ^q -penalties (Daubechies et al. 2004; Grasmair et al. 2008). Such handcrafted regularization functionals have limited complexity and are unlikely to accurately model complex signal classes arising in applications such as medical imaging. On the other hand, their regularization effects are well understood, efficient numerical

algorithms have been developed for their realization, they work reasonably well in practice, and they have been rigorously analyzed mathematically.

Data-Driven Reconstruction

Recently data-driven methods based on neural networks and deep learning demonstrated to significantly outperform existing variational and iterative reconstruction algorithms for solving inverse problems. The essential idea is to use neural networks to define a class $(\mathbf{R}_\theta)_{\theta \in \Theta}$ of reconstruction networks $\mathbf{R}_\theta: \mathbb{Y} \rightarrow \mathbb{X}$ and to select the parameter vector $\theta \in \Theta$ of the network in a data-driven manner. The selection is based on a set of training data $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)$, where $\mathbf{x}_i \in \mathbb{M}$ are desired reconstructions and $\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i^*) + \xi_i \in \mathbb{Y}$ are corresponding data. Even if the set \mathbb{M} of desired reconstructions is unknown, the available samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ can be used to select the particular reconstruction method. A typical selection strategy is to minimize a penalized least-squares functional having the form

$$\theta^* \in \arg \min_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{R}_\theta(\mathbf{y}_i)\|^2 + P(\theta) \right\}. \quad (2)$$

The final neural network-based reconstruction method is then given by $\mathbf{R}_{\theta^*}: \mathbb{Y} \rightarrow \mathbb{X}$ and is such that in average, it performs well on the given training dataset.

Existing deep learning-based methods include post-processing networks (Han et al. 2016; Jin et al. 2017), null-space networks (Schwab et al. 2019, 2020), variational networks (Kobler et al. 2017), iterative networks (Yang et al. 2016; Adler and Öktem 2017; Aggarwal et al. 2018), network cascades (Kofler et al. 2018; Schlemper et al. 2017), and learned regularization functional (Li et al. 2020; Lunz et al. 2018; Obmann et al. 2020b). We refer to the review Arridge et al. (2019) for other data-driven reconstruction methods such as GANs (Bora et al. 2017; Mardani et al. 2018), dictionary learning, deep basis pursuit (Sulam et al. 2019), or deep image priors (Ulyanov et al. 2018; Van Veen et al. 2018; Dittmer et al. 2020), which we do not touch in this chapter. Post-processing networks and null-space networks are explicit, where the reconstruction network is given explicitly and its parameters are trained to fit the given training data. Methods using learned regularizers are implicit, and the reconstruction network $\mathbf{R}_\theta(\mathbf{y}) = \arg \min \mathcal{T}_{\theta, \mathbf{y}}$ is defined by minimizing a properly trained Tikhonov functional $\mathcal{T}_{\theta, \mathbf{y}}: \mathbb{X} \rightarrow [0, \infty]$. Variational networks and iterative networks are in between, where $\arg \min \mathcal{T}_{\theta, \mathbf{y}}$ is approximated via an iterative scheme using L steps.

Any reasonable method for solving an inverse problem, including all learned reconstruction schemes, has to include some form of regularization. However, regularization may be imposed implicitly, even without noticing by the researcher developing the algorithm. Partially, this is the case because discretization, early stopping, or other techniques to numerically stabilizing an optimization algorithm

at the same time have a regularization effect on the underlying inverse problem. Needless to say, understanding and analyzing where exactly the regularization effect comes from will increase the reliability of any algorithm and allows its further improvement. In conclusion, any data-driven reconstruction method has to include either explicitly or implicitly a form of regularization. In this chapter, we will analyze the regularization properties of various deep learning methods for solving inverse problems.

Outline

The outline of this chapter is as follows. In section “[Preliminaries](#)”, we present the background of inverse problems and deep learning. In section “[Regularizing Networks](#)”, we analyze direct neural network-based reconstructions, whereas in section “[The NETT Approach](#)”, we study variational and iterative reconstruction methods based on neural networks. The chapter concludes with a discussion and some final remarks given in section “[Conclusion and Outlook](#)”. While the concepts presented in the subsequent sections are known, most of the presented results extend existing ones. Therefore, this chapter is much more than just a review over existing results.

For the sake of clarity, in this chapter, we focus on linear inverse problems; even several results can be extended non-linear problems as well. We will provide remarks pointing to such results. Throughout the study, we allow an infinite-dimensional setting, because in many applications, the unknowns to be recovered as well as the data are most naturally modeled as functions that lie in infinite-dimensional spaces \mathbb{X} and \mathbb{Y} . However, everything said in this chapter applies to finite dimensional spaces as well. In limited data problems, such as sparse-view CT, the finite dimension of the data space \mathbb{Y} is even an intrinsic part of the forward model. Therefore, the reader not familiar with infinite-dimensional vector space can think of \mathbb{X} and \mathbb{Y} as finite-dimensional vector spaces each equipped with a standard vector norm.

Preliminaries

In this section, we provide necessary background on linear inverse problems, their regularization, and their solution with neural networks.

Throughout the following, \mathbb{X} and \mathbb{Y} are Banach spaces. We study solving inverse problems of the form (1) in a deterministic setting with a bounded linear forward operator $\mathbf{A}: \mathbb{X} \rightarrow \mathbb{Y}$. Hence we aim to estimate the unknown signal $\mathbf{x}^* \in \mathbb{X}$ from the available data $\mathbf{y} = \mathbf{A}(\mathbf{x}^*) + \xi$, where $\xi \in \mathbb{Y}$ is the noise that is assumed to satisfy an estimate of the form $\|\xi\| \leq \delta$. Here, $\delta \geq 0$ is called the noise level, and in the case $\delta = 0$, we call $\mathbf{y} = \mathbf{A}(\mathbf{x}^*)$ the exact data.

Right Inverses

As we have explained in the Introduction, the main feature of inverse problems is their ill-posedness. Regularization methods approach the ill-posedness by two steps. In the first step, they address (I1) and (I2) by restricting the image and the pre-image spaces, which gives a certain right inverse defined on $\text{ran}(\mathbf{A})$. In order to address the instability issue (I3), in a second step, regularization methods are applied for stabilization. We first consider right inverse and their instability and consider the regularization in the following subsection.

Definition 1 (Right inverse). A possibly non-linear mapping $\mathbf{B}: \text{ran}(\mathbf{A}) \subseteq \mathbb{Y} \rightarrow \mathbb{X}$ is called right inverse of \mathbf{A} if $\mathbf{A}(\mathbf{B}(\mathbf{y})) = \mathbf{y}$ for all $\mathbf{y} \in \text{ran}(\mathbf{A})$.

Clearly, a right inverse always exists because for any $\mathbf{y} \in \text{ran}(\mathbf{A})$, there exists an element $\mathbf{B}\mathbf{y} := \mathbf{x}$, such that $\mathbf{A}\mathbf{x} = \mathbf{y}$. However, in general, no continuous right inverse exists. More precisely, we have the following result (compare Nashed 1987).

Proposition 1 (Continuous right inverses). Let $\mathbf{B}: \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}$ be a continuous right inverse. Then, $\text{ran}(\mathbf{A})$ is closed.

Proof. By continuity, \mathbf{B} can be extended in a unique way to a mapping $\mathbf{H}: \overline{\text{ran}(\mathbf{A})} \rightarrow \mathbb{X}$. The continuity of \mathbf{H} and \mathbf{A} implies $\mathbf{A} \circ \mathbf{H} = \text{Id}_{\overline{\text{ran}(\mathbf{A})}}$. Therefore $\overline{\text{ran}(\mathbf{A})} = \text{ran}(\mathbf{A} \circ \mathbf{H}) \subseteq \text{ran}(\mathbf{A}) \subseteq \overline{\text{ran}(\mathbf{A})}$ which shows that $\text{ran}(\mathbf{A})$ is closed.

Proposition 1 implies that whenever $\text{ran}(\mathbf{A})$ is non-closed, \mathbf{A} does not have a continuous right inverse.

The next question we study is the existence of a linear right inverse. For that purpose recall that a mapping $\mathbf{P}: \mathbb{X} \rightarrow \mathbb{X}$ is called projection if $\mathbf{P}^2 = \mathbf{P}$. If \mathbf{P} is a linear bounded projection, then $\text{ran}(\mathbf{P})$ and $\ker(\mathbf{P})$ are closed subspaces and $\mathbb{X} = \text{ran}(\mathbf{P}) \oplus \ker(\mathbf{P})$.

Definition 2 (Complemented subspace). A closed (linear) subspace \mathbb{V} of \mathbb{X} is called complemented in \mathbb{X} if there exists a bounded linear projection \mathbf{P} with $\text{ran}(\mathbf{P}) = \mathbb{V}$

A closed subspace $\mathbb{V} \subseteq \mathbb{X}$ is complemented if and only if there is another closed subspace $\mathbb{U} \subseteq \mathbb{X}$ with $\mathbb{X} = \mathbb{U} \oplus \mathbb{V}$. In a Hilbert space, any closed subspace is complemented, and $\mathbb{X} = \mathbb{V}^\perp \oplus \mathbb{V}$ with the orthogonal complement $\mathbb{V}^\perp := \{u \in \mathbb{X} \mid \forall v \in \mathbb{V}: \langle u, v \rangle = 0\}$. However, as shown in Lindenstrauss and Tzafriri (1971), in every Banach space that is not isomorphic to a Hilbert space, there exist closed subspaces which are not complemented.

Proposition 2 (Linear right inverses).

- (a) \mathbf{A} has a linear right inverse with bounded $\mathbf{B} \circ \mathbf{A}$ if and only if $\ker(\mathbf{A})$ is complemented.
- (b) A linear right inverse as in (a) is continuous if and only if $\text{ran}(\mathbf{A})$ is closed.

Proof. (a) First, suppose that \mathbf{A} has a linear right inverse $\mathbf{B}: \mathbb{X} \rightarrow \mathbb{X}$ such that $\mathbf{B} \circ \mathbf{A}$ is bounded. For any $\mathbf{x} \in \mathbb{X}$, we have $(\mathbf{B} \circ \mathbf{A})^2(\mathbf{x}) = \mathbf{B} \circ (\mathbf{A} \circ \mathbf{B})(\mathbf{A}(\mathbf{x})) = (\mathbf{B} \circ \mathbf{A})(\mathbf{x})$. Hence, $\mathbf{B} \circ \mathbf{A}$ is a linear bounded projection. This implies the topological decomposition $\mathbb{X} = \text{ran}(\mathbf{B} \circ \mathbf{A}) \oplus \ker(\mathbf{B} \circ \mathbf{A})$ with closed subspaces $\text{ran}(\mathbf{B} \circ \mathbf{A})$ and $\ker(\mathbf{B} \circ \mathbf{A})$. It holds $\ker(\mathbf{B} \circ \mathbf{A}) \supseteq \ker(\mathbf{A}) = \ker(\mathbf{A} \circ \mathbf{B} \circ \mathbf{A}) \supseteq \text{ran}(\mathbf{B} \circ \mathbf{A})$, which shows that $\ker(\mathbf{A}) = \text{ran}(\mathbf{B} \circ \mathbf{A})$ is complemented. Conversely let $\ker(\mathbf{A})$ be complemented, and write $\mathbb{X} = \mathbb{X}_1 \oplus \ker(\mathbf{A})$. Then $\mathbf{A}_{\text{res}}: \mathbb{X}_1 \rightarrow \text{ran}(\mathbf{A})$ is bijective and therefore has a linear inverse $\mathbf{A}_{\text{res}}^{-1}$ defining a desired right inverse for \mathbf{A} .

(b) For any continuous right inverse, $\text{ran}(\mathbf{A})$ is closed according to Proposition 1. Conversely, let $\mathbf{B}: \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}$ be linear right inverse such that $\mathbf{B} \circ \mathbf{A}$ is bounded and $\text{ran}(\mathbf{A})$ closed. In particular, $\ker(\mathbf{A})$ is complemented, and we can write $\mathbb{X} = \mathbb{X}_1 \oplus \ker(\mathbf{A})$. The restricted mapping $\mathbf{A}_{\text{res}}: \mathbb{X}_1 \rightarrow \text{ran}(\mathbf{A})$ is bijective, therefore bounded according to the bounded inverse theorem. This implies that \mathbf{B} is bounded, too. □

In a Hilbert space \mathbb{X} , the kernel $\ker(\mathbf{A})$ of a bounded linear operator is complemented, as any other closed subspace of \mathbb{X} . Therefore, according to Proposition 2, any bounded linear operator defined on a Hilbert space has a linear right inverse. However, in a general Banach space, this is not the case, as the following example shows.

Example 1 (Bounded linear operator without linear right inverse). Consider the set $c_0(\mathbb{N})$ of all sequences converging to zero as a subspace of the space $\ell^\infty(\mathbb{N})$ of all bounded sequences $\mathbf{x}: \mathbb{N} \rightarrow \mathbb{R}$ with the supremum norm $\|\mathbf{x}\|_\infty := \sup_{n \in \mathbb{N}} |\mathbf{x}(n)|$. Note that $c_0(\mathbb{N}) \subseteq \ell^\infty(\mathbb{N})$ is a classic example for a closed subspace that is not complemented in a Banach space, as first shown in Phillips (1940). Now consider the quotient space $\mathbb{Y} = \ell^\infty(\mathbb{N}) / c_0(\mathbb{N})$, where elements in $\ell^\infty(\mathbb{N})$ are identified if their difference is contained in $c_0(\mathbb{N})$. Then the quotient map $\mathbf{A}: \ell^\infty(\mathbb{N}) \rightarrow \mathbb{Y}: \mathbf{x} \mapsto [\mathbf{x}]$ is clearly linear, bounded, and onto with $\ker(\mathbf{A}) = c_0(\mathbb{N})$. It is clear that a right inverse of \mathbf{A} exists, which can be constructed by simply choosing any representative in $[\mathbf{x}]$. However, because $c_0(\mathbb{N})$ is not complemented, the kernel of \mathbf{A} is not complemented, and according to Proposition 2, no linear right inverse \mathbf{B} of \mathbf{A} such that $\mathbf{B} \circ \mathbf{A}$ is bounded.

At first glance it might be surprising that bounded linear forward operators do not always have suitable linear right inverses. However, following Example 1, one constructs bounded linear operators without linear right inverses for every Banach space that is not isomorphic to a Hilbert space. This in particular includes the function spaces $L^p(\Omega)$ with $p \neq 2$, where inverse problems are often formulated on.

Proposition 3 (Right inverses in Hilbert spaces). *Let \mathbb{X} be a Hilbert space and let $\mathbf{P}_{\ker(\mathbf{A})} : \mathbb{X} \rightarrow \mathbb{X}$ denote the orthogonal projection onto $\ker(\mathbf{A})$.*

- (a) \mathbf{A} has a unique linear right inverse $\mathbf{A}^\dagger : \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}$ with $\mathbf{A} \circ \mathbf{A}^\dagger = \text{Id} - \mathbf{P}_{\ker(\mathbf{A})}$.
- (b) $\forall \mathbf{y} \in \text{ran}(\mathbf{A}) : \mathbf{A}^\dagger(\mathbf{y}) = \arg \min \{ \|\mathbf{x}\| \mid \mathbf{A}(\mathbf{x}) = \mathbf{y} \}$.
- (c) \mathbf{A}^\dagger is continuous if and only if $\text{ran}(\mathbf{A})$ is closed.
- (d) If $\text{ran}(\mathbf{A})$ is non-closed, then any right inverse is discontinuous.

Proof. In a Hilbert space, the orthogonal complement $\ker(\mathbf{A})^\perp$ defines a complement of $\ker(\mathbf{A})$, and therefore, (a), (c), (d) follow from Propositions 1 and 2. Item (b) holds because any solution of the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has the form $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 \in \ker(\mathbf{A})^\perp \oplus \ker(\mathbf{A})$, and we have $\|\mathbf{x}\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2$ according to Pythagoras theorem. □

In the case that \mathbb{X} and \mathbb{Y} are both Hilbert spaces, there is a unique extension $\mathbf{A}^\dagger : \text{ran}(\mathbf{A}) \oplus \text{ran}(\mathbf{A})^\perp \rightarrow \mathbb{X}$, such that $\mathbf{A}^\dagger(\mathbf{y}_1 \oplus \mathbf{y}_2) = \mathbf{A}^\dagger(\mathbf{y}_1)$ for all $\mathbf{y}_1 \oplus \mathbf{y}_2 \in \text{ran}(\mathbf{A}) \oplus \text{ran}(\mathbf{A})^\perp$. The operator \mathbf{A}^\dagger is referred to as the Moore-Penrose inverse of \mathbf{A} . For more background on generalized inverses in Hilbert and Banach spaces, see Nashed (1987).

Regularization Methods

Let $\mathbf{B} : \text{ran}(\mathbf{A}) \subseteq \mathbb{Y} \rightarrow \mathbb{X}$ be a right inverse of \mathbf{A} , set $\mathbb{M} := \text{ran}(\mathbf{B})$ and suppose $\mathbb{M}^* \subseteq \mathbb{M}$. Moreover, let $\mathcal{D} : \mathbb{Y} \times \mathbb{Y} \rightarrow [0, \infty]$ be some functional measuring closeness in the data space. The standard choice is the norm distance $\mathbf{d}_{\mathbb{Y}}(\mathbf{y}, \mathbf{y}^\delta) := \|\mathbf{y} - \mathbf{y}^\delta\|$ but also other choices will be considered in this chapter.

Definition 3 (Regularization method). A function $\mathbf{R} : (0, \infty) \times \mathbb{Y} \rightarrow \mathbb{X}$ with

$$\forall \mathbf{x} \in \mathbb{M}^* : \limsup_{\delta \rightarrow 0} \left\{ \|\mathbf{x} - \mathbf{R}(\delta, \mathbf{y}^\delta)\| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) \leq \delta \right\} = 0 \tag{3}$$

is called (convergent) regularization method for (1) on the signal class $\mathbb{M}^* \subseteq \mathbb{M}$ with respect to the similarity measure \mathcal{D} . We also write $(\mathbf{R}_\delta)_{\delta > 0}$ instead of \mathbf{R} .

The following lemma gives a useful guideline for creating regularization methods based on point-wise approximations of \mathbf{B} .

Proposition 4 (Point-wise approximations are regularizations). *Let $(\mathbf{B}_\alpha)_{\alpha > 0}$ be a family of continuous operators $\mathbf{B}_\alpha : \mathbb{Y} \rightarrow \mathbb{X}$ that converge uniformly to \mathbf{B} on $\mathbb{A}(\mathbb{M}^*)$ as $\alpha \rightarrow 0$. Then, there is a function $\alpha_0 : (0, \infty) \rightarrow (0, \infty)$ such that*

$$\mathbf{R} : (0, \infty) \times \mathbb{Y} \rightarrow \mathbb{X} : (\delta, \mathbf{y}^\delta) \mapsto \mathbf{R}(\delta, \mathbf{y}^\delta) := \mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta) \tag{4}$$

is a regularization method for (1) on the signal class \mathbb{M}^* with respect to the norm distance $\mathbf{d}_{\mathbb{Y}}$. One calls α_0 an a-prior parameter choice over the set \mathbb{M}^* .

Proof. For any $\epsilon > 0$, choose $\alpha(\epsilon)$ such $\|\mathbf{B}_{\alpha(\epsilon)}(\mathbf{y}) - \mathbf{x}\| \leq \epsilon/2$ for all $\mathbf{x} \in \mathbb{M}^*$. Moreover, choose $\tau(\epsilon)$ such that for all $\mathbf{z} \in \mathbb{Y}$ with $\|\mathbf{y} - \mathbf{z}\| \leq \tau(\epsilon)$, we have $\|\mathbf{B}_{\alpha(\epsilon)}(\mathbf{y}) - \mathbf{B}_{\alpha(\epsilon)}(\mathbf{z})\| \leq \epsilon/2$. Without loss of generality, we can assume that $\tau(\epsilon)$ is strictly increasing and continuous with $\tau(0+) = 0$. We define $\alpha_0 := \alpha \circ \tau^{-1}$. Then, for every $\delta > 0$ and $\|\mathbf{y} - \mathbf{y}^\delta\| \leq \delta$,

$$\begin{aligned} \|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta) - \mathbf{x}\| &\leq \|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}) - \mathbf{x}\| + \|\mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}) - \mathbf{B}_{\alpha_0(\delta)}(\mathbf{y}^\delta)\| \\ &= \|\mathbf{B}_{\alpha \circ \tau^{-1}(\delta)}(\mathbf{y}) - \mathbf{x}\| + \|\mathbf{B}_{\alpha \circ \tau^{-1}(\delta)}(\mathbf{y}) - \mathbf{B}_{\alpha \circ \tau^{-1}(\delta)}(\mathbf{y}^\delta)\| \\ &\leq \tau^{-1}(\delta)/2 + \tau^{-1}(\delta)/2 = \tau^{-1}(\delta). \end{aligned}$$

Because $\tau^{-1}(\delta) \rightarrow 0$ as $\delta \rightarrow 0$ this completes the proof. \square

A popular class of regularization methods is convex variational regularization defined by a convex functional $\mathcal{R}: \mathbb{X} \rightarrow [0, \infty]$. These methods approximate right inverses, given by the \mathcal{R} -minimizing solutions of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. Such solutions are elements in $\arg \min\{\mathcal{R}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{X} \wedge \mathbf{A}(\mathbf{x}) = \mathbf{y}\}$. Note that an \mathcal{R} -minimizing solution exists whenever \mathbb{X} is reflexive, \mathcal{R} is coercive and weakly lower semi-continuous, and the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has at least one solution in the domain of \mathcal{R} . Moreover, the \mathcal{R} -minimizing solution is unique if \mathcal{R} is strictly convex. In this case this immediately defines a right inverse for \mathbf{A} . Convex variational regularization is defined by minimizing the Tikhonov functional $\mathcal{T}_{\mathbf{y}^\delta, \alpha}: \mathbb{X} \rightarrow [0, \infty]: \mathbf{x} \mapsto \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha \mathcal{R}(\mathbf{x})$ for data $\mathbf{y}^\delta \in \mathbb{Y}$ and regularization parameter $\alpha > 0$. In section “[The NETT Approach](#)”, we will study a more general form, including non-convex regularizers defined by a neural network. At this point, we only state one result on convex variational regularization.

Theorem 1 (Tikhonov regularization in Banach spaces). *Let \mathbb{X} be reflexive, strictly convex, and $p, q > 1$. Moreover, suppose that \mathbb{X} satisfies the Radon–Riesz property; that is, for any sequence $(\mathbf{x}_k)_{k \in \mathbb{N}} \in \mathbb{X}^{\mathbb{N}}$, the weak convergence $\mathbf{x}_k \rightharpoonup \mathbf{x} \in \mathbb{X}$ together with the convergence in the norm $\|\mathbf{x}_k\| \rightarrow \|\mathbf{x}\|$ implies $\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}$ in the norm topology. Then the following holds:*

- (a) $\mathbf{A}^\dagger: \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}: \mathbf{y} \mapsto \arg \min\{\|\mathbf{x}\| \mid \mathbf{x} \in \mathbb{X} \wedge \mathbf{A}(\mathbf{x}) = \mathbf{y}\}$ is well defined.
- (b) For all, $\alpha > 0$ the mapping $\mathbf{B}_\alpha: \mathbb{Y} \rightarrow \mathbb{X}: \mathbf{y}^\delta \mapsto \arg \min\{\|\mathbf{A}(\mathbf{x}) - \mathbf{y}^\delta\|^p + \|\mathbf{x}\|^q \mid \mathbf{x} \in \mathbb{X}\}$ is well defined and continuous.
- (c) For any $\alpha_0: (0, \infty) \rightarrow (0, \infty)$ with $\alpha_0 \rightarrow 0$ and $\delta^p/\alpha_0(\delta) \rightarrow 0$ as $\delta \rightarrow 0$, the mapping defined by (4) and (b) is a regularization method for (1) on $\mathbf{A}^\dagger(\mathbb{X})$ with respect to the norm distance $\mathbf{d}_{\mathbb{X}}$.

Proof. See Ivanov et al. (2002) and Scherzer et al. (2009).

In the Hilbert space setting, the mapping \mathbf{A}^\dagger defined In Theorem 1 is given by the Moore-Penrose inverse; see Proposition 3 and the text below this Proposition.

Deep Learning

In this subsection, we give a brief review of neural networks and deep learning. Deep learning can be characterized as the field where deep neural networks are used to solve various learning problems (LeCun et al. 2015; Goodfellow et al. 2016). Several such methods recently appeared as a new paradigm for solving inverse problems. In deep learning literature, neural networks are often formulated in a finite dimensional setting. To allow a unified treatment, we consider here a general setting, including the finite dimensional as well as the infinite-dimensional setting.

Problem 1 (The supervised learning problem). Suppose the aim is to find an unknown function $\Phi: \mathbb{Y} \rightarrow \mathbb{X}$ between two Banach spaces. Similar to classical regression, we are given data $(\mathbf{y}_i, \mathbf{x}_i) \in \mathbb{Y} \times \mathbb{X}$ with $\Phi(\mathbf{y}_i) \simeq \mathbf{x}_i$ for $i = 1, \dots, N$. From this data, we aim to estimate the function Φ . For that purpose, one chooses a certain class $(\Phi_\theta)_{\theta \in \Theta}$ of functions $\Phi_\theta: \mathbb{Y} \rightarrow \mathbb{X}$ and defines $\Phi := \Phi_{\theta^*}$ where θ^* minimizes the penalized empirical risk functional

$$R_N: \Theta \rightarrow [0, \infty]: \theta \mapsto \frac{1}{N} \sum_{i=1}^N L(\Phi_\theta(\mathbf{y}_i), \mathbf{x}_i) + P(\theta). \tag{5}$$

Here $L: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ is the so-called loss function, which is a measure for the error made by the function Φ_θ on the training examples, and P is a penalty that prevents overfitting of the network and also stabilizes the training process.

Both the numerical minimization of the functional (5) and investigating properties of θ^* as $N \rightarrow \infty$ are of interest in its own (Glorot and Bengio 2010; Chen et al. 2018) but not subject of our analysis. Instead, most theory in this chapter is developed under the assumption of suitable trained prediction function.

Definition 4 (Neural network). Let Θ be a parameter set and $H_{\ell, \theta}: \mathbb{X}_0 \times \dots \times \mathbb{X}_{\ell-1} \rightarrow \mathbb{X}_\ell$, for $\ell = 1, \dots, L$ and $\theta \in \Theta$ be mappings between Banach spaces with $\mathbb{X}_0 = \mathbb{Y}$ and $\mathbb{X}_L = \mathbb{X}$. We call a family $(\Phi_\theta)_{\theta \in \Theta}$ of recursively defined mappings

$$\Phi_\theta := a_{L, \theta}: \mathbb{Y} \rightarrow \mathbb{X} \quad \text{where } \forall \ell \in \{1, \dots, L\}: a_{\ell, \theta} = H_{\ell, \theta}(\text{Id}, a_{1, \theta}, \dots, a_{\ell-1, \theta}) \tag{6}$$

a neural network. In that context, $\mathbb{X}_1, \dots, \mathbb{X}_{L-1}$ are called the hidden spaces. We refer to the individual members Φ_θ of a neural network as neural network functions.

A neural network in finite dimension can be seen as discretization of $(\Phi_\theta)_{\theta \in \Theta}$, where \mathbb{Y} and \mathbb{X} are discretized using any standard discretization approach.

Example 2 (Layered neural network). As a typical example for a neural network, consider a layered neural network $(\Phi_\theta)_{\theta \in \Theta}$ with L layers between finite dimensional spaces. In this case, each network function has the form $\Phi_\theta: \mathbb{R}^p \rightarrow \mathbb{R}^q: \mathbf{y} \mapsto (\sigma_L \circ \mathcal{V}_L^\theta) \circ \dots \circ (\sigma_1 \circ \mathcal{V}_1^\theta)(\mathbf{y})$, where $\mathcal{V}_\ell^\theta: \mathbb{R}^{d(\ell-1)} \rightarrow \mathbb{R}^{d(\ell)}$ are affine mappings and $\sigma_\ell: \mathbb{R}^{d(\ell)} \rightarrow \mathbb{R}^{d(\ell)}$ are nonlinear mappings with $d(0) = p$ and $d(L) = q$. The notation indicates that the affine mappings depend on the parameters $\theta \in \Theta$, while the nonlinear mappings are taken fixed. Although this is standard in neural networks, modifications where the nonlinearities contain trainable parameters have been proposed (Agostinelli et al. 2014; Ramachandran et al. 2017). The affine parts \mathcal{V}_ℓ^θ , which are the learned parts in the neural network, can be represented by a $d(\ell) \times d(\ell - 1)$ matrix for the linear part and a vector of size $1 \times d(\ell)$ for the translation part.

In standard neural networks, the entries of the matrix and the bias vector are taken as independent parameters. For typical inverse problems where the dimensions p and q are large, learning all these numbers is challenging and perhaps an impossible task. For example, the matrix describing the linear part of a layer mapping a 200×200 image to an image of the same size already contains 1.6 billion parameters. Learning these parameters from data seems challenging. Recent neural networks and, in particular, convolutional neural networks (CNNs) use the concepts of sparsity and weight sharing to significantly reduce the number of parameters to be learned.

Example 3 (CNNs using sparsity and weight sharing). In order to reduce the number of free parameter between a linear mapping between images, say of sizes $q = n \times n$ and $p = n \times n$, CNNs implement sparsity and weight sharing via convolution operators. In fact, a convolution operation $\mathbf{K}: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ with kernel size $k \times k$ is represented by k^2 numbers, which clearly enormously reduces the number n^4 of parameters required to represent a general linear mapping on $\mathbb{R}^{n \times n}$. To enrich the expressive power of the neural network, actual CNN architectures use multiple-input multiple-output convolutions $\mathbf{K}: \mathbb{R}^{n \times n \times c} \rightarrow \mathbb{R}^{n \times n \times d}$, which uses one convolution kernel for each pair in $\{1, \dots, c\} \times \{1, \dots, d\}$ formed between each input channel and each output channel. This now increases the number of learnable parameters to cdk^2 , but overall the number of parameters remains much smaller than for a full dense layer between large images. Moreover, the use of multiple-input multiple-output convolutions in combination with typical nonlinearities introduces a flexible and complex structure, which demonstrated to give state-of-the art results in various imaging tasks.

Regularizing Networks

Throughout, this section let $\mathbf{A}: \mathbb{X} \rightarrow \mathbb{Y}$ be a linear forward operator between Banach spaces and $\mathbf{B}: \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}$ a linear right inverse with $\mathbb{U} := \text{ran}(\mathbf{B})$. In particular, the kernel of \mathbf{A} is complemented, and we can write $\mathbb{X} = \mathbb{U} \oplus \ker(\mathbf{A})$.

The results in this section generalize the methods and some of the results of Schwab et al. (2019) from the Hilbert case to the Banach space case.

Null-Space Networks

The idea of post-processing networks is to improve a given right inverse by applying a network. Standard networks, however, will destroy data consistency of the initial reconstruction. Null-space networks are the natural class of neural networks restoring data consistency.

Definition 5 (Null-space network). We call the family $(\text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta})_{\theta \in \Theta}$ a null-space network if $(\mathbf{N}_{\theta})_{\theta \in \Theta}$ is any network of Lipschitz continuous functions $\mathbf{N}_{\theta} : \mathbb{X} \rightarrow \mathbb{X}$. We will also refer to individual functions $\Phi_{\theta} = \text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta}$ as null-space networks.

Any null-space network $\Phi_{\theta} = \text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta}$ preserves data consistency in the sense that $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ implies $\mathbf{A}(\Phi_{\theta}(\mathbf{x})) = \mathbf{y}$, which can be seen from

$$\mathbf{A} \circ (\text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta})(\mathbf{x}) = \mathbf{A}(\mathbf{x}) + \mathbf{A} \circ \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta}(\mathbf{x}) = \mathbf{y}. \tag{7}$$

A standard residual network $\text{Id}_{\mathbb{X}} + \mathbf{N}_{\theta}$ often used as post-processing network in general does not satisfy this such a data consistency property.

Remark 1 (Computation of the projection layer). One of the main ingredient in the null-space network is the computation of the projection layer $\mathbf{P}_{\ker(\mathbf{A})}$. In some cases, it can be computed explicitly. For example, if $\mathbf{A} = \mathbf{S}_I \circ \mathbf{F}$ is the subsampled Fourier transform, then $\mathbf{P}_{\ker(\mathbf{A})} = \mathbf{F}^* \circ \mathbf{S}_I \circ \mathbf{F}$. For a general forward operator between Hilbert spaces, the projection $\mathbf{P}_{\ker(\mathbf{A})}\mathbf{z}$ can be implemented via standard methods for solving linear equation. For example, using the starting value \mathbf{z} and solving the equation $\mathbf{A}(\mathbf{x}) = 0$ with the CG (conjugate gradient) method for the normal equation or Landwebers methods gives a sequence that converges to the projection $\mathbf{P}_{\ker(\mathbf{A})}\mathbf{z} = \arg \min \{\|\mathbf{x} - \mathbf{z}\| \mid \mathbf{A}(\mathbf{x}) = 0\}$.

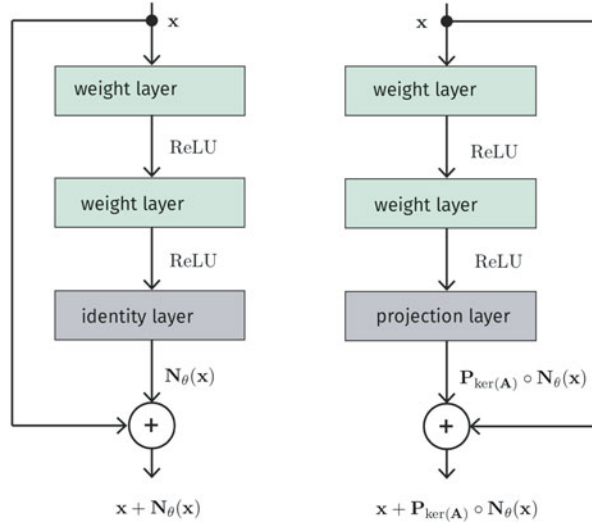
An example comparing a standard residual network $\text{Id}_{\mathbb{X}} + \mathbf{N}_{\theta}$ and a null-space network $\text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta}$ both with two weight layers are shown in Fig. 1.

Proposition 5 (Right inverses defined by null-space networks). Let $\mathbf{B} : \text{ran}(\mathbf{A}) \rightarrow \mathbb{X}$ be a given linear right inverse such that $\mathbf{B} \circ \mathbf{A}$ is bounded and $\Phi_{\theta} = \text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta}$ be a null-space network. Then the composition

$$\Phi_{\theta} \circ \mathbf{B} : \text{ran}(\mathbf{A}) \rightarrow \mathbb{X} : \mathbf{y} \mapsto (\text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_{\theta})(\mathbf{B}\mathbf{y}) \tag{8}$$

is right inverse of \mathbf{A} . Moreover, the following assertions are equivalent:

Fig. 1 Residual network $\text{Id} + \mathbf{N}_\theta$ (left) versus null-space network $\text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ (right). The difference between the two architectures is the projection layer $\mathbf{P}_{\ker(\mathbf{A})}$ in the null-space network after the last weight layer



- (i) $\Phi_\theta \circ \mathbf{B}$ is continuous
- (ii) \mathbf{B} is continuous
- (iii) $\text{ran}(\mathbf{A})$ is closed.

Proof. Because \mathbf{B} is a right inverse, we have $(\mathbf{A} \circ \mathbf{B})(\mathbf{x}) = \mathbf{y}$ for all $\mathbf{y} \in \text{ran}(\mathbf{A})$. Hence, the data consistency property (7) implies $\mathbf{A}(((\text{Id}_{\mathbb{X}} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta) \circ \mathbf{B})(\mathbf{x})) = \mathbf{y}$, showing that $\Phi_\theta \circ \mathbf{B}$ is a right inverse of \mathbf{A} . The implication (i) \Rightarrow (ii) follows from the identity $\mathbf{P}_{\cup} \circ \Phi_\theta \circ \mathbf{B} = \mathbf{B}$ and the continuity of the projection. The implication (ii) \Rightarrow (iii) follows from the continuity of Φ_θ . Finally, the equivalence (ii) \Leftrightarrow (iii) has been established in Proposition 2. \square

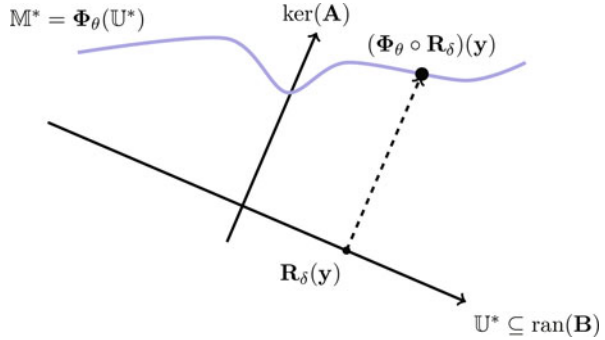
The benefit of non-linear right inverses defined by null-space networks is that they can be adjusted to a given image class. A possible network training is given as follows:

Remark 2 (Possible training strategy). The null-space network $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ can be trained to map elements in \mathbb{M} to the elements from the desired class of images. For that purpose, select training data pairs $(\mathbf{x}_1, \mathbf{z}_1), \dots, (\mathbf{x}_N, \mathbf{z}_N)$ with $\mathbf{z}_i = \mathbf{B} \circ \mathbf{A}(\mathbf{x}_i)$ and minimize the regularized empirical risk,

$$R_N : \Theta \rightarrow [0, \infty] : \theta \mapsto \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_i - \Phi_\theta(\mathbf{z}_i)\|^2 + \beta \|\theta\|_2^2. \tag{9}$$

Note that for our analysis, it is not required that (9) is exactly minimized. Any null-space network Φ_θ where $\sum_{n=1}^N \|\mathbf{x}_i - \Phi_\theta(\mathbf{z}_i)\|^2$ is small yields a right inverse $\Phi_\theta \mathbf{B}$ that does a better job in estimating \mathbf{x}_n from data $\mathbf{A}\mathbf{x}_n$ than the original right inverse \mathbf{B} .

Fig. 2 Linear Regularization $(\mathbf{R}_\delta)_{\delta>0}$ combined with a null-space network $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$. We start with a linear regularization $\mathbf{R}_\delta \mathbf{y}$ and the null-space network $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ adds missing parts along the null space $\ker(\mathbf{A})$



Proposition 5 implies that the solution of ill-posed problems by null-space networks requires the use of stabilization methods similar to the case of classical methods. In the following subsection, we show that the combination of null-space network with a regularization of \mathbf{B} in fact yields a regularization method on a signal class related to the null-space network.

Convergence Analysis

Throughout the following, let $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ be null-space network and $(\mathbf{R}_\delta)_{\delta>0}$ be a regularization method for (1) on the signal class $\mathbb{U}^* \subseteq \mathbb{U}$ with respect to the similarity measure \mathcal{D} as introduced in Definition 3. As illustrated in Fig. 2, we consider the family $(\Phi_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ of compositions of the regularization method with the null-space network.

Theorem 2 (Regularizing null-space network). *For a given null-space network $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$ and a given regularization method $(\mathbf{R}_\delta)_{\delta>0}$ on the signal class \mathbb{U}^* , the family $(\Phi_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ is regularization method for (1) on the signal class $\Phi_\theta(\mathbb{U}^*)$ with respect to the similarity measure \mathcal{D} . We call $(\Phi_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ a regularizing null-space network.*

Proof. Let L be a Lipschitz constant of Φ_θ and recall $\Phi_\theta = \text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta$. For any $\mathbf{x} \in \Phi_\theta(\mathbb{U}^*)$ and $\mathbf{y}^\delta \in \mathbb{Y}$, we have

$$\begin{aligned} \|\mathbf{x} - \Phi_\theta \circ \mathbf{R}_\delta(\mathbf{y}^\delta)\| &= \|(\text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)(\mathbf{B} \circ \mathbf{A}(\mathbf{x})) - (\text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)(\mathbf{R}_\delta(\mathbf{y}^\delta))\| \\ &\leq L \|(\mathbf{B} \circ \mathbf{A})(\mathbf{x}) - \mathbf{R}_\delta(\mathbf{y}^\delta)\|. \end{aligned}$$

Here, we have used the identity $\mathbf{x} = (\text{Id} + \mathbf{P}_{\ker(\mathbf{A})} \circ \mathbf{N}_\theta)((\mathbf{B} \circ \mathbf{A})(\mathbf{x}))$ for $\mathbf{x} \in \text{ran}(\Phi_\theta)$. Consequently

$$\begin{aligned} & \sup \{ \|\mathbf{x} - (\Phi_\theta \circ \mathbf{R}_\delta)(\mathbf{y}^\delta) \| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{y}^\delta, \mathbf{y}) \leq \delta \} \\ & \leq L \sup \{ \|(\mathbf{B} \circ \mathbf{A})(\mathbf{x}) - \mathbf{R}_\delta(\mathbf{y}^\delta) \| \mid \mathbf{y}^\delta \in \mathbb{Y} \wedge \mathcal{D}(\mathbf{y}^\delta, \mathbf{y}) \leq \delta \} \rightarrow 0. \end{aligned}$$

In particular, $(\Phi_\theta \circ \mathbf{R}_\delta)_{\delta>0}$ is a regularization method for (1) on $\Phi_\theta(\mathbb{U}^*)$ with respect to the similarity measure \mathcal{D} \square

In Hilbert spaces, a wide class of regularizing reconstruction networks can be defined by regularizing filters.

Example 4 (Regularizations defined by filters). Let \mathbb{X} and \mathbb{Y} be Hilbert spaces. A family $(g_\alpha)_{\alpha>0}$ of functions $g_\alpha: [0, \|\mathbf{A}^* \circ \mathbf{A}\|] \rightarrow \mathbb{R}$ is called a regularizing filter if it satisfies

- For all $\alpha > 0$, g_α is piecewise-continuous and bounded.
- $\exists C > 0: \sup \{ |\lambda g_\alpha(\lambda)| \mid \alpha > 0 \wedge \lambda \in [0, \|\mathbf{A}^* \circ \mathbf{A}\|] \} \leq C$.
- $\forall \lambda \in (0, \|\mathbf{A}^* \circ \mathbf{A}\|): \lim_{\alpha \rightarrow 0} g_\alpha(\lambda) = 1/\lambda$.

For a given regularizing filter $(g_\alpha)_{\alpha>0}$, define $\mathbf{B}_\alpha := g_\alpha(\mathbf{A}^* \circ \mathbf{A})\mathbf{A}^*$. Then for a suitable parameter choice $\alpha = \alpha(\delta, \mathbf{y})$, the family $(\mathbf{B}_{\alpha(\delta, \cdot)})_{\delta>0}$ is a regularization method on $\text{ran}(\mathbf{A}^\dagger)$. Therefore, according to Theorem 2, the family $(\Phi_\theta \circ \mathbf{B}_{\alpha(\delta, \cdot)})_{\delta>0}$ is a regularization method on $\Phi_\theta(\text{ran}(\mathbf{A}^\dagger))$. Note that in this setting, one can derive quantitative error estimates (convergence rates); we refer the interested reader to the original paper Schwab et al. (2019).

Extensions

The regularizing null-space networks defined in Theorem 2 are of the form $\Phi_\theta \circ \mathbf{R}_\delta$, where \mathbf{R}_δ is a classical regularization and Φ_θ only acts in the null space of \mathbf{A} . In order to better account for noise, it is beneficial to allow the networks to modify \mathbf{R}_δ also on the complement \mathbb{U} .

Definition 6 (Regularizing family of networks). Let $(\mathbf{R}_\delta)_{\delta>0}$ be a regularization method for (1) on the signal class $\mathbb{U}^* \subseteq \mathbb{U}$ with respect to the similarity measure \mathcal{D} as introduced in Definition 3. A family $(\Phi_\theta(\delta) \circ \mathbf{R}_\delta)_{\delta>0}$ is called regularizing family of networks if $(\Phi_\theta)_{\theta \in \Theta}$ is a neural network, such that the network functions $\Phi_\theta(\delta): \mathbb{X} \rightarrow \mathbb{X}$, for $\delta > 0$, are uniformly Lipschitz continuous and

$$\forall \mathbf{z} \in \text{ran}(\mathbf{B}): \quad \lim_{\delta \rightarrow 0} \Phi_\theta(\delta)(\mathbf{R}_\delta \circ \mathbf{A}(\mathbf{z})) = \mathbf{N}(\mathbf{z}),$$

for some null-space network \mathbf{N} .

Regularizing families of networks have been introduced in Schwab et al. (2020), where it has been shown that a regularizing family of networks defines a regularization method together with convergence rates. Moreover, an example in the form of a data-driven extension of truncated SVD regularization has been given. In a finite dimensional setting, related extension of null-space networks named deep decomposition learning has been introduced in Chen and Davies (2019). A combination of null-space learning with shearlet reconstruction for limited angle tomography has been introduced in Bubba et al. (2019). In Dittmer and Maass (2019), a neural network-based projection approach based on approximate data consistency sets has been studied. Relaxed versions of null-space networks, where approximate data consistency is incorporated via a confidence region or a soft penalty, are proposed in Huang et al. (2020) and Kofler et al. (2020). Finally, extensions of the null-space approach to non-linear problems are studied in Boink et al. (2020).

The NETT Approach

Let us recall that convex variational regularization of the inverse problem (1) consists in minimizing the generalized Tikhonov functional $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha \mathcal{R}(\mathbf{x})$, where \mathcal{R} is a convex functional and \mathcal{D} a similarity measure (see section “[Regularization Methods](#)”). The regularization term \mathcal{R} is traditionally a semi-norm defined on a dense subspace of \mathbb{X} . In this section, we will extend this setup by using deep learning techniques with learned regularization functionals.

Learned Regularization Functionals

We assume that the regularizer takes the form

$$\forall \mathbf{x} \in \mathbb{X}: \quad \mathcal{R}(\mathbf{x}) = \mathcal{R}_\theta(\mathbf{x}) := \psi_\theta(\Phi_\theta(\mathbf{x})). \quad (10)$$

Here $\psi_\theta: \mathbb{E} \rightarrow [0, \infty]$ is a scalar functional, and $\Phi_\theta(\cdot): \mathbb{X} \rightarrow \mathbb{E}$ a neural network where $\theta \in \Theta$, for some vector space Θ containing free parameters that can be adjusted by available training data. From the representation learning point of view (Bengio et al. 2013), $\Phi_\theta(\mathbf{x})$ can be interpreted as a learned representation of \mathbf{x} . It could be constructed in such a way that $\psi_\theta \circ \Phi_\theta$ is minimal for a low-dimensional manifold where the true signals \mathbf{x} are clustered around. Finding such manifold for biomedical images has been an active research topic on manifold learning (Georg et al. 2008; Wachinger et al. 2012). Deep learning has also been used for this purpose (Brosch et al. 2013). A learned regularizer $\mathcal{R}_\theta = \psi_\theta \circ \Phi_\theta$ reflects the statistics of the signal space, which penalizes those who deviate from the data manifold.

The similarity measure is taken as $\mathcal{D}_\theta: \mathbf{C} \times \mathbf{C} \rightarrow [0, \infty]$, where \mathbf{C} is a conic closed subset in \mathbb{Y} . It is not necessarily symmetric in its arguments. One may take $\mathcal{D}_\theta(\mathbf{A}(\mathbf{x}), \mathbf{y})$ to be a common hard-coded consistency measure such as $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}) =$

$\|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|^2$ or the Kullback-Leibler divergence (which, among others, is used in emission tomography). On the other hand, it can be a learned measure, defined via a neural network. A learned consistency measure \mathcal{D}_θ reflects the statistics in the data (measurement) space. It can learn to reduce uncertainty in the data measurement process, e.g., by identifying non-functional transducers. It can also learn to reduce the error in the forward model (Aljadaany et al. 2019). Finally, it may encode the range description of the forward operator, which has not been successfully exploited in inverse problems by traditional methods. In summary, it can be said that learned consistency measures have potentially high impact in solving inverse problems.

Using the neural network-based learned regularizer (10) and a learned discrepancy measure as discussed above results in the following optimization problem:

$$\arg \min_{\mathbf{x} \in \mathbb{D}} \mathcal{T}_\theta(\mathbf{x}) := \mathcal{D}_\theta(\mathbf{A}(\mathbf{x}), \mathbf{y}) + \alpha \mathcal{R}_\theta(\mathbf{x}). \quad (11)$$

Solving (11) can be seen as a neural network-based variant of generalized Tikhonov regularization for solving (1). Following Li et al. (2020) we therefore call (11) the network Tikhonov (NETT) approach for solving inverse problems. Currently, there are two main approaches for integrating neural networks in the NETT approach (11): (T1) training the neural networks simultaneously with solving the optimization problem and (T2) training the network independently before solving the optimization problem.

Approach (T1) fuses the data with a solution method of the optimization problem (11). The resulting neural networks, therefore, depend on the method to solve the optimization. This approach enforces the neural networks to learn particular representations that are useful for the chosen optimization technique. These representations will be called *solver-dependent*. The biggest advantage of this end-to-end approach is to provide a direct and relatively fast solution \mathbf{x} for given new data \mathbf{y} . It is commonly realized by unrolling an iterative process (Arridge et al. 2019). The resulting neural network is a cascade of relatively small neural networks; each of them is possibly a variant of those appearing in the data consistency or regularization term. It is worth noting that the neural network does not aim for representation learning. Each layer or block serves to move the approximate solution closer to the exact solution. In contrast to typical iterative methods, each block in an unrolled neural network can be different from others. This is explained to speed up the convergence of the learned iterative method. The success of this approach is an interesting phenomenon that needs further investigation. The use of neural networks to implement and accelerate iterative methods to solve traditional regularization methods has been intensively studied. We refer the reader to Arridge et al. (2019) and the references contained therein.

The approach (T2) is more modular (Li et al. 2020; Lutz et al. 2018) and results in smaller training problems and is closer to the meaning of representation learning. The training of the regularizer may or may not depend on the forward operator \mathbf{A} . In the former case, the resulting representation is called *model-dependent*, while the latter is *model-independent*. Model-dependent representation seems to be crucial in

inverse problem for two reasons. The first reason is that it aligns with the inverse problem (and better serves any solution approach). Secondly, in medical imaging applications, the training signals are often not the groundtruth signals. They are normally obtained with a reconstruction method from high-quality data. Therefore, while training the regularizer, one should also keep in mind the reconstruction mechanism of the training data. A possible approach is to first train a baseline neural network to learn model-independent representation. Then an additional block is added on top to train for model-dependent representation. This has been shown in Obmann et al. (2020a) to be a very efficient strategy.

Let us mention that approach (T1) has richer literature than (T2) but less (convergence) analysis. In this section, we focus more on (T2), where we establish the convergence analysis and convergence rate in section “[Convergence Analysis](#)”. This is an extension of our works Haltmeier et al. (2019) and Obmann et al. (2020a). In section “[Related Methods](#)”, we review a few existing methods that are most relevant to our discussion, including some works in approach (T1). We also propose INDIE, which can be regarded as an operator inversion-free variant of the MODL technique (Aggarwal et al. 2018) and can make better use of parallel computation.

Convergence Analysis

Analysis for regularization with neural networks has been studied in Li et al. (2020) and Haltmeier et al. (2019). In this section, we further investigate the issue. To this end, we consider the approach (T2), where the neural networks are trained independently of the optimization problem (11). That is, $\theta = \theta^*$ is already fixed a priori. For the sake of simplicity, we will drop θ from the notation of \mathcal{R}_θ and \mathcal{D}_θ . We focus on how the problem depends on the regularization parameter α and noise level δ in the data. Such analysis in standard situations is well studied; see, e.g., Scherzer et al. (2009). However, we need to extend the analysis to more general cases to accommodate the fact that \mathcal{R} comes from a neural network and is likely non-convex.

Let us make several assumptions on the regularizer and fidelity term.

Condition 3.

- (A1) Network regularizer $\mathcal{R}: \mathbb{X} \rightarrow [0, \infty]$ satisfies
- (a) $0 \in \text{dom}(\mathcal{R}) := \{\mathbf{x} \mid \mathcal{R}(\mathbf{x}) < \infty\}$;
 - (b) \mathcal{R} is lower semi-continuous;
 - (c) $\mathcal{R}(\cdot)$ is coercive, that is $\mathcal{R}(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\| \rightarrow \infty$.
- (A2) Data consistency term $\mathcal{D}: \mathbf{C} \times \mathbf{C} \rightarrow [0, \infty]$ satisfies
- (a) $\text{dom}(\mathcal{D}(0, \cdot)) = \mathbf{C}$;
 - (b) If $\mathcal{D}(\mathbf{y}_0, \mathbf{y}_1) < \infty$ and $\mathcal{D}(\mathbf{y}_1, \mathbf{y}_2) < \infty$ then $\mathcal{D}(\mathbf{y}_0, \mathbf{y}_2) < \infty$;
 - (c) $\mathcal{D}(\mathbf{y}, \mathbf{y}') = 0 \iff \mathbf{y} = \mathbf{y}'$;
 - (d) $\mathcal{D}(\mathbf{y}, \mathbf{y}') \geq C \|\mathbf{y} - \mathbf{y}'\|^2$ holds in any bounded subset of $\text{dom}(\mathcal{D})$;
 - (e) For any \mathbf{y} , the function $\mathcal{D}(\mathbf{y}, \cdot)$ is continuous and coercive on its domain;

- (f) The functional $(\mathbf{x}, \mathbf{y}) \mapsto \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y})$ is sequentially lower semi-continuous in the weak topology of \mathbb{X} and strong topology of \mathbb{Y} .

For (A1), the coercivity condition (c) is the most restrictive. However, it can be accommodated. One such regularizer is proposed in our recent work Haltmeier et al. (2019) as follows:

$$\mathcal{R}(\mathbf{x}) = \phi(\mathbf{E}(\mathbf{x})) + \frac{\beta}{2} \|\mathbf{x} - (\mathbf{D} \circ \mathbf{E})(\mathbf{x})\|_2^2. \tag{12}$$

Here, $\mathbf{D} \circ \mathbf{E}: \mathbb{X} \rightarrow \mathbb{X}$ is an encoder-decoder network. The regularizer \mathcal{R} is to enforce that a reasonable solution \mathbf{x} satisfies $\mathbf{x} \simeq (\mathbf{D} \circ \mathbf{E})(\mathbf{x})$ and $\phi(\mathbf{E}(\mathbf{x}))$ is small. The term $\phi(\mathbf{E}(\mathbf{x}))$ implements learned prior knowledge, which is normally a sparsity measure in a non-linear basis. The second term $\|\mathbf{x} - (\mathbf{D} \circ \mathbf{E})(\mathbf{x})\|_2^2$ forces \mathbf{x} to be close to data manifold \mathcal{M} . Their combination also guarantees the coercivity of the regularization functional \mathcal{R} . Another choice for \mathcal{R} was suggested in Li et al. (2020).

For (A2), \mathbf{C} is a conic set in \mathbb{Y} . For any $\mathbf{y} \in \mathbf{C}$, we define $\text{dom}(\mathcal{D}(\mathbf{y}, \cdot)) = \{\mathbf{y}' \mid \mathcal{D}(\mathbf{y}, \mathbf{y}') < \infty\}$. The data consistency conditions in (A2) are flexible enough to be satisfied by a few interesting cases. The first example is that $\mathcal{D}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|^2$, which is probably the most popular data consistency measure. Another case is the Kullback-Leibler divergence, which reads as follows. Let $\mathbb{Y} = \mathbb{R}^n$, and $\mathbf{A}: \mathbb{X} \rightarrow \mathbb{Y}$ is a bounded linear positive operator.¹ Consider nonnegative cone $\mathbf{C} = \{(\mathbf{y}_1, \dots, \mathbf{y}_n) \mid \forall i: \mathbf{y}_i \geq 0\}$. We define $\mathcal{D}: \mathbf{C} \times \mathbf{C} \rightarrow [0, \infty]$ by

$$\mathcal{D}(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^n \mathbf{y}_i \log \frac{\mathbf{y}_i}{\mathbf{y}'_i} + \mathbf{y}'_i - \mathbf{y}_i.$$

It is straight forward to check that Condition (A2) is satisfied in this case. In particular, item (d) has been verified in Resmerita and Anderssen (2007, Equation (13)).

To emphasize the fact that our data is the noisy version \mathbf{y}^δ of \mathbf{y} , we rewrite (11) as follows:

$$\arg \min_{\mathbf{x} \in \mathbb{D}} \mathcal{T}_{\mathbf{y}^\delta, \alpha}(\mathbf{x}) := \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}^\delta) + \alpha \mathcal{R}(\mathbf{x}). \tag{13}$$

Here, \mathbb{D} is a weakly closed conic set in \mathbb{X} such that $\mathbf{A}(\mathbb{D}) \subseteq \mathbf{C}$.

Theorem 4 (Well-posedness and convergence). *Let Condition 3 be satisfied. Then the following assertions hold true:*

- (a) Existence: For all $\mathbf{y} \in \mathbf{C}$ and $\alpha > 0$, there exists a minimizer of $\mathcal{T}_{\mathbf{y}, \alpha}$ in \mathbb{D} .

¹ \mathbf{A} is positive if: $\mathbf{y} \geq 0 \Rightarrow \mathbf{A}\mathbf{y} \geq 0$.

- (b) Stability: If $\mathbf{y}_k \rightarrow \mathbf{y}$, $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) < \infty$ and $\mathbf{x}_k \in \arg \min \mathcal{T}_{\alpha; \mathbf{y}_k}$, then weak accumulation points of $(\mathbf{x}_k)_{k \in \mathbb{N}}$ exist and are minimizers of $\mathcal{T}_{\alpha; \mathbf{y}}$.
- (c) Convergence: Let $\mathbf{y} \in \text{ran}(\mathbf{A}) \cap \mathbf{C}$ and $(\mathbf{y}_k)_{k \in \mathbb{N}}$ satisfy $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) \leq \delta_k$ for some sequence $(\delta_k)_{k \in \mathbb{N}} \in (0, \infty)^{\mathbb{N}}$ with $\delta_k \rightarrow 0$. Suppose $\mathbf{x}_k \in \arg \min_{\mathbf{x}} \mathcal{T}_{\mathbf{y}_k, \alpha(\delta_k)}(\mathbf{x})$, and let the parameter choice $\alpha: (0, \infty) \rightarrow (0, \infty)$ satisfy

$$\lim_{\delta \rightarrow 0} \alpha(\delta) = \lim_{\delta \rightarrow 0} \frac{\delta}{\alpha(\delta)} = 0. \tag{14}$$

Then the following holds:

- (1) All weak accumulation points of $(\mathbf{x}_k)_{k \in \mathbb{N}}$ are \mathcal{R} -minimizing solutions of the equation $\mathbf{A}(\mathbf{x}) = \mathbf{y}$;
- (2) $(\mathbf{x}_k)_{k \in \mathbb{N}}$ has at least one weak accumulation point \mathbf{x}^\dagger ;
- (3) Every subsequence $(\mathbf{x}_{k(n)})_{n \in \mathbb{N}}$ that weakly converges to \mathbf{x}^\dagger satisfies $\mathcal{R}(\mathbf{x}_{k(n)}) \rightarrow \mathcal{R}(\mathbf{x}^\dagger)$;
- (4) If the \mathcal{R} -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ is unique, then $\mathbf{x}_k \rightarrow \mathbf{x}^\dagger$.

Before starting the proof, we recall that \mathbf{x}^\dagger is an \mathcal{R} -minimizing solution of the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ if $\mathbf{x}^\dagger \in \arg \min \{ \mathcal{R}(\mathbf{x}) \mid \mathbf{x} \in \mathbb{D} \wedge \mathbf{A}\mathbf{x} = \mathbf{y} \}$.

Proof. (a) First, we observe that $c := \inf_{\mathbf{x}} \mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}) \leq \mathcal{T}_{\mathbf{y}, \alpha}(0) < \infty$. Let $(\mathbf{x}_k)_k$ be a sequence such that $\mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}_k) \rightarrow c$. There exists $M > 0$ such that $\mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}_k) \leq M$, which implies $\alpha \mathcal{R}(\mathbf{x}_k) \leq M$. Since \mathcal{R} is coercive, we obtain that $(\mathbf{x}_k)_k$ is bounded. By passing into a subsequence, $\mathbf{x}_{k_i} \rightarrow \mathbf{x}^* \in \mathbb{D}$. Due to the lower semi-continuity of $\mathcal{T}_{\alpha, \cdot}(\cdot)$, we have $\mathbf{x}^* \in \arg \min \mathcal{T}_{\mathbf{y}, \alpha}$.

(b) Since $\mathbf{x}_k \in \arg \min \mathcal{T}_{\mathbf{y}, \alpha}$, it holds $\mathcal{T}_{\mathbf{y}_k, \alpha}(\mathbf{x}_k) \leq \mathcal{T}_{\mathbf{y}_k, \alpha}(0) = \mathcal{D}(0, \mathbf{y}_k) + \alpha \mathcal{R}(0)$. Thanks to the continuity of $\mathcal{D}(0, \cdot)$ on \mathbf{C} , $(\mathcal{D}(0, \mathbf{y}_k))_k$ is a bounded sequence. Therefore, $\alpha \mathcal{R}(\mathbf{x}_k) \leq \mathcal{T}_{\mathbf{y}_k, \alpha}(\mathbf{x}_k) \leq M$, for a constant M independent of k . Since \mathcal{R} is coercive, $(\mathbf{x}_k)_k$ is bounded and hence has a weakly convergent subsequence $\mathbf{x}_{k_i} \rightarrow \mathbf{x}^\dagger$.

Let us now prove that \mathbf{x}^\dagger is a minimizer of $\mathcal{T}_{\mathbf{y}, \alpha}$. Since $\mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x})$ is lower semi-continuous in \mathbf{x} and \mathbf{y} ,

$$\liminf_{k_i \rightarrow \infty} \mathcal{T}_{\mathbf{y}_{k_i}, \alpha}(\mathbf{x}_{k_i}) \geq \mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}^\dagger). \tag{15}$$

On the other hand, let $\mathbf{x} \in \mathbb{D}$ be such that $\mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}) < \infty$. We obtain $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}) < \infty$ and $\mathcal{R}(\mathbf{x}) < \infty$. Condition (A2)(d) and $\mathcal{D}(\mathbf{y}, \mathbf{y}_k) < \infty$ give $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}_k) < \infty$. That is, $\mathbf{y}_k \in \text{dom}(\mathcal{D}(\mathbf{A}(\mathbf{x}), \cdot))$. The continuity of $\mathcal{D}(\mathbf{A}(\mathbf{x}), \cdot)$ on its domain implies $\mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y}_k) \rightarrow \mathcal{D}(\mathbf{A}(\mathbf{x}), \mathbf{y})$. Since \mathbf{x}_k is the minimizer of $\mathcal{T}_{\mathbf{y}_k, \alpha}$, $\mathcal{T}_{\mathbf{y}_k, \alpha}(\mathbf{x}_k) \leq \mathcal{T}_{\mathbf{y}_k, \alpha}(\mathbf{x})$. Taking the limit, we obtain $\limsup_k \mathcal{T}_{\mathbf{y}_k, \alpha}(\mathbf{x}_k) \leq \mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x})$. From (15), $\mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x}^\dagger) \leq \mathcal{T}_{\mathbf{y}, \alpha}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{D}$. We conclude that \mathbf{x}^\dagger is a minimizer of $\mathcal{T}_{\mathbf{y}, \alpha}$.

(c) We prove the properties item by item.

- (1) Since $\mathbf{y} \in \mathbb{R}(\mathbf{A})$, we can pick solution $\bar{\mathbf{x}}$ of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. We have

$$\mathcal{D}(\mathbf{A}(\mathbf{x}_k), \mathbf{y}_k) + \alpha_k \mathcal{R}(\mathbf{x}_k) \leq \mathcal{D}(\mathbf{y}, \mathbf{y}_k) + \alpha_k \mathcal{R}(\bar{\mathbf{x}}) \leq \delta_k + \alpha_k \mathcal{R}(\bar{\mathbf{x}}). \tag{16}$$

Assume that \mathbf{x}^\dagger is a weak accumulation point of \mathbf{x}_k , then

$$\mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}) \leq \liminf_{k \rightarrow \infty} \mathcal{D}(\mathbf{A}(\mathbf{x}_k), \mathbf{y}_k) \leq \liminf_{k \rightarrow \infty} (\delta_k + \alpha_k \mathcal{R}(\bar{\mathbf{x}})) = 0.$$

Therefore, $\mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}) = 0$ or $\mathbf{A}(\mathbf{x}^\dagger) = \mathbf{y}$. Moreover, $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\bar{\mathbf{x}})$, which implies $\mathcal{R}(\mathbf{x}^\dagger) \leq \liminf \mathcal{R}(\mathbf{x}_k) \leq \mathcal{R}(\bar{\mathbf{x}})$. Since this holds for all possible solution $\bar{\mathbf{x}}$ of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, we conclude that \mathbf{x}^\dagger is a \mathcal{R} -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$.

- (2) Using again the inequality $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\bar{\mathbf{x}})$ and \mathcal{R} is coercive, we obtain that $\{\mathbf{x}_k\}$ is bounded. Therefore, $\{\mathbf{x}_k\}$ has a weak accumulation point \mathbf{x}^\dagger .
- (3) Using (16) again for $\bar{\mathbf{x}} = \mathbf{x}^\dagger$, we obtain $\mathcal{R}(\mathbf{x}_k) \leq \delta_k/\alpha_k + \mathcal{R}(\mathbf{x}^\dagger)$, which gives $\limsup_k \mathcal{R}(\mathbf{x}_k) \leq \mathcal{R}(\mathbf{x}^\dagger)$. This together with the fact that \mathcal{R} is lower semi-continuous gives $\mathcal{R}(\mathbf{x}_{k(n)}) \rightarrow \mathcal{R}(\mathbf{x}^\dagger)$.
- (4) The last conclusion follows straightforwardly from the above three.

□

Let us proceed to obtain some convergence results in the norm. Following Li et al. (2020), we introduce the absolute Bregman distance.

Definition 7 (Absolute Bregman distance). Let $\mathbb{F}: \mathbb{D} \subseteq \mathbb{X} \rightarrow \mathbb{R}$ be Gâteaux differentiable at $\mathbf{x} \in \mathbb{X}$. The *absolute Bregman distance* $\Delta_{\mathbb{F}}(\cdot, \mathbf{x}): \mathbb{D} \rightarrow [0, \infty]$ with respect to \mathbb{F} at \mathbf{x} is defined by

$$\forall \tilde{\mathbf{x}} \in \mathbb{X}: \quad \Delta_{\mathbb{F}}(\tilde{\mathbf{x}}, \mathbf{x}) := \left| \mathbb{F}(\tilde{\mathbf{x}}) - \mathbb{F}(\mathbf{x}) - \mathbb{F}'(\mathbf{x})(\tilde{\mathbf{x}} - \mathbf{x}) \right|. \tag{17}$$

Here $\mathbb{F}'(\mathbf{x})$ denotes the Gâteaux derivative of \mathbb{F} at \mathbf{x} .

From Theorem 4, we can conclude convergence of \mathbf{x}_α^δ to the exact solution in the absolute Bregman distance $\Delta_{\mathcal{R}}$. Below we show that this implies strong convergence under some additional assumption on the regularization functional. For this purpose, we define the concept of total non-linearity, which was introduced in Li et al. (2020).

Definition 8 (Total non-linearity). Let $\mathbb{F}: \mathbb{D} \subseteq \mathbb{X} \rightarrow \mathbb{R}$ be Gâteaux differentiable at $\mathbf{x} \in \mathbb{D}$. We define the *modulus of total non-linearity* of \mathbb{F} at \mathbf{x} as $\nu_{\mathbb{F}}(\mathbf{x}, \cdot) : [0, \infty) \rightarrow [0, \infty]$,

$$\forall t > 0: \quad \nu_{\mathbb{F}}(\mathbf{x}, t) := \inf \{ \Delta_{\mathbb{F}}(\tilde{\mathbf{x}}, \mathbf{x}) \mid \tilde{\mathbf{x}} \in \mathbb{D} \wedge \|\tilde{\mathbf{x}} - \mathbf{x}\| = t \}. \tag{18}$$

The function \mathbb{F} is called *totally non-linear* at \mathbf{x} if $\nu_{\mathbb{F}}(\mathbf{x}, t) > 0$ for all $t \in (0, \infty)$.

The following result, due to Li et al. (2020), connects the convergence in absolute Bregman distance and in norm

Proposition 6. *For $\mathbf{F}: D \subseteq \mathbb{X} \rightarrow \mathbb{R}$ and any $\mathbf{x} \in D$, the followings are equivalent:*

- (i) *The function \mathbf{F} is totally nonlinear at \mathbf{x} ;*
- (ii) *$\forall(\mathbf{x}_n): (\lim_{n \rightarrow \infty} \mathcal{B}_{\mathbf{F}}(\mathbf{x}_n, \mathbf{x}) = 0 \wedge (\mathbf{x}_n) \text{ bounded}) \Rightarrow \lim_{n \rightarrow \infty} \|\mathbf{x}_n - \mathbf{x}\| = 0.$*

As a consequence, we have the following convergence result in the norm topology.

Theorem 5 (Strong convergence). *Assume that $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ has a solution; let \mathcal{R}_θ be totally nonlinear at all \mathcal{R}_θ -minimizing solutions of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, and let $(\mathbf{x}_k)_{k \in \mathbb{N}}, (y_k)_{k \in \mathbb{N}}, (\alpha_k)_{k \in \mathbb{N}}, (\delta_k)_{k \in \mathbb{N}}$ be as in Theorem 4. Then there is a subsequence $(\mathbf{x}_{k(\ell)})_{\ell \in \mathbb{N}}$ of $(\mathbf{x}_k)_{k \in \mathbb{N}}$ and an \mathcal{R}_θ -minimizing solution \mathbf{x}^\dagger of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, such that $\lim_{\ell \rightarrow \infty} \|\mathbf{x}_{k(\ell)} - \mathbf{x}^\dagger\| = 0$. Moreover, if the \mathcal{R}_θ -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ is unique, then $\mathbf{x} \rightarrow \mathbf{x}^\dagger$ in the norm topology.*

We now focus on the convergence rate. To this end, we make the following assumptions:

- (B1) \mathbb{Y} is a finite dimensional space;
- (B2) \mathcal{R} is coercive and weakly sequentially lower semi-continuous;
- (B3) \mathcal{R} is Lipschitz;
- (B4) \mathcal{R} is Gâteaux differentiable.

The most restrictive condition in the above list is that \mathbf{A} has finite-dimensional range. However, this assumption holds true in practical applications such as sparse data tomography, which is the main focus of deep learning techniques for inverse problems. For infinite-dimensional space result, see Li et al. (2020).

We start our analysis with the following result.

Proposition 7. *Let (B1)–(B4) be satisfied and assume that \mathbf{x}^\dagger is an \mathcal{R} -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$. Then there exists a constant $C > 0$ such that*

$$\forall \mathbf{x} \in \mathbb{X}: \quad \Delta_{\mathcal{R}}(\mathbf{x}, \mathbf{x}^\dagger) \leq \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + C\|\mathbf{A}(\mathbf{x}) - \mathbf{A}(\mathbf{x}^\dagger)\|.$$

The proof follows Obmann et al. (2020a). We present it here for the sake of completeness.

Proof. Let us first prove that for some constant $\gamma \in (0, \infty)$, it holds

$$\forall \mathbf{x} \in \mathbb{X}: \quad \mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \leq \gamma\|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\|. \tag{19}$$

Indeed, let \mathbf{P} be the orthogonal projection onto $\ker(\mathbf{A})$ and define $\mathbf{x}_0 = (\mathbf{x}^\dagger - \mathbf{P}(\mathbf{x}^\dagger)) + \mathbf{P}(\mathbf{x})$. Then, we have $\mathbf{A}(\mathbf{x}_0) = \mathbf{A}(\mathbf{x}^\dagger)$ and $\mathbf{x} - \mathbf{x}_0 \in \ker(\mathbf{A})^\perp$. Since the restricted operator $\mathbf{A}|_{\ker(\mathbf{A})^\perp} : \ker(\mathbf{A})^\perp \rightarrow \mathbb{Y}$ is injective and has finite-dimensional range, it is bounded from below by a constant γ_0 . Therefore,

$$\|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\| = \|\mathbf{A}(\mathbf{x}_0) - \mathbf{A}(\mathbf{x})\| = \|\mathbf{A}(\mathbf{x}_0 - \mathbf{x})\| \geq \gamma_0 \|\mathbf{x}_0 - \mathbf{x}\|. \tag{20}$$

On the other hand, since \mathbf{x}^\dagger is the \mathcal{R} -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$ and \mathcal{R} is Lipschitz, we have $\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \leq \mathcal{R}(\mathbf{x}_0) - \mathcal{R}(\mathbf{x}) \leq L\|\mathbf{x}_0 - \mathbf{x}\|$. Together with (20) we obtain (19).

Next we prove that there is a constant γ_1 such that

$$\left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x} \right\rangle \leq \gamma_1 \|\mathbf{A}(\mathbf{x}^\dagger) - \mathbf{A}(\mathbf{x})\|. \tag{21}$$

Indeed, since \mathbf{x}^\dagger is an \mathcal{R} -minimizing solution of $\mathbf{A}(\mathbf{x}) = \mathbf{y}$, we obtain $\left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x}_0 \right\rangle \leq 0$. Therefore,

$$\begin{aligned} \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x} \right\rangle &= \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}^\dagger - \mathbf{x}_0 \right\rangle + \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}_0 - \mathbf{x} \right\rangle \\ &\leq \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x}_0 - \mathbf{x} \right\rangle \leq \|\mathcal{R}'(\mathbf{x}^\dagger)\| \|\mathbf{x}_0 - \mathbf{x}\|. \end{aligned}$$

Using (20), again we obtain (21).

To finish the proof, we consider two cases:

- $\mathcal{R}(\mathbf{x}^\dagger) \leq \mathcal{R}(\mathbf{x}) \Rightarrow \left| \mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \right| = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger)$
- $\mathcal{R}(\mathbf{x}^\dagger) \geq \mathcal{R}(\mathbf{x}) \Rightarrow \left| \mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \right| = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + 2(\mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}))$.

Therefore, using (19) and (21), we obtain

$$\begin{aligned} \Delta_{\mathcal{R}}(\mathbf{x}, \bar{\mathbf{x}}) &\leq \left| \mathcal{R}(\mathbf{x}^\dagger) - \mathcal{R}(\mathbf{x}) \right| + \left| \left\langle \mathcal{R}'(\mathbf{x}^\dagger), \mathbf{x} - \mathbf{x}^\dagger \right\rangle \right| \\ &\leq \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{x}^\dagger) + (2\gamma + \gamma_1) \|\mathbf{A}(\mathbf{x}) - \mathbf{A}(\mathbf{x}^\dagger)\|, \end{aligned}$$

which concludes our proof with $C := 2\gamma_0 + \gamma_1$. □

Here is our convergence rates result, which is an extension of Obmann et al. (2020a, Theorem 3.1).

Theorem 6 (Convergence rates results). *Let (B1)–(B4) be satisfied, and suppose $\alpha \sim \delta$. Then $\Delta_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) = \mathcal{O}(\delta)$ as $\delta \rightarrow 0$.*

Proof. From Proposition 7, we obtain

$$\begin{aligned}
 \alpha \Delta_{\mathbf{F}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) &\leq \alpha \mathcal{R}(\mathbf{x}_\alpha^\delta) - \alpha \mathcal{R}(\mathbf{x}^\dagger) + C\alpha \|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{A}(\mathbf{x}^\dagger)\| \\
 &= \mathcal{T}_{\alpha,\delta}(\mathbf{x}_\alpha^\delta) - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) - \left(\mathcal{T}_{\alpha,\delta}(\mathbf{x}^\dagger) - \mathcal{D}(\mathbf{A}(\mathbf{x}^\dagger), \mathbf{y}^\delta) \right) \\
 &\quad + C\alpha \|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{A}(\mathbf{x}^\dagger)\| \\
 &\leq \delta^2 + C\alpha\delta - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) + C\alpha \|\mathbf{A}(\mathbf{x}_\alpha^\delta) - \mathbf{y}^\delta\| \\
 &\leq \delta^2 + C\alpha\delta - \mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta) + C\alpha \sqrt{\mathcal{D}(\mathbf{A}(\mathbf{x}_\alpha^\delta), \mathbf{y}^\delta)}.
 \end{aligned}$$

Cauchy’s inequality gives $\alpha \Delta_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) \leq \delta^2 + C\alpha\delta + C^2\alpha^2/4$. For $\alpha \sim \delta$, we easily conclude $\Delta_{\mathcal{R}}(\mathbf{x}_\alpha^\delta, \mathbf{x}^\dagger) = \mathcal{O}(\delta)$. □

Related Methods

The use of neural networks as regularizers or similarity measures is an active research direction. Many interesting works have been done. We briefly review several techniques: variational networks (Kobler et al. 2017), deep cascaded networks (Kofler et al. 2018; Schlemper et al. 2017), and the MODL approach (Aggarwal et al. 2018). Further, we propose INDIE as a new operator-inversion-free variant of MODL. As opposed to the discussion in section “Convergence Analysis”, these works make use of the approach (T1): employing solver-dependent training. Finally, we will discuss a synthesis variant of the NETT framework.

Variational networks: Variational networks (Kobler et al. 2017) connect variational methods and deep learning. They are based on the fields of experts model (Roth and Black 2005) and consider the Tikhonov functional

$$\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) = \sum_{c=1}^{N_c} \mathcal{T}_c(\mathbf{x}) := \sum_{c=1}^{N_c} \left(\sum_j \sum_i \phi_i^c((\bar{K}_i^c \mathbf{x})_j) + \alpha \sum_j \sum_i \psi_i^c((K_i^c(\mathbf{A}(\mathbf{x}) - \mathbf{y}))_j) \right),$$

where \bar{K}_i^c and K_i^c are learnable convolutional operators, and ϕ_i, ψ_i are learnable functionals. Alternating gradient descent method for minimizing $\mathcal{T}_{\mathbf{y},\alpha}$ provides the update formula

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla_\theta \mathcal{T}_{c(n)}(\mathbf{x}_n) \quad \text{where } c(n) = 1 + (n \bmod N_c). \tag{22}$$

Direct calculations show $\nabla_\theta \mathcal{T}_c(\mathbf{x}) = \sum_i (\bar{K}_i^c)^T (\phi_i^c)'(K_i^c \mathbf{x}) + \mathbf{A}^T \sum_i (K_i^c)^T (\psi_i^c)'(K_i^c(\mathbf{A}(\mathbf{x}) - \mathbf{y}))$. Minimizing the $\mathcal{T}_{\mathbf{y},\alpha}$ is then replaced by training the neural network that consists of a L blocks realizing the iterative update (22).

Network cascades: Deep network cascades (Kofler et al. 2018; Schlemper et al. 2017) alternate between the application of post-processing networks and so-called

data consistency layers. The data consistency condition proposed in Kofler et al. (2018) for sparse data problems $\mathbf{A} = \mathbf{S} \circ \mathbf{A}_F$, where \mathbf{S} is a sampling operator and \mathbf{A}_F a full data forward operator (such as the fully sampled Radon transform), takes the form

$$\mathbf{x}_{n+1} = \mathbf{B}_F \left(\arg \min_{\mathbf{z}} \|\mathbf{z} - \mathbf{A}_F(\mathcal{N}_{\theta(n)}(\mathbf{x}_n))\|_2^2 + \alpha \|\mathbf{y} - \mathbf{S}(\mathbf{z})\|_2^2 \right), \tag{23}$$

with initial reconstruction $\mathbf{x}_0 = (\mathbf{B}_F \circ \mathbf{S}^*)(\mathbf{y})$, where $\mathbf{B}_F: \mathbb{Y} \rightarrow \mathbb{X}$ is a reconstruction method for the full data forward operator and $\mathcal{N}_{\theta(n)}$ are networks. For example, in MRT the operator \mathbf{B}_F is the inverse Fourier transform (Schlemper et al. 2017), and in CT, the operator \mathbf{B}_F can be implemented by the filtered backprojection (Kofler et al. 2018). The resulting neural network consists of L steps of (23) that can be trained end-to-end.

MODL approach: The model-based deep learning (MODL) approach of Aggarwal et al. (2018) starts with the Tikhonov functional $\mathcal{T}_{\mathbf{y},\alpha}(\mathbf{x}) = \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|_2^2 + \alpha \|\mathbf{x} - \mathcal{N}_{\theta}(\mathbf{x})\|_2^2$, where $\mathcal{N}_{\theta}(\mathbf{x})$ is interpreted as denoising network. By designing \mathcal{N}_{θ} as a convolutional block, then $\mathbf{x} - \mathcal{N}_{\theta}(\mathbf{x})$ is a small residual network (He et al. 2016). The authors of Aggarwal et al. (2018) proposed the following heuristic iterative scheme $\mathbf{x}_{n+1} = \arg \min_{\mathbf{x}} \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|_2^2 + \alpha \|\mathbf{x} - \mathcal{N}_{\theta}(\mathbf{x}_n)\|_2^2$ based on $\mathcal{T}_{\mathbf{y},\alpha}$ whose closed-form solution is

$$\mathbf{x}_{n+1} = (\mathbf{A}^T \mathbf{A} + \alpha \text{Id})^{-1} (\mathbf{A}^T \mathbf{y} + \alpha \mathcal{N}_{\theta}(\mathbf{x}_n)). \tag{24}$$

Concatenating these steps together, one arrives at a deep neural network. Similar to network cascades, each block (24) consists of a trainable layer $\mathbf{z}_n = \mathbf{A}^T \mathbf{y} + \alpha \mathcal{N}_{\theta}(\mathbf{x}_n)$ and a non-trainable data consistency layer $\mathbf{x}_{n+1} = (\mathbf{A}^T \mathbf{A} + \lambda \text{Id})^{-1}(\mathbf{z}_n)$.

INDIE approach: Let us present an alternative to the above procedures, inspired by Daubechies et al. (2004). Namely, we propose the iterative update

$$\begin{aligned} \mathbf{x}_{n+1} &= \arg \min \mathcal{L}_n(\mathbf{x}) \\ \mathcal{L}_n(\mathbf{x}) &:= \|\mathbf{A}(\mathbf{x}) - \mathbf{y}\|^2 + \alpha \|\mathbf{x} - \mathcal{N}_{\theta}(\mathbf{x}_n)\|^2 + C \|\mathbf{x} - \mathbf{x}_n\|^2 - \|\mathbf{A}(\mathbf{x} - \mathbf{x}_n)\|^2. \end{aligned}$$

Here the constant $C > 0$ is an upper bound for the operator norm $\|\mathbf{A}\|$. Elementary manipulations show the identity

$$\begin{aligned} \mathcal{L}_n(\mathbf{x}) &= -2 \left\langle \mathbf{A}^T(\mathbf{y} - \mathbf{A}(\mathbf{x}_n)) + \alpha \mathcal{N}_{\theta}(\mathbf{x}_n) + C \mathbf{x}_n, \mathbf{x} \right\rangle \\ &\quad + (\alpha + C) \|\mathbf{x}\|^2 - (\alpha \|\mathcal{N}_{\theta}(\mathbf{x}_n)\| + C \|\mathbf{x}_n\|^2) - \|\mathbf{A}(\mathbf{x}_n)\|^2 + \|\mathbf{y}\|^2. \end{aligned}$$

The minimizer of \mathcal{L}_n can therefore be computed explicitly by setting the gradient of the latter expression to zero. This results in the proposed network block

$$\mathbf{x}_{n+1} = \frac{1}{\alpha + C} \left(\mathbf{A}^\top (\mathbf{y} - \mathbf{A}(\mathbf{x}_n)) + \alpha \mathcal{N}_\theta(\mathbf{x}_n) + C \mathbf{x}_n \right). \quad (25)$$

This results at a deep neural network similar to the MODL iteration. However, each block in (25) is clearly simpler than the blocks in (24). In fact, as opposed to MODL, our proposed learned iterative scheme does not require costly matrix inversion. We name the resulting iteration INDIE (for **in**version-free **deep** **iterative**) cascades. We consider the numerical comparison of MODL and INDIE as well as the theoretical analysis of both architectures to be interesting lines of future research.

Learned synthesis regularization: Let us finish this section by pointing out that regularization by neural network is not restricted to the form (11). For example, one can consider the synthesis version, which reads (Obmann et al. 2020b)

$$\mathbf{x}^{\text{syn}} = \mathbf{D}_\theta \left(\arg \min_{\xi} \|\mathbf{A} \circ \mathbf{D}_\theta(\xi) - \mathbf{y}\|^2 + \alpha \sum_{\lambda \in \Lambda} \omega_\lambda |\xi_\lambda|^p \right), \quad (26)$$

where Λ is a countable set, $1 \leq p < 2$, and $\mathcal{D}_\theta: \ell^2(\Lambda) \rightarrow \mathbb{X}$ is a learned operator that performs nonlinear synthesis of \mathbf{x} . Rigorous analysis of the above formulation was derived in Obmann et al. (2020b).

Finally, note that one can generalize the frameworks (11) and (26) by allowing the involved neural networks to depend on the regularization parameter α or the noise-level δ . The dependence on α has been studied in, for example, Obmann et al. (2020b). The dependence on δ can be realized by mimicking the Morozov's stopping criteria, when training the neural networks, either independently or together with the optimization problem. In the later case, δ can help decide the depth of the unrolled neural network.

Conclusion and Outlook

Inverse problems are central to solving a wide range of important practical problems within and outside of imaging and computer vision. Inverse problems are characterized by the ambiguity and instability of their solution. Therefore, stabilizing solution methods based on regularization techniques is necessary to solve them in a reasonable way. In recent years, neural networks and deep learning have emerged as the rising stars for the solution of inverse problems. In this chapter, we have developed the mathematical foundations for solving inverse problems with deep learning. In addition, we have shown stability and convergence for selected neural networks to solve inverse problems. The investigated methods, which combine

the strengths of both worlds, are regularizing null-space networks and the NETT (Network-Tikhonov) approach for inverse problems.

References

- Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Probl.* **10**(6), 1217–1229 (1994)
- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**(12), 124007 (2017)
- Aggarwal, H.K., Mani, M.P., Jacob, M.: MoDL: model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imaging* **38**(2), 394–405 (2018)
- Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks. arXiv:1412.6830 (2014)
- Aljadaany, R., Pal, D.K., Savvides, M.: Douglas-rachford networks: learning both the image prior and data fidelity terms for blind image deconvolution. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 10235–10244 (2019)
- Arridge, S., Maass, P., Öktem, O., Schönlieb C.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal.* **35**(8), 1798–1828 (2013)
- Boink, Y.E., Haltmeier, M., Holman, S., Schwab, J.: Data-consistent neural networks for solving nonlinear inverse problems. arXiv:2003.11253 (2020), to appear in *Inverse Probl. Imaging*
- Bora, A., Jalal, A., Price, E., Dimakis, A.G.: Compressed sensing using generative models. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 537–546 (2017)
- Brosch, T., Tam, R., et al.: Manifold learning of brain MRIs by deep learning. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 633–640. Springer (2013)
- Bubba, T.A., Kutyniok, G., Lassas, M., Maerz, M., Samek, W., Siltanen, S., Srinivasan, V.: Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.* **35**(6), 064002 (2019)
- Chen, D., Davies, M.E.: Deep decomposition learning for inverse imaging problems. In *European Conference on Computer Vision*, pp. 510–526. Springer, Cham (2020)
- Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: *Advances in Neural Information Processing Systems*, pp. 6571–6583 (2018)
- Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**(11), 1413–1457 (2004)
- Dittmer, S., Maass, P.: A projectional ansatz to reconstruction. arXiv:1907.04675 (2019)
- Dittmer, S., Kluth, T., Maass, P., Bagger, D.O.: Regularization by architecture: a deep prior approach for inverse problems. *J. Math. Imaging Vis.* **62**, 456–470 (2020)
- Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*, vol. 375. Kluwer Academic Publishers Group, Dordrecht (1996)
- Georg, M., Souvenir, R., Hope, A., Pless, R.: Manifold learning for 4D CT reconstruction of the lung. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE (2008)
- Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256 (2010)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, London (2016)
- Grasmair, M., Haltmeier, M., Scherzer, O.: Sparse regularization with l^q penalty term. *Inverse Probl.* **24**(5), 055020 (2008)

- Han, Y., Yoo, J.J., Ye, J.C.: Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis (2016). <http://arxiv.org/abs/1611.06391>
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- Huang, Y., Preuhs, A., Manhart, M., Lauritsch, G., Maier, A.: Data consistent ct reconstruction from insufficient data with learned prior images. arXiv:2005.10034 (2020)
- Ivanov, V.K., Vasin, V.V., Tanana, V.P.: Theory of Linear Ill-Posed Problems and Its Applications. Inverse and Ill-Posed Problems Series, 2nd edn. VSP, Utrecht, (2002). Translated and revised from the 1978 Russian original
- Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. IEEE Trans. Image Process. **26**(9), 4509–4522 (2017)
- Kobler, E., Klatzer, T., Hammernik, K., Pock, T.: Variational networks: connecting variational methods and deep learning. In: German Conference on Pattern Recognition, pp. 281–293. Springer (2017)
- Kofler, A., Haltmeier, M., Kolbitsch, C., Kachelrieß, M., Dewey, M.: A U-Nets cascade for sparse view computed tomography. In: Proceedings of 1st Workshop on Machine Learning for Medical Image Reconstruction, pp. 91–99. Springer (2018)
- Kofler, A., Haltmeier, M., Schaeffter, T., Kachelrieß, M., Dewey, M., Wald, C., Kolbitsch, C.: Neural networks-based regularization of large-scale inverse problems in medical imaging. Phys. Med. Biol. **65**, 135003 (2020)
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
- Li, H., Schwab, J., Antholzer, S., Haltmeier, M.: NETT: solving inverse problems with deep neural networks. Inverse Probl. **36**, 065005 (2020)
- Lindenstrauss, J., Tzafriri, L.: On the complemented subspaces problem. Israel J. Math. **9**(2), 263–269 (1971)
- Lunz, S., Öktem, O., Schönlieb, C.: Adversarial regularizers in inverse problems. In: Advances in Neural Information Processing Systems, vol. 31, pp. 8507–8516 (2018)
- Mardani, M., Gong, E., Cheng, J.Y., Vasanawala, S.S., Zaharchuk, G., Xing, L., Pauly, J.M.: Deep generative adversarial neural networks for compressive sensing MRI. IEEE Trans. Med. Imag. **38**(1), 167–179 (2018)
- Nashed, M.Z.: Inner, outer, and generalized inverses in banach and hilbert spaces. Numer. Func. Anal. Opt. **9**(3–4), 261–325 (1987)
- Obmann, D., Nguyen, L., Schwab, J., Haltmeier, M.: Sparse aNETT for solving inverse problems with deep learning. In: 2020 IEEE 17th International Symposium on Biomedical Imaging Workshops (ISBI Workshops) (pp. 1–4). IEEE (2020a)
- Obmann, D., Schwab, J., Haltmeier, M.: Deep synthesis network for regularizing inverse problems. Inverse Problems, **37**(1), 015005 (2020b)
- Obmann, D., Nguyen, L., Schwab, J., Haltmeier, M.: Augmented NETT regularization of inverse problems. J. Phys. Commun. **5**(10), 105002 (2021)
- Phillips, R.S.: On linear transformations. Trans. Am. Math. Soc. **48**(3), 516–541 (1940)
- Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv:1710.05941 (2017)
- Resmerita, E., Anderssen, R.S.: Joint additive Kullback–Leibler residual minimization and regularization for linear inverse problems. Math. Methods Appl. Sci. **30**(13), 1527–1544 (2007)
- Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 860–867. IEEE (2005)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Applied Mathematical Sciences, vol. 167. Springer, New York (2009)
- Schlemper, J., Caballero, J., Hajnal, J.V., Price, A., Rueckert, D.: A deep cascade of convolutional neural networks for MR image reconstruction. In: Proceedings of Information Processing in Medical Imaging, pp. 647–658. Springer (2017)

- Schwab, J., Antholzer, S., Haltmeier, M.: Deep null-space learning for inverse problems: convergence analysis and rates. *Inverse Probl.* **35**(2), 025008 (2019)
- Schwab, J., Antholzer, S., Haltmeier, M.: Big in Japan: regularizing networks for solving inverse problems. *J. Math. Imaging Vis.* **62**, 445–455 (2020)
- Sulam, J., Aberdam, A., Beck, A., Elad, M.: On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(8), 1968–1980 (2019)
- Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454 (2018)
- Van Veen, D., Jalal, A., Soltanolkotabi, M., Price, E., Vishwanath, S., Dimakis, A.G.: Compressed sensing with deep image prior and learned regularization. *arXiv:1806.06438* (2018)
- Wachinger, C., Yigitsoy, M., Rijkhorst, E., Navab, N.: Manifold learning for image-based breathing gating in ultrasound and MRI. *Med. Image Anal.* **16**(4), 806–818 (2012)
- Yang, Y., Sun, J., Li, H., Xu, Z.: Deep ADMM-net for compressive sensing MRI. In: *Proceedings of 30th International Conference on Neural Information Processing Systems*, pp. 10–18 (2016)



Gitta Kutyniok

Contents

Introduction	1096
The Applied Harmonic Analysis Viewpoint	1096
Frame Theory Comes into Play	1097
Wavelets	1098
From Wavelets to Shearlets	1098
From Inverse Problems to Deep Learning	1099
Outline	1100
Continuous Shearlet Systems	1100
Classical Continuous Shearlet Systems	1101
Cone-Adapted Continuous Shearlet Systems	1103
Resolution of the Wavefront Set	1104
Discrete Shearlet Systems	1105
Cone-Adapted Discrete Shearlet Systems	1105
Frame Properties	1106
Sparse Approximation	1109
Extensions of Shearlets	1111
Higher Dimensions	1111
α -Molecules	1113
Universal Shearlets	1114
Digital Shearlet Systems	1116
Digital 2D Shearlet Transform	1116
Extensions of the Digital 2D Shearlet Transform and ShearLab3D	1118
Applications of Shearlets	1119
Sparse Regularization Using Shearlets	1120
Shearlets Meet Deep Learning	1124
Conclusion	1129
References	1129

G. Kutyniok (✉)
Ludwig-Maximilians-Universität München, Mathematisches Institut, München, Germany
e-mail: kutyniok@math.lmu.de

Abstract

Many important problem classes are governed by anisotropic features, which typically appear as singularities concentrated on lower-dimensional embedded manifolds. Examples include edges in images or shock fronts in solutions of transport-dominated equations. Shearlets are the first representation system which exhibits optimal sparse approximation properties in combination with a unified treatment of the continuum and digital realm, leading to faithful implementations. A prominent class of applications are inverse problems, foremost in imaging science, where shearlets are utilized for sparse regularization. Recently, shearlet systems have also been used in combination with data-driven approaches, predominately deep neural networks. This chapter shall serve as an introduction to and a survey about the theory of shearlets and their applications.

Keywords

Deep neural networks · Frames · Shearlets · Sparse approximation · Wavelets

Introduction

In the twenty-first century, technological advances have generated an unprecedented deluge of highly complex data sets, posing enormous challenges to provide efficient methodologies for acquisition and analysis. While there exists a huge variety of different types of data, the majority of it falls into the category of images and videos. Prominent examples of areas in science producing massive data sets of this type are astronomy, medicine, or seismology. One key problem to tackle is the question of suitable representations of such data. This led to an intense study in the research community of applied harmonic analysis aiming to provide highly efficient multivariate encoding methodologies.

The Applied Harmonic Analysis Viewpoint

The viewpoint of applied harmonic analysis concerning the application of representation systems in data processing can be summarized as follows: Let C be a class of data in a Hilbert space \mathcal{H} , and assume $(\psi_\lambda)_{\lambda \in \Lambda} \subset \mathcal{H}$ is a carefully constructed collection of vectors with Λ being a countable indexing set.

On the one hand, $(\psi_\lambda)_{\lambda \in \Lambda}$ can then be utilized to *decompose* the data by

$$C \ni f \mapsto (\langle \cdot, \psi_\lambda \rangle)_{\lambda \in \Lambda}. \quad (1)$$

This can be regarded as an encoding step, often aiming to reveal important features of the data f such as singularities by analyzing the associated coefficient sequence. On the other hand, $(\psi_\lambda)_{\lambda \in \Lambda}$ can also serve as a means to *expand* the data by representing it as

$$f = \sum_{\lambda \in \Lambda} c(f)_\lambda \psi_\lambda \quad \text{for all } f \in \mathcal{C}. \quad (2)$$

Since efficient expansions are typically desirable, one usually aims for the coefficient sequence $(c(f)_\lambda)_{\lambda \in \Lambda}$ to be sparse in the sense of rapid decay to allow efficient encoding of the data f .

In case that $(\psi_\lambda)_{\lambda \in \Lambda}$ forms an orthonormal basis, it is well-known that $c(f)_\lambda = \langle f, \psi_\lambda \rangle$ for all $\lambda \in \Lambda$. However, it might not be possible to design an orthonormal basis with the desirable properties, or redundancy is for other reasons such as robustness required. This then leads to the notion of a frame, in which case (1) and (2) cannot be that easily linked, but requires methods from frame theory.

Frame Theory Comes into Play

The area of frame theory focuses on redundant representation systems in the sense of nonunique expansions, thereby going beyond the concept of orthonormal bases. It provides a general framework for redundant systems $(\psi_\lambda)_{\lambda \in \Lambda}$ while allowing to control their stability.

A system $(\psi_\lambda)_{\lambda \in \Lambda}$ is called a *frame* for \mathcal{H} , if there exist constants $0 < A \leq B < \infty$ such that

$$A \|f\|^2 \leq \sum_{\lambda \in \Lambda} |\langle f, \psi_\lambda \rangle|^2 \leq B \|f\|^2 \quad \text{for all } f \in \mathcal{H}.$$

In case $A = B = 1$, it is coined a *Parseval frame*. In fact, referring to section “[The Applied Harmonic Analysis Viewpoint](#)”, Parseval frames are the most general systems which can satisfy $c(f)_\lambda = \langle f, \psi_\lambda \rangle$ for all $\lambda \in \Lambda$.

The associated *frame operator* is defined by

$$S : \mathcal{H} \rightarrow \mathcal{H}, \quad f \mapsto \sum_{\lambda \in \Lambda} \langle f, \psi_\lambda \rangle \psi_\lambda,$$

which is self-adjoint with spectrum $\sigma(S) \subset [A, B]$. The sequence $(\tilde{\psi}_\lambda)_{\lambda \in \Lambda} := (S^{-1} \psi_\lambda)_{\lambda \in \Lambda}$ is then referred to as the *canonical dual frame*. It allows reconstruction of some $f \in \mathcal{H}$ from the decomposition (1) and the construction of an explicit coefficient sequence in the expansion (2) by considering

$$f = \sum_{\lambda \in \Lambda} \langle f, \psi_\lambda \rangle \tilde{\psi}_\lambda \quad \text{and} \quad f = \sum_{\lambda \in \Lambda} \langle f, \tilde{\psi}_\lambda \rangle \psi_\lambda \quad \text{for all } f \in \mathcal{H},$$

respectively. The coefficient sequence $(\langle f, \tilde{\psi}_\lambda \rangle)_{\lambda \in \Lambda}$ can even be shown to be the smallest in ℓ_2 norm among all possible ones.

For further information on frame theory, we refer to Casazza et al. (2012) and Christensen (2003).

Wavelets

One first highlight in applied harmonic analysis was the development of the system of wavelets, based on translation ($x \mapsto x - m$) and dilation ($x \mapsto 2^j x$) leading to the representation of functions in $L^2(\mathbb{R}^d)$ at different locations and different resolution levels.

Definition 1. For $\psi^1, \dots, \psi^L \in L^2(\mathbb{R}^d)$, the associated (*discrete*) *wavelet system* is defined by

$$\{\psi_{j,m}^\ell = 2^{\frac{dj}{2}} \psi^\ell(2^j \cdot -m) : j \in \mathbb{Z}, m \in \mathbb{Z}^d, \ell = 1, \dots, L\}. \quad (3)$$

The generating functions ψ^1, \dots, ψ^L can be chosen such that the associated wavelet system forms an orthonormal basis (more generally, a frame) for $L^2(\mathbb{R}^d)$. The functions ψ^1, \dots, ψ^L are then typically referred to as *wavelets* with the parameter j serving as *scale* and m as *position*. In fact, one key aspect of wavelet theory which has significantly contributed to its success is its rich mathematical structure. This allows to design families of wavelets with various desirable properties expressed in terms of regularity, decay, or vanishing moments. On the application side, wavelets have revolutionized various areas such as imaging science, for instance, for compression tasks by developing JPEG2000, and numerical analysis of partial differential equations.

The literature on wavelets is very rich, and for the sake of brevity, we here just refer to the books Cohen (2003), Daubechies (1992), Mallat (1998), and references therein.

From Wavelets to Shearlets

Multivariate functions are distinctively different from univariate functions, since they are, in particular, typically governed by anisotropic (i.e., directional) singularities. Let us exemplarily mention that indeed edges are prominent features in images similar to shock fronts in the solutions of transport-dominated equations. More generally, in high-dimensional data information is often contained in lower-dimensional embedded manifolds. Thus, it is fair to say that a system which aims for efficient encoding of such data should, in particular, be able to efficiently encode anisotropic features.

Although wavelets can be shown to optimally encode functions governed by point singularities in the sense of decay rates of the error of best N -term approximation, it is evident that due to their isotropic structure, they are not capable to efficiently encode anisotropic features. Indeed the isotropic scaling matrix with a dyadic scaling factor 2^j (see (3)) prevents a wavelet system from delivering optimal approximation rates of such data.

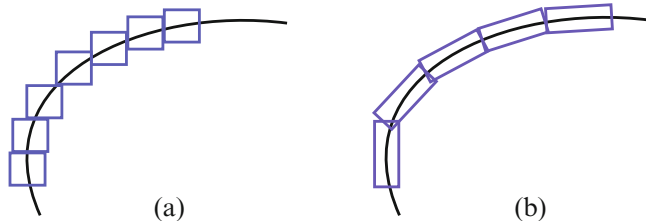


Fig. 1 (a) Approximation of a curvilinear structure by isotropic elements. (b) Approximation of a curvilinear structure by anisotropic elements

This argumentation shows the need to develop anisotropic representation systems, by going beyond systems consisting of translation and dilation. Figure 1 depicts the problem of an isotropic system such as a wavelet system as opposed to the advantage of anisotropically shaped elements. A list of desirable properties for an anisotropic representation system can be summarized as follows:

- (1) Underlying group structure for availability of deep mathematical tools.
- (2) Provably optimal sparse approximations of anisotropic features.
- (3) Compactly supported analyzing elements for high spatial localization.
- (4) Uniform treatment of the continuum and digital realm.
- (5) Fast implementation of the associated decomposition.

This problem has led to the development of various novel anisotropic representation systems within the area of applied harmonic analysis. Some of the key contributions are *steerable pyramid* by Simoncelli et al. (1992), *directional filter banks* by Bamberger and Smith (1992), *2D directional wavelets* by Antoine et al. (1993), *curvelets* by Candès and Donoho (2004), *contourlets* by Do and Vetterli (2005), *bandelet*s by Le Pennec and Mallat (2005), and *shearlets* (Guo et al. 2006; Labate et al. 2005). Shearlet systems indeed satisfy all desiderata one commonly requires from an anisotropic system as stated before.

From Inverse Problems to Deep Learning

The main application areas of shearlets are inverse problems, foremost in imaging. A common approach to solve an ill-posed inverse problem $Tf = g$ for a linear, bounded operator $T : \mathcal{H} \rightarrow \mathcal{H}$ is by Tikhonov regularization. A generalization of this conceptual approach to sparse regularization was suggested in Daubechies et al. (2004). Given a representation system $(\psi_\lambda)_{\lambda \in \Lambda}$, an approximation of the solution can be computed by minimizing the functional

$$\|Tf - g\|^2 + \beta \cdot \|(\langle \cdot, \psi_\lambda \rangle)_{\lambda \in \Lambda}\|_{\ell_1}, \quad (4)$$

with β being the regularization parameter. This approach exploits the fact that, when carefully designing the system $(\psi_\lambda)_{\lambda \in \Lambda}$, the solution of $Tf = g$ exhibits a sparse coefficient sequence $(\langle \cdot, \psi_\lambda \rangle)_{\lambda \in \Lambda}$. Exemplary general inverse problems are inpainting (Genzel and Kutyniok 2014; King et al. 2014), morphological component analysis (Donoho and Kutyniok 2013; Kutyniok and Lim 2012) and segmentation (Häuser and Steidl 2013) or inverse problems from medical diagnosis such as magnetic resonance imaging (Kutyniok and Lim 2018).

Recently, deep learning has swept the area of imaging science with deep neural network-based approaches often outperforming the to-date state-of-the-art algorithms. The last years though have shown that in fact hybrid methods, i.e., combinations of model-based and data-driven approaches, typically lead to the best results by taking the best out of both worlds. Since the shearlet representation is particularly well suited to analyze anisotropic features, several hybrid approaches were suggested which combine the shearlet transform with deep neural networks such as for limited-angle computed tomography (Bubba et al. 2019) as well as for wavefront set and semantic edge detection (Andrade-Loarca et al. 2020, 2019).

In the following, we will provide an introduction to and a survey about the theory and applications of shearlets. For additional information, we refer to Kutyniok and Labate (2012).

Outline

We start by discussing continuous shearlet systems and their associated transforms in section “[Continuous Shearlet Systems](#)”, including their ability to resolve the wavefront set. This is followed by the introduction of their discrete counterparts with a presentation of their optimal sparse approximation properties for anisotropic features (see section “[Discrete Shearlet Systems](#)”). Section “[Extensions of Shearlets](#)” is devoted to extensions of shearlet systems such as extensions to higher dimensions, α -molecules, and universal shearlets. The faithful digitalization as also implemented in www.ShearLab.org is then presented in section “[Digital Shearlet Systems](#)”. Finally, in section “[Applications of Shearlets](#)” applications of shearlets to inverse problems, also in combination with deep learning, are discussed.

Continuous Shearlet Systems

We start by introducing the main notation and the definition of continuous shearlets. Shearlet systems are composed of three operators, namely, scaling, shearing, and translation, applied to a generating function, related to different resolution levels, orientations, and positions, respectively. The term “continuous” indicates that continuous parameter sets are considered. Notice that also the continuous shearlet system and associated transform can be generalized in a canonical way to $L^2(\mathbb{R}^n)$ for $n \geq 3$ with the results from sections “[Classical Continuous Shearlet Systems](#)”

and “Cone-Adapted Continuous Shearlet Systems” holding in a similar manner (Dahlke et al. 2008, 2009, 2010, 2013).

Classical Continuous Shearlet Systems

We will first present the classical version of continuous shearlet systems. For this, let the *parabolic scaling matrix* A_a , $a \in \mathbb{R}^* := \mathbb{R} \setminus \{0\}$ and the *shearing matrix* S_s , $s \in \mathbb{R}$, be given by

$$A_a = \begin{pmatrix} a & 0 \\ 0 & |a|^{1/2} \end{pmatrix} \quad \text{and} \quad S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}, \quad (5)$$

respectively. Letting now the *dilation operator* $D_M : L^2(\mathbb{R}^2) \rightarrow L^2(\mathbb{R}^2)$, $M \in \mathbb{R}^{2 \times 2}$, be defined by

$$(D_M f)(x) \mapsto |\det(M)|^{-1/2} f(M^{-1}x)$$

and the *translation operator* $T_t : L^2(\mathbb{R}^2) \rightarrow L^2(\mathbb{R}^2)$, $t \in \mathbb{R}^2$, by $(T_t f)(x) \mapsto f(x - t)$ yields the definition of continuous shearlet systems.

Definition 2. For $\psi \in L^2(\mathbb{R}^2)$, the *continuous shearlet system* $\mathcal{SH}(\psi)$ is defined by

$$\mathcal{SH}(\psi) = \{\psi_{a,s,t} := T_t D_{A_a} D_{S_s} \psi = a^{-3/4} \psi(A_a^{-1} S_s^{-1}(x-t)) : a \in \mathbb{R}^*, s \in \mathbb{R}, t \in \mathbb{R}^2\},$$

and the associated *continuous shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is given by

$$SH_\psi f(a, s, t) := \langle f, \psi_{a,s,t} \rangle, \quad (a, s, t) \in \mathbb{R}^* \times \mathbb{R} \times \mathbb{R}^2.$$

This transform is invertible provided ψ satisfies an admissibility condition, whose definition requires to take a group theoretic viewpoint. We now endow $\mathbb{R}^* \times \mathbb{R} \times \mathbb{R}^2$ with a group structure, namely, the (full) *shearlet group* $\mathbb{S} := \mathbb{R}^* \times \mathbb{R} \times \mathbb{R}^2$ with group operation given by

$$(a, s, t) \circ (a', s', t') = (aa', s + \sqrt{|a|}s', t + S_s A_a t').$$

This is a locally compact group with left Haar measure $d_\mu(a, s, t) = da/|a|^3 ds dt$ (Dahlke et al. 2009). The map from \mathbb{S} into the group of unitary operators on $L^2(\mathbb{R}^2)$, $\mathcal{U}(L^2(\mathbb{R}^2))$, given by $(a, s, t) \mapsto \psi_{a,s,t}$ can now be regarded as a unitary representation of the shearlet group. This allows to analyze square-integrability of this mapping, i.e., irreducibility and the existence of a nontrivial *admissible function* $\psi \in L^2(\mathbb{R}^2)$ which, for all $f \in L^2(\mathbb{R}^2)$, satisfies the *admissibility condition*

$$\int_{\mathbb{S}} |\langle f, \psi_{a,s,t} \rangle|^2 d\mu(a, s, t) < \infty.$$

A function is then defined to be a shearlet, if a condition equivalent to the admissibility condition is fulfilled.

Definition 3. A function $\psi \in L^2(\mathbb{R}^2)$ is called a *shearlet*, if

$$\int_{\mathbb{R}^2} \frac{|\hat{\psi}(\xi)|^2}{|\xi_1|^2} d\xi < \infty,$$

where $\xi = (\xi_1, \xi_2)$ and $\hat{\psi}$ denote the Fourier transform of ψ .

This leads to the following result, which heavily relies on group theoretic arguments:

Theorem 1 (Dahlke et al. 2008). Let $\psi \in L^2(\mathbb{R}^2)$ be a shearlet. Then

$$SH_{\psi} : L^2(\mathbb{R}^2) \rightarrow L^2(\mathbb{S}), \quad f \mapsto SH_{\psi} f(a, s, t)$$

is an isometry.

Let us now consider some examples of shearlets. The first and most extensively studied shearlet is the so-called classical shearlet, which is a band-limited function introduced in Labate et al. (2005). For an illustration of the support of the associated Fourier transform, we refer to Fig. 2a.

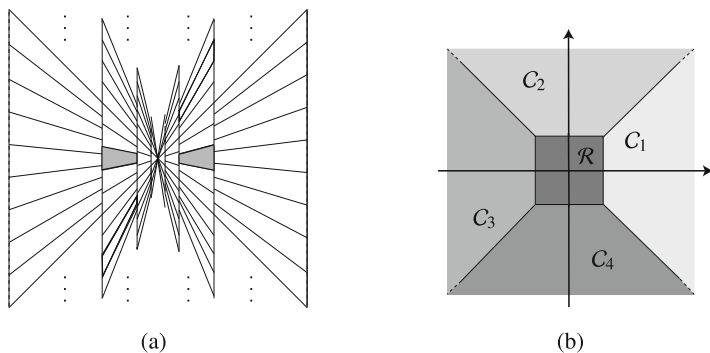


Fig. 2 (a) Partitioning of Fourier domain by supports of several elements of the classical shearlet system, with the support of the Fourier transform of the classical shearlet itself being highlighted. (b) The partition of Fourier domain into four conic regions $C_1 - C_4$ and a centered rectangle $\mathcal{R} = \{(\xi_1, \xi_2) : |\xi_1|, |\xi_2| \leq 1\}$ as the low-frequency regime

Example 1. A classical shearlet $\psi \in L^2(\mathbb{R}^2)$ is defined by

$$\hat{\psi}(\xi) = \hat{\psi}(\xi_1, \xi_2) = \hat{\psi}_1(\xi_1) \hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right),$$

where $\psi_1 \in L^2(\mathbb{R})$ is a wavelet, i.e., it satisfies the discrete Calderón condition given by

$$\sum_{j \in \mathbb{Z}} |\hat{\psi}_1(2^{-j}\xi)|^2 = 1 \quad \text{for a.e. } \xi \in \mathbb{R},$$

with $\hat{\psi}_1 \in C^\infty(\mathbb{R})$ and $\text{supp } \hat{\psi}_1 \subseteq [-\frac{5}{4}, -\frac{1}{4}] \cup [\frac{1}{4}, \frac{5}{4}]$, and $\psi_2 \in L^2(\mathbb{R})$ is a ‘‘bump function,’’ namely,

$$\sum_{k=-1}^1 |\hat{\psi}_2(\xi + k)|^2 = 1 \quad \text{for a.e. } \xi \in [-1, 1],$$

satisfying $\hat{\psi}_2 \in C^\infty(\mathbb{R})$ and $\text{supp } \hat{\psi}_2 \subseteq [-1, 1]$.

In general, shearlets of both band-limited and compactly supported type have been constructed and analyzed (see, e.g., Dahlke et al. 2008, 2011 and Grohs 2011b). We would also like to remark, that by using coorbit space theory, associated smoothness spaces can be derived together with their atomic decompositions and (Banach) frames for these spaces, (see, e.g., Dahlke et al. (2009, 2011) and Labate et al. (2013)).

Cone-Adapted Continuous Shearlet Systems

The group-theoretic approach leading to continuous shearlet systems allows to directly apply various results and methodologies from abstract harmonic analysis. This approach is however problematic, since it leads to a directional bias of the system in the sense of an imbalance of the directional sensitivity; for an illustration, we refer to Fig. 2a. This creates a problem when shearlet systems are, for instance, applied to resolve the wavefront set.

To circumvent this issue, cone-adapted continuous shearlet systems were constructed. The key idea is to decompose the Fourier domain in a suitable way which enforces a more balanced decomposition of the different directions as depicted in Fig. 2b. In the following definition, notice that the system $\Psi(\psi)$ is associated with the horizontal cones $C_1 \cup C_3$, whereas choosing the shearlet $\tilde{\psi}$ with the roles of ξ_1 and ξ_2 reversed, i.e., $\tilde{\psi}(\xi_1, \xi_2) = \psi(\xi_2, \xi_1)$, the system $\tilde{\Psi}(\tilde{\psi})$ is then associated with the vertical cones $C_2 \cup C_4$.

Definition 4. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the *cone-adapted continuous shearlet system* is defined by $\mathcal{SH}(\phi, \psi, \tilde{\psi}) = \Phi(\phi) \cup \Psi(\psi) \cup \tilde{\Psi}(\tilde{\psi})$, where

$$\Phi(\phi) = \{\phi_t = \phi(\cdot - t) : t \in \mathbb{R}^2\},$$

$$\Psi(\psi) = \{\psi_{a,s,t} = a^{-\frac{3}{4}} \psi(A_a^{-1} S_s^{-1}(\cdot - t)) : a \in (0, 1], |s| \leq 1 + a^{1/2}, t \in \mathbb{R}^2\},$$

$$\tilde{\Psi}(\tilde{\psi}) = \{\tilde{\psi}_{a,s,t} = a^{-\frac{3}{4}} \tilde{\psi}(\tilde{A}_a^{-1} S_s^{-T}(\cdot - t)) : a \in (0, 1], |s| \leq 1 + a^{1/2}, t \in \mathbb{R}^2\},$$

and $\tilde{A}_a = \text{diag}(a^{1/2}, a)$.

The associated transform can be defined in a similar manner as before in the pure group-theoretic approach.

Definition 5. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, let $\mathcal{SH}(\phi, \psi, \tilde{\psi})$ be the associated cone-adapted continuous shearlet system. Then the associated *cone-adapted continuous shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is given by

$$\mathcal{SH}_{\phi, \psi, \tilde{\psi}} f(a, s, t, \iota) := \begin{cases} \langle f, \psi_{a,s,t} \rangle : \iota = -1, \\ \langle f, \phi_t \rangle : \iota = 0, \\ \langle f, \tilde{\psi}_{a,s,t} \rangle : \iota = 1. \end{cases}$$

where $(a, s, t) \in \mathbb{R}^* \times \mathbb{R} \times \mathbb{R}^2$.

We mention that this transform satisfies similar isometry properties as the continuous shearlet transform (cf. Kutyniok and Labate 2009).

Resolution of the Wavefront Set

The ability of a (cone-adapted) continuous shearlet system to resolve different directions can be analyzed using the notion of a wavefront set from microlocal analysis. Coarsely speaking, a wavefront set consists of the elements of the singular support of a distribution together with the directions in which the singularity propagates. For more details on microlocal analysis and wavefront sets, we refer to Hörmander (2003).

Definition 6. Let f be a distribution. Then a point $(x, \lambda) \in \mathbb{R}^2 \times \mathbb{S}^1$ is a *regular directed point* of f , if there exist open neighborhoods U_x and U_λ of x and λ , respectively, and a smooth function $\phi \in C^\infty(\mathbb{R}^2)$ with $\text{supp} \phi \subset U_x$ and $\phi(x) = 1$ such that

$$|\widehat{\phi f}(\xi)| \leq C_k (1 + |\xi|)^{-k} \quad \text{for all } \xi \in \mathbb{R}^2 \setminus \{0\} \text{ such that } \xi/|\xi| \in V_\lambda$$

holds for some $C_k > 0$. The *wavefront set* $WF(f)$ is then defined as the complement of the set of all regular directed points.

The notion of wavefront allows us to derive a precise statement about the resolution of different directions by cone-adapted continuous shearlet systems.

Theorem 2 (Kutyniok and Labate 2009). *Let $\psi \in L^2(\mathbb{R}^2)$ be a shearlet, and $f \in L^2(\mathbb{R}^2)$. Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, where $\mathcal{D}_1 = \{(t_0, s_0) \in \mathbb{R}^2 \times [-1, 1] : \text{for } (s, t) \text{ in a neighborhood } U \text{ of } (s_0, t_0), |SH_{\phi, \psi, \tilde{\psi}} f(a, s, t, -1)| = O(a^k) \text{ as } a \rightarrow 0, \text{ for all } k \in \mathbb{N}, \text{ with the } O(\cdot)\text{-term uniform over } (s, t) \in U\}$ and $\mathcal{D}_2 = \{(t_0, s_0) \in \mathbb{R}^2 \times [1, \infty) : \text{for } (\frac{1}{s}, t) \text{ in a neighborhood } U \text{ of } (s_0, t_0), |SH_{\phi, \psi, \tilde{\psi}} f(a, s, t, 1)| = O(a^k) \text{ as } a \rightarrow 0, \text{ for all } k \in \mathbb{N}, \text{ with the } O(\cdot)\text{-term uniform over } (\frac{1}{s}, t) \in U\}$. Then*

$$WF(f)^c = \mathcal{D}.$$

An extension of this result to a more general class of shearlets $\psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ was derived in Grohs (2011a). Stronger results in the sense of more precise decay estimates can be found in Guo et al. (2009) for the band-limited case and in Kutyniok and Petersen (2017) for the compactly supported case.

Discrete Shearlet Systems

Discrete shearlet systems are derived by sampling the parameter set of continuous shearlet systems. Thus, similar to continuous shearlet systems, both a “classical” and a cone-adapted variant are available. Due to the fact that the first variant in the discrete setting not only is incapable of detecting the horizontal direction precise – only asymptotically – but also faces numerical instabilities due to the occurrence of arbitrarily small support sets, we will focus in the sequel only on the cone-adapted variant.

Cone-Adapted Discrete Shearlet Systems

The discretization of the parameter sets of parabolic scaling and shearing as defined in (5) is typically performed by choosing A_{2^j} and S_k with $j, k \in \mathbb{Z}$. Coorbit theory (cf. section “Classical Continuous Shearlet Systems”) then yields the discretization (for $c \in (\mathbb{R}_+)^2$ to add flexibility)

$$(a, s, t) \mapsto (2^{-j}, -k2^{-j/2}, A_{2^j}^{-1} S_k^{-1} cm),$$

which when applied to Definition 4 leads to the following definition of a cone-adapted discrete shearlet system: The association of the different subsystems with the conic regions from Fig. 2b evidently carries over the discrete situation.

Definition 7. Let $c = (c_1, c_2) \in (\mathbb{R}_+)^2$. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the cone-adapted discrete shearlet system $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c)$ is defined by

$$\begin{aligned} \Phi(\phi; c_1) &= \{\phi_m := \phi(\cdot - m) : m \in c_1\mathbb{Z}^2\}, \\ \Psi(\psi; c) &= \{\psi_{j,k,m} := 2^{\frac{3}{4}j} \psi(S_k A_{2^j} \cdot -m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in M_c \mathbb{Z}^2\}, \\ \tilde{\Psi}(\tilde{\psi}; c) &= \{\tilde{\psi}_{j,k,m} := 2^{\frac{3}{4}j} \tilde{\psi}(S_k^T \tilde{A}_{2^j} \cdot -m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \tilde{M}_c \mathbb{Z}^2\}, \end{aligned}$$

where $\tilde{A}_{2^j} = \text{diag}(2^{j/2}, 2^j)$, $M_c = \text{diag}(c_1, c_2)$ and $\tilde{M}_c = \text{diag}(c_2, c_1)$. If $c = (1, 1)$, we also use the notions $\Phi(\phi)$, $\Psi(\psi)$, and $\tilde{\Psi}(\tilde{\psi})$.

One often refers to ϕ as a *scaling function* and to the functions ψ and $\tilde{\psi}$ as (*discrete*) *shearlets*.

As in the continuous setting, we can also define an associated transform, which arises as a discretization of the continuous version.

Definition 8. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ and $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, let $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c)$ be the associated cone-adapted discrete shearlet system. Then the associated cone-adapted discrete shearlet transform of $f \in L^2(\mathbb{R}^2)$ is given by

$$\mathcal{SH}_{\phi, \psi, \tilde{\psi}} f(j, k, m, \iota) := \begin{cases} \langle f, \psi_{j,k,m} \rangle : \iota = -1, j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in M_c \mathbb{Z}^2, \\ \langle f, \phi_m \rangle : \iota = 0, m \in c_1 \mathbb{Z}^2, \\ \langle f, \tilde{\psi}_{j,k,m} \rangle : \iota = 1, j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \tilde{M}_c \mathbb{Z}^2. \end{cases}$$

The tiling of Fourier domain provided by the cone-adapted discrete shearlet system with classical shearlets as generating functions is depicted in Fig. 3a. Figure 3b shows a classical shearlet in spatial domain. One notices the “needlelike” structure of the function, which is of size $2^{-j} \times 2^{-j/2}$, hence becoming even more anisotropic shaped as $j \rightarrow \infty$.

Frame Properties

It is evident that the frame properties of a cone-adapted discrete shearlet system are closely linked to the chosen shearlets. One can identify band-limited shearlets and compactly supported shearlets as the two main classes of shearlets. Some applications such as seismology have a natural band-limited structure which then makes the first type of shearlets preferable, whereas other applications might require high spatial localization, which requires the second type.

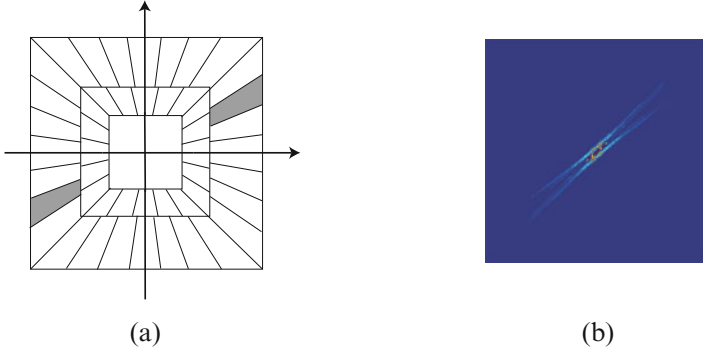


Fig. 3 (a) Partitioning of Fourier domain by the cone-adapted discrete shearlet system with classical shearlets as generating functions. (b) One shearlet in spatial domain

Band-Limited Shearlets

Classical shearlets as introduced in Example 1 are the most well-known type of band-limited shearlets. With slight modifications, the associated cone-adapted discrete shearlet system forms a Parseval frame for $L^2(\mathbb{R}^2)$.

Theorem 3 (Guo et al. 2006). *Let $C = \{(\xi_1, \xi_2) \in \mathbb{R}^2 : |\xi_2/\xi_1| \leq 1\}$ and $\tilde{C} = \mathbb{R}^2 \setminus C$ with P_C and $P_{\tilde{C}}$ denoting the associated orthogonal projections in $L^2(\mathbb{R}^2)$. Further, let $\psi \in L^2(\mathbb{R}^2)$ be a classical shearlet, let $\tilde{\psi}(\xi_1, \xi_2) = \psi(\xi_2, \xi_1)$, and let $\phi \in L^2(\mathbb{R}^2)$ be chosen so that, for a.e. $\xi \in \mathbb{R}^2$,*

$$|\hat{\phi}(\xi)|^2 + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} |\hat{\psi}(S_{-k}^T A_{2^{-j}} \xi)|^2 \chi_C + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} |\hat{\tilde{\psi}}(S_{-k} \tilde{A}_{2^{-j}} \xi)|^2 \chi_{\tilde{C}} = 1.$$

Then the modified cone-adapted discrete shearlet system $\Phi(\phi) \cup P_C \Psi(\psi) \cup P_{\tilde{C}} \tilde{\Psi}(\tilde{\psi})$ is a Parseval frame for $L^2(\mathbb{R}^2)$.

Refinements of this result leading to a smooth Parseval frame were derived in Guo and Labate (2013) and Bodmann et al. (2019). Moreover, in Grohs (2013), constructions of band-limited shearlet frames with dual frames such that both frames possess distinctive time-frequency localization properties were provided.

Compactly Supported Shearlets

Despite the advantage of high spatial localization, the construction of a Parseval frame is not as straightforward as in the band-limited case. In fact, it is still not clear whether a cone-adapted discrete shearlet system associated with compactly supported shearlets can be introduced, which forms a Parseval frame for $L^2(\mathbb{R}^2)$. Despite this obstacle, various constructions of compactly supported shearlets were suggested which yield cone-adapted discrete shearlet frames for $L^2(\mathbb{R}^2)$ with

numerically proven ratio of the frame bounds of approximately 4, hence sufficiently stable from a numerical standpoint.

We now describe the general framework introduced in Kittipoom et al. (2012) for deriving sufficient conditions for cone-adapted discrete shearlet systems to form a frame alongside with theoretical estimates for the associated frame bounds. We start by defining the rectangle Ω_0 and the conic region Ω_1 by

$$\Omega_0 = \{\xi \in \mathbb{R}^2 : \max\{|\xi_1|, |\xi_2|\} \leq \frac{1}{2}\}, \quad \Omega_1 = \{\xi \in \mathbb{R}^2 : \frac{1}{2} < |\xi_2| < 1, |\xi_2|/|\xi_1| < 1\}.$$

Letting $\psi \in L^2(\mathbb{R}^2)$ and $\phi \in L^2(\mathbb{R}^2)$, we assume that

$$\operatorname{ess\,inf}_{\xi \in \Omega_0} |\hat{\phi}(\xi)| > 0 \quad \text{and} \quad \operatorname{ess\,inf}_{\xi \in \Omega_1} |\hat{\psi}(\xi)| > 0. \tag{6}$$

Setting $\tilde{\psi}(x_1, x_2) = \psi(x_2, x_1)$, it can be shown that those conditions ensure

$$\operatorname{ess\,inf}_{\xi \in \mathbb{R}^2} |\hat{\phi}(\xi)|^2 + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} (|\hat{\psi}(S_k^T A_{2^{-j}} \xi)|^2 + |\hat{\tilde{\psi}}(\tilde{S}_k^T \tilde{A}_{2^{-j}} \xi)|^2) > 0.$$

The following result then proves that, provided the Fourier transforms of the scaling function and shearlets decay fast enough with sufficient vanishing moments and satisfy (6), we obtain a shearlet frame $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c)$.

Theorem 4 (Kittipoom et al. 2012). *Let $\phi, \psi \in L^2(\mathbb{R}^2)$ be functions such that*

$$\begin{aligned} \hat{\phi}(\xi_1, \xi_2) &\leq C_1 \cdot \min\{1, |\xi_1|^{-\gamma}\} \cdot \min\{1, |\xi_2|^{-\gamma}\} \quad \text{and} \\ |\hat{\tilde{\psi}}(\xi_1, \xi_2)| &\leq C_2 \cdot \min\{1, |\xi_1|^\alpha\} \cdot \min\{1, |\xi_1|^{-\gamma}\} \cdot \min\{1, |\xi_2|^{-\gamma}\}, \end{aligned} \tag{7}$$

for some positive constants $C_1, C_2 < \infty$ and $\alpha > \gamma > 3$. Define $\tilde{\psi}(x_1, x_2) = \psi(x_2, x_1)$ and assume that ϕ, ψ satisfy (6). Then, there exists some positive constant c^* such that $\mathcal{SH}(\phi, \psi, \tilde{\psi}, c)$ forms a frame for $L^2(\mathbb{R}^2)$ for any $c = (c_1, c_2)$ with $\max\{c_1, c_2\} \leq c^*$.

For various explicit constructions of compactly supported shearlets leading to numerically stable cone-adapted discrete shearlet systems, we refer to Kittipoom et al. (2012). Let us further remark that this theorem can also be applied to band-limited cone-adapted discrete shearlet systems, since band-limited shearlets trivially satisfy condition (7).

Sparse Approximation

Recalling the goal to derive suitable decompositions (1) and efficient representations (2) of data, we will now show that within a certain model setting, shearlets can be proven to serve for both tasks in an optimal way.

For this, we first focus on the approximation properties of shearlets and introduce the related basic notions of approximation theory. Given a class of functions and a representation system, one main goal of approximation theory is to analyze the suitability of this system for uniformly approximating functions from this class. This leads to the notion of best N -term approximation.

Definition 9. Letting $N \in \mathbb{N}$, $C \subseteq L^2(\mathbb{R}^2)$ be a class of functions and $(\psi_\lambda)_{\lambda \in \Lambda} \subset L^2(\mathbb{R}^2)$ be a representation system, we call $f_N \in L^2(\mathbb{R}^2)$ *best N -term approximation of f* , if

$$\|f - f_N\|_{L^2} \leq \|f - g\|_{L^2} \quad \text{for all } g = \sum_{\lambda \in \Lambda_N} c_\lambda \psi_\lambda, \quad \text{where } \#\Lambda_N = N, \quad \Lambda_N \subseteq \Lambda.$$

The *error of best N -term approximation* of some $f \in C$ is then given by $\|f - f_N\|_{L^2}$. The largest $\gamma > 0$ such that

$$\sup_{f \in C} \|f - f_N\|_{L^2} = O(N^{-\gamma}) \quad \text{as } N \rightarrow \infty$$

determines the *optimal (sparse) approximation rate* of C by $(\psi_\lambda)_{\lambda \in \Lambda}$.

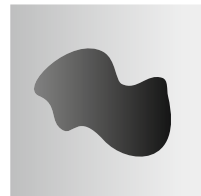
Thus, the optimal (sparse) approximation rate relates approximation accuracy with the complexity of the approximating system in terms of sparsity.

We discussed earlier that one key aspect of multivariate functions is the fact that they are typically governed by anisotropic features. The model class of cartoonlike functions introduced by Donoho in (2001) makes this mathematically precise. We refer to Fig. 4 for an illustration.

Definition 10. For fixed $\nu > 0$, the *class \mathcal{E}_ν^2 of cartoonlike functions* is the set of functions $f : \mathbb{R}^2 \rightarrow \mathbb{C}$ of the form

$$f = f_0 + f_1 \chi_B,$$

Fig. 4 Example of a cartoonlike function



where $B \subset [0, 1]^2$ with ∂B being a closed C^2 -curve with curvature bounded by ν as well as $f_i \in C^2(\mathbb{R}^2)$ with $\text{supp } f_i \subset [0, 1]^2$ and $\|f_i\|_{C^2} \leq 1$ for each $i = 0, 1$.

The optimal (sparse) approximation rate for cartoonlike functions was proven by Donoho as well and can be stated in the situation of frames as follows. We wish to emphasize that the original result is proven for more general function systems.

Theorem 5 (Donoho 2001). *Let $(\psi_\lambda)_{\lambda \in \Lambda}$ be a frame for $L^2(\mathbb{R}^2)$. Then the optimal asymptotic approximation error of $f \in \mathcal{E}_\nu^2$ is given by*

$$\|f - f_N\|_{L^2} \leq C \cdot N^{-1} \quad \text{as } N \rightarrow \infty,$$

with f_N being a best N -term approximation of f and $C > 0$.

This benchmark result allows to make the phrase “optimal sparse approximations of cartoonlike functions” mathematically precise, namely, being justified in case a representation system does satisfy this rate. Indeed, it can be proven that, under weak assumptions on the generating functions, cone-adapted discrete shearlet systems associated with compactly supported shearlets provide this optimal rate up to a log-factor, which is typically assumed to be negligible.

Theorem 6 (Kutyniok and Lim 2011). *Let $c > 0$, and let $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ be compactly supported. Suppose that, in addition, for all $\xi = (\xi_1, \xi_2) \in \mathbb{R}^2$, the shearlet ψ satisfies*

- (i) $|\hat{\psi}(\xi)| \leq C_1 \cdot \min\{1, |\xi_1|^\alpha\} \cdot \min\{1, |\xi_1|^{-\gamma}\} \cdot \min\{1, |\xi_2|^{-\gamma}\}$ and
- (ii) $\left| \frac{\partial}{\partial \xi_2} \hat{\psi}(\xi) \right| \leq |h(\xi_1)| \cdot \left(1 + \frac{|\xi_2|}{|\xi_1|}\right)^{-\gamma},$

where $\alpha > 5, \gamma \geq 4, h \in L^1(\mathbb{R})$, and C_1 are constant, and suppose that the shearlet $\tilde{\psi}$ satisfies (i) and (ii) with the roles of ξ_1 and ξ_2 reversed. Further, suppose that $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c)$ forms a frame for $L^2(\mathbb{R}^2)$. Then, for any $\nu > 0$, the shearlet frame $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c)$ provides (almost) optimal sparse approximations of functions $f \in \mathcal{E}_\nu^2$ in the sense that there exists some $C > 0$ such that

$$\|f - f_N\|_{L^2} \leq C \cdot N^{-1} \cdot (\log N)^{\frac{3}{2}} \quad \text{as } N \rightarrow \infty,$$

where f_N is the nonlinear N -term approximation obtained by choosing the N largest shearlet coefficients of f .

A similar result can also be derived in the setting of band-limited shearlets (Guo and Labate 2007).

Extensions of Shearlets

Several extensions of the described discrete shearlet systems were developed. In the sequel, we will discuss shearlet systems for arbitrary dimensions, α -molecules, and universal shearlets. Besides those, other generalizations of shearlets include irregular discrete shearlet frames arising from a different type of sampling of continuous shearlet systems (Kittipoom et al. 2011) and bendlets, which can be regarded as a second-order shearlet system (Lessig et al. 2019).

Higher Dimensions

We will first describe the extension to the three-dimensional situation, i.e., to derive a frame for $L^2(\mathbb{R}^3)$. In this situation, the four cones will be replaced by six pyramids again leading to a uniform way to treat the different directions. Accordingly, we define *paraboloidal scaling matrices* A_{2^j} , \tilde{A}_{2^j} and \check{A}_{2^j} , $j \in \mathbb{Z}$ by

$$A_{2^j} = \text{diag}(2^j, 2^{j/2}, 2^{j/2}), \quad \tilde{A}_{2^j} = \text{diag}(2^{j/2}, 2^j, 2^{j/2}), \quad \text{and} \quad \check{A}_{2^j} = \text{diag}(2^{j/2}, 2^{j/2}, 2^j)$$

as well as *shear matrices* S_k , \tilde{S}_k , and \check{S}_k , $k = (k_1, k_2) \in \mathbb{Z}^2$ by

$$S_k = \begin{pmatrix} 1 & k_1 & k_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{S}_k = \begin{pmatrix} 1 & 0 & 0 \\ k_1 & 1 & k_2 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \check{S}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ k_1 & k_2 & 1 \end{pmatrix}.$$

The lattices in \mathbb{R}^3 for defining translation are chosen as $M_c = \text{diag}(c_1, c_2, c_2)$, $\tilde{M}_c = \text{diag}(c_2, c_1, c_2)$, and $\check{M}_c = \text{diag}(c_2, c_2, c_1)$, where $c_1, c_2 > 0$.

The discrete shearlet system is then defined according to a partition of the Fourier domain into a rectangular region and six pyramids (see Fig. 5) similar to the conic regions of cone-adapted discrete shearlet systems.

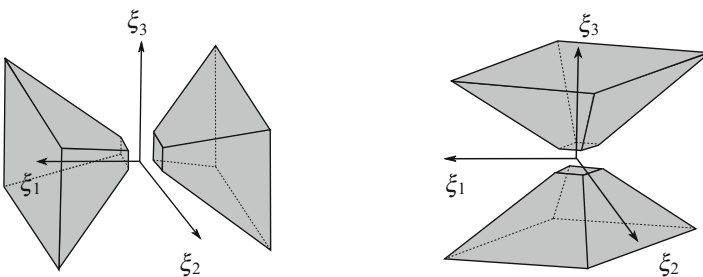


Fig. 5 The partition of Fourier domain by four of the six pyramids

The definition of pyramid-adapted discrete shearlet systems can now be stated as follows: Each part of the system is responsible for covering one set of pyramids, similar to covering the set of cones in Definition 7.

Definition 11. For $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, the *pyramid-adapted discrete shearlet system* $\mathcal{SH}(\phi, \psi, \tilde{\psi}, \check{\psi}; c)$ generated by $\phi, \psi, \tilde{\psi}, \check{\psi} \in L^2(\mathbb{R}^3)$ is defined by

$$\mathcal{SH}(\phi, \psi, \tilde{\psi}, \check{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c) \cup \check{\Psi}(\check{\psi}; c),$$

where

$$\begin{aligned} \Phi(\phi; c_1) &= \{\phi_m := \phi(\cdot - m) : m \in c_1\mathbb{Z}^3\}, \\ \Psi(\psi; c) &= \{\psi_{j,k,m} := 2^j \psi(S_k A_{2^j} \cdot -m) : j \geq 0, \|k\|_\infty \leq \lceil 2^{j/2} \rceil, m \in M_c \mathbb{Z}^3\}, \\ \tilde{\Psi}(\tilde{\psi}; c) &= \{\tilde{\psi}_{j,k,m} := 2^j \tilde{\psi}(\tilde{S}_k \tilde{A}_{2^j} \cdot -m) : j \geq 0, \|k\|_\infty \leq \lceil 2^{j/2} \rceil, m \in \tilde{M}_c \mathbb{Z}^3\}, \\ \check{\Psi}(\check{\psi}; c) &= \{\check{\psi}_{j,k,m} := 2^j \check{\psi}(\check{S}_k \check{A}_{2^j} \cdot -m) : j \geq 0, \|k\|_\infty \leq \lceil 2^{j/2} \rceil, m \in \check{M}_c \mathbb{Z}^3\}. \end{aligned}$$

The sufficient conditions for a cone-adapted discrete shearlet system to form a frame for $L^2(\mathbb{R}^2)$ as stated in Theorem 4 can be extended to the shearlet systems from Definition 11 (see Kutyniok et al. (2012)). The notion of cartoonlike functions (Definition 10) was in Kutyniok et al. (2012) suitably extended as well by considering a 3D body with a surface singularity, leading to a benchmark result as Theorem 5 now with the rate $N^{-\frac{1}{2}}$. In Kutyniok et al. (2012), it could then be proven that, similar to Theorem 6, also pyramid-adapted discrete shearlet systems lead to (almost) optimal sparse approximations of cartoonlike functions.

We also wish to mention that in fact a generalization of the definition of a discrete shearlet system, the frame properties, and the sparse approximation result to $L^2(\mathbb{R}^d)$, for $d \in \mathbb{N}$ is similarly possible, in this case the optimal rate being $N^{-\frac{1}{d-1}}$ (Kutyniok et al. 2012). In fact, the crucial step is from dimension 2 to 3, since this is the first time that anisotropic features of different dimensions can occur, namely, in this situation filament-like and sheetlike structures. Why do we then have just one type of shearlets for $L^2(\mathbb{R}^3)$? In fact, the shearlet elements we defined are in the spatial domain of size $2^{-j} \times 2^{-j/2} \times 2^{-j/2}$, making them “platelike” as $j \rightarrow \infty$. A different, seemingly also valid strategy would be to consider the scaling matrix $A_{2^j} = \text{diag}(2^j, 2^j, 2^{j/2})$ with similar changes for \tilde{A}_{2^j} and \check{A}_{2^j} , leading to “needlelike” shearlet elements (of size $2^{-j} \times 2^{-j} \times 2^{-j/2}$) as $j \rightarrow \infty$. However, a shearlet system consisting of such “needlelike” shearlet elements lacks frame properties, making them unattractive for image processing. Moreover, the sparse approximation results show that even when introducing one-dimensional singularities as well, the rate is always determined by the singularities of the largest dimension.

α -Molecules

Several anisotropic representation systems based on parabolic scaling such as band-limited shearlets, compactly supported shearlets, and also second-generation curvelets provide (almost) optimal sparse approximations of cartoonlike functions (cf. Theorem 6), proven on a case-by-case basis. This raises the question whether such approximation results hold for a much more general class of anisotropic systems. In fact, the unified framework of *parabolic molecules* introduced in Grohs and Kutyniok (2014) provides such a general class, encompassing all known anisotropic frame constructions based on parabolic scaling. It allows to transfer approximation results from one system to another, thereby enabling that all the desirable approximation properties of shearlets can be deduced for virtually any other system based on parabolic scaling (see Grohs and Kutyniok (2014)).

The framework of parabolic molecules was even further generalized to α -molecules (Grohs et al. 2016a), which, for instance, also includes ridgelets and wavelets. In this approach, the parameter α measures the degree of anisotropy. The conceptual idea relies on the introduction of a general *parameter space*

$$\mathbb{P} := \mathbb{R}_+ \times \mathbb{T} \times \mathbb{R}^2,$$

where $(s, \theta, x) \in \mathbb{P}$ describes scale 2^s , orientation θ , and location x , and a flexibly applicable *parametrization* defined as a pair (Λ, Φ_Λ) , where Λ is a discrete index set and Φ_Λ is a mapping

$$\Phi_\Lambda : \begin{cases} \Lambda \rightarrow \mathbb{P}, \\ \lambda \mapsto (s_\lambda, \theta_\lambda, x_\lambda). \end{cases}$$

This allows the definition of α -molecules, which includes a variety of anisotropic systems such as ridgelets for $\alpha = 0$, curvelets and shearlets for $\alpha = \frac{1}{2}$, and wavelets for $\alpha = 1$. It also includes α -shearlets, which are defined as a cone-adapted discrete shearlet system with the scaling depending on α and suitable adaption of shearing (Kutyniok et al. 2012).

Definition 12. Let $\alpha \in [0, 1]$, and let (Λ, Φ_Λ) be a parametrization. Then $(m_\lambda)_{\lambda \in \Lambda}$ is a system of α -molecules of order $(L, M, N_1, N_2) \in (\mathbb{Z}_+ \cup \{\infty\})^2 \times \mathbb{Z}_+^2$, if, for all $\lambda \in \Lambda$,

$$m_\lambda(x) = s_\lambda^{(1+\alpha)/2} a^{(\lambda)} (A_{\alpha, s_\lambda} R_{\theta_\lambda} (x - x_\lambda)), \quad \Phi_\Lambda(\lambda) = (s_\lambda, \theta_\lambda, x_\lambda),$$

such that, for all $|\beta| \leq L$,

$$|\partial^\beta \hat{a}^{(\lambda)}(\xi)| \lesssim \min\left(1, s_\lambda^{-1} + |\xi_1| + s_\lambda^{-(1-\alpha)} |\xi_2|\right)^M \left(1 + |\xi|^2\right)^{-\frac{N_1}{2}} \left(1 + \xi_2^2\right)^{-\frac{N_2}{2}}.$$

We next state the key result enabling the transfer of sparse approximation results from one system to all other systems within this framework for the same α . It provides an estimate for the decay of the entries of the cross-Gramian matrix away from the main diagonal, which requires an appropriate notion of distance. For this, let (Λ, Φ_Λ) and $(\tilde{\Lambda}, \Phi_{\tilde{\Lambda}})$ be parametrizations. For $\lambda \in \Lambda$ and $\mu \in \tilde{\Lambda}$, we then define the *index distance* by

$$\omega(\lambda, \mu) := \omega(\Phi_\Lambda(\lambda), \Phi_{\tilde{\Lambda}}(\mu)) := 2^{|s_\lambda - s_\mu|} \left(1 + 2^{\arg\min(s_\lambda, s_\mu)} d(\lambda, \mu) \right),$$

where

$$d(\lambda, \mu) := |\theta_\lambda - \theta_\mu|^2 + |x_\lambda - x_\mu|^2 + |(\cos(\theta_\lambda), \sin(\theta_\lambda))^\top, x_\lambda - x_\mu|.$$

This allows us to now formulate the result.

Theorem 7 (Grohs and Kutyniok 2014; Grohs et al. 2016a). *Let $\alpha \in [0, 1]$, $N > 0$, and let $(m_\lambda)_{\lambda \in \Lambda}$, $(p_\mu)_{\mu \in \tilde{\Lambda}}$ be systems of α -molecules of order (L, M, N_1, N_2) with*

$$L \geq 2N, \quad M > 3N - \frac{3 - \alpha}{2}, \quad N_1 \geq N + \frac{1 + \alpha}{2}, \quad N_2 \geq 2N.$$

Then, for all $\lambda \in \Lambda$ and $\mu \in \tilde{\Lambda}$,

$$\left| \langle m_\lambda, p_\mu \rangle \right| \lesssim \omega(\lambda, \mu)^{-N}.$$

For detailed results concerning frame properties of parabolic molecules as well as the more general α -molecules and sufficient conditions for those to provide optimal sparse approximation properties up to a log-factor, we refer to Grohs and Kutyniok (2014) and Grohs et al. (2016a,b).

Universal Shearlets

Another extension of cone-adapted discrete shearlet systems are *universal shearlets* introduced in Genzel and Kutyniok (2014), which provide even more flexibility in the type of scaling than α -molecules. In fact, universal shearlets allow a different type of scaling at each scaling level of α -shearlets by setting $\alpha = (\alpha_j)_j$ with j being the scale and $\alpha_j \in (0, 2)$. The generalized dilation matrices $A_{\alpha_j, 2^j}$ and $\tilde{A}_{\alpha_j, 2^j}$ are then defined by

$$A_{\alpha_j, 2^j} := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{\frac{\alpha_j}{2}j} \end{pmatrix} \quad \text{and} \quad \tilde{A}_{\alpha_j, 2^j} := \begin{pmatrix} 2^{\frac{\alpha_j}{2}j} & 0 \\ 0 & 2^j \end{pmatrix}.$$

Based on these, universal shearlet systems are defined as follows (Genzel and Kutyniok 2014):

Definition 13. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, $\alpha = (\alpha_j)_j$, $\alpha_j \in (0, 2)$, and $c = (c^j)_j$ with $c^j = (c_1^j, c_2^j) \in (\mathbb{R}_+)^2$ for each scale j , the *universal shearlet system* $\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c)$ is defined by

$$\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c) := \Phi(\phi; c_1^0) \cup \Psi(\psi; \alpha, c) \cup \tilde{\Psi}(\tilde{\psi}; \alpha, c),$$

where

$$\begin{aligned} \Phi(\phi; c_1^0) &:= \{\phi_m = \phi(\cdot - c_1^0 m) : m \in \mathbb{Z}^3\}, \\ \Psi(\psi; \alpha, c) &:= \{\psi_{j,k,m} = 2^{\frac{\alpha_j+2}{4}j} \psi(S_k A_{\alpha_j, 2^j} \cdot - M_{c^j} m) : j \geq 0, |k| \leq \lceil 2^{\frac{j(2-\alpha_j)}{2}} \rceil, m \in \mathbb{Z}^2\}, \\ \tilde{\Psi}(\tilde{\psi}; \alpha, c) &:= \{\tilde{\psi}_{j,k,m} = 2^{\frac{\alpha_j+2}{4}j} \tilde{\psi}(S_k^T \tilde{A}_{\alpha_j, 2^j} \cdot - \tilde{M}_{c^j} m) : j \geq 0, |k| \leq \lceil 2^{\frac{j(2-\alpha_j)}{2}} \rceil, m \in \mathbb{Z}^2\}. \end{aligned}$$

In the special situation when all α_j and c^j coincide, i.e., $\alpha_j = \alpha_0$ and $(c_1^j, c_2^j) = (c_1, c_2)$ for all scales j , and $\alpha_0 = 1$, the system reduces to cone-adapted discrete shearlet systems in the sense that $\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c) = \mathcal{SH}(\phi, \psi, \tilde{\psi}; c)$. If in this situation $\alpha_0 = 2$, then the universal shearlet systems reduce to isotropic wavelet systems. Finally, for $\alpha_0 \rightarrow 0$, the system of ridgelets is approached.

Since the implementation of ShearLab3D in www.ShearLab.org relies on universal shearlets, we also state the associated transform explicitly.

Definition 14. Retain the notions from Definition 13, and let $\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c)$ be a universal shearlet system. Then the associated *universal shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is given by

$$SH_{\phi, \psi, \tilde{\psi}} f(j, k, m, \iota) := \begin{cases} \langle f, \psi_{j,k,m} \rangle : \iota = -1, j \geq 0, |k| \leq \lceil 2^{\frac{j(2-\alpha_j)}{2}} \rceil, m \in \mathbb{Z}^2, \\ \langle f, \phi_m \rangle : \iota = 0, m \in \mathbb{Z}^2, \\ \langle f, \tilde{\psi}_{j,k,m} \rangle : \iota = 1, j \geq 0, |k| \leq \lceil 2^{\frac{j(2-\alpha_j)}{2}} \rceil, m \in \mathbb{Z}^2. \end{cases}$$

On the theoretical side, this approach has so far been only analyzed for band-limited generators concerning their frame properties. More precisely, in Genzel and Kutyniok (2014), it has been shown that there exists a large class of scaling sequences $\alpha = (\alpha_j)_j$ such that, using classical shearlets with small modifications, the system $\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c)$ forms a Parseval frame for $L^2(\mathbb{R}^2)$.

Digital Shearlet Systems

One main advantage of shearlets is the fact that they admit a faithful digitalization and hence a consistent implementation, mainly due to the fact that directional sensitivity is incorporated by a shearing operator (instead of, for instance, a rotation operator, which would change the digital grid). The first digital version was introduced in Easley et al. (2008) as the nonsubsampling shearlet transform in 2D and 3D, which digitalized the cone-adapted discrete shearlet transform based on band-limited shearlets. The first faithful digital shearlet transform using compactly supported shearlets was suggested in Lim (2010). It utilizes separable shearlets to achieve low complexity. This approach was later improved in Lim (2013) by an implementation called nonseparable shearlet transform. It uses the fact that nonseparable compactly supported shearlet generators can much better approximate classical band-limited shearlets, which in turn can be designed to form Parseval frames.

Digital 2D Shearlet Transform

In the sequel, we will describe the concept of digital shearlet systems and associated transforms as developed in Lim (2013). In fact, these are also the basis for the software package ShearLab3D provided on the webpage www.ShearLab.org (see also Kutyniok et al. (2016)), which extends this concept to both universal shearlets and the 3D situation.

The digital shearlet systems we will introduce are a faithful digitalization of cone-adapted discrete shearlet systems $\mathcal{SH}(\phi, \psi, \tilde{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c)$ as in Definition 7. Since the component $\Phi(\phi; c_1)$ is just the scaling part coinciding with a wavelet scaling part, we refer for its digitalization to the common wavelet literature (Daubechies 1992; Mallat 1998). Furthermore, we restrict to discussing $\Psi(\psi; c)$, since $\tilde{\Psi}(\tilde{\psi}; c)$ can be digitalized similarly except for switching the order of variables.

We first define a separable shearlet $\psi^{\text{sep}} \in L^2(\mathbb{R}^2)$, which will be the basis for defining a nonseparable variant. For this, let ψ^1 and $\phi^1 \in L^2(\mathbb{R})$ be a compactly supported 1D wavelet and an associated (orthonormal) scaling function, respectively, satisfying the two scale relations

$$\phi^1(x_1) = \sum_{n_1 \in \mathbb{Z}} h(n_1) \sqrt{2} \phi^1(2x_1 - n_1)$$

and

$$\psi^1(x_1) = \sum_{n_1 \in \mathbb{Z}} g(n_1) \sqrt{2} \phi^1(2x_1 - n_1),$$

with some appropriately chosen filters g and h in the sense that both ψ^1 and ϕ^1 are sufficiently smooth and ψ^1 has sufficient vanishing moments. For later use, we also define

$$H_j(\xi_1) := \prod_{k=0}^{j-1} H(2^k \xi_1) \quad \text{and} \quad G_j(\xi_1) := G(2^{j-1} \xi_1) H_{j-1}(\xi_1),$$

where $H(\xi) := \sum_{n \in \mathbb{Z}^d} h_n e^{-2\pi i \langle n, \xi \rangle}$. Then the separable shearlet generator is chosen as $\psi^{\text{sep}} := \psi^1 \otimes \phi^1$.

Based on this, we define the nonseparable generator ψ such as

$$\hat{\psi}(\xi) = P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^{\text{sep}}(\xi), \quad (8)$$

where the trigonometric polynomial P is a 2D fan filter (cf. Do and Vetterli 2005). With a suitable choice for P , we indeed have

$$P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^1(\xi_1) \hat{\phi}^1(\xi_2) \approx \hat{\psi}^1(\xi_1) \hat{\psi}^2\left(\frac{\xi_2}{\xi_1}\right),$$

where $\hat{\psi}^1(\xi_1) \hat{\psi}^2\left(\frac{\xi_2}{\xi_1}\right)$ is a classical shearlet as introduced in Example 1. The functions P , ψ^1 , and ϕ^1 can be chosen in such a way that the sufficient conditions (cf. Theorem 4) for the resulting shearlet system to form a frame are satisfied.

The second step consists of digitalizing the associated shearlet coefficients $\langle f, \psi_{j,k,m} \rangle$ for $j = 0, \dots, J-1$, of a function f given as

$$f(x) = \sum_{m \in \mathbb{Z}^2} f_J(m) 2^J \phi^1(2^J x_1 - m_1) \phi^1(2^J x_2 - m_2), \quad (9)$$

where

$$\psi_{j,k,m}(x) := 2^{\frac{3}{4}j} \psi(S_k A_{2^j} x - M_c m) = \psi_{j,0,m}(S_{k2^{-j/2}} x) \quad (10)$$

with the sampling matrix given by $M_c = \text{diag}(c_1, c_2)$. Without loss of generality, we will from now on assume that $j/2$ is integer; otherwise, we take either $\lceil j/2 \rceil$ or $\lfloor j/2 \rfloor$.

To obtain a faithful discretization of $\psi_{j,0,m}$ in (10) by using the structure of the multiresolution analysis associated with (8), we let $p_j := (p_{j,n})_{n \in \mathbb{Z}^2}$ and g_j be the Fourier coefficients of $P(2^{J-j-1} \xi_1, 2^{J-j/2} \xi_2)$ and G_j , respectively. Then we have

$$\langle f, \psi_{j,0,m} \rangle = (f_J * \overline{(p_j * g_j)})_{A_{2^j}^{-1} 2^J M_c m}, \quad (11)$$

assuming that the sampling matrix M_{c_j} satisfies $A_{2^j}^{-1} 2^J M_c m \in \mathbb{Z}^2$. The associated discrete filter coefficients for $\psi_{j,0,m}$ can be shown to equal $p_j * g_j$. Digitizing (10) also requires a digital shearing operator. Since the shear matrix $S_{k2^{-j/2}}$ does not preserve the regular grid \mathbb{Z}^2 , this problem is resolved by refining the regular grid \mathbb{Z}^2 along the horizontal axis x_1 by a factor of $2^{-j/2}$, leading to the new grid $2^{-j/2} \mathbb{Z} \times \mathbb{Z}$. Now, let $\uparrow 2^{j/2}$, $\downarrow 2^{j/2}$, and $*_1$ be the 1D upsampling, downsampling, and

convolution operator along the horizontal axis x_1 by a factor of $2^{j/2}$, respectively. For a 2D discrete signal $f^d = (f_n^d)_{n \in \mathbb{Z}^2} \in \ell^2(\mathbb{Z}^2)$, the shear operator $S_{k2^{-j/2}}$ can then be digitalized by

$$S_{k2^{-j/2}}^d(f^d) := \left(((\tilde{f}^d)_{S_k(\cdot)} * \overline{h_{j/2}}) \right)_{\downarrow 2^{j/2}}, \tag{12}$$

where \tilde{f}^d given by $\tilde{f}^d := ((f^d)_{\uparrow 2^{j/2}} * h_{j/2})$ is resampled by S_k .

The discussed digitalization of (10) leads to the following definition of a faithful digital shearlet transform:

Definition 15. Let $f_j \in \ell^2(\mathbb{Z}^2)$ be the scaling coefficients given in (9), and retain the notions from this subsection. Then the *digital shearlet transform* associated with $\Psi(\psi; c)$ is defined by

$$DST_{\psi}^{2D} f(j, k, m) := (f_j * \overline{\psi_{j,k}^d})(2^J A_{2^j}^{-1} M_c m) \quad \text{for } j = 0, \dots, J - 1,$$

where

$$\psi_{j,k}^d := S_{k2^{-j/2}}^d(p_j * g_j),$$

with the shearing operator defined by (12) and the sampling matrix M_c chosen so that $2^J A_{2^j}^{-1} M_c m \in \mathbb{Z}^2$.

Considering the full shearlet system and not only $\Psi(\psi; c)$ then leads in a canonical manner to the digital shearlet transform $DST_{\phi, \psi, \tilde{\psi}}^{2D} f(j, k, m, \iota)$ with ι playing a similar role as in Definition 7.

Extensions of the Digital 2D Shearlet Transform and ShearLab3D

We now discuss the extension of the digital shearlet transform to both universal shearlets and the 3D situation, as it is implemented in ShearLab3D (www.ShearLab.org). For details of the implementation, we refer to Kutyniok et al. (2016).

For this, recall the notion from the definition of a universal shearlet system $\mathcal{SH}(\phi, \psi, \tilde{\psi}; \alpha, c)$ (Definition 13). The nonseparable shearlet in $L^2(\mathbb{R}^3)$ is now chosen as

$$\hat{\psi}(\xi) = \left(P\left(\frac{\xi_1}{2}, \xi_2\right) \hat{\psi}^1(\xi_1) \hat{\phi}^1(\xi_2) \right) \left(P\left(\frac{\xi_1}{2}, \xi_3\right) \hat{\phi}^1(\xi_3) \right).$$

Canonically extending the arguments in section “Digital 2D Shearlet Transform” and as before only focusing on $\Psi(\psi; \alpha, c)$, we can digitalize the shearlet coefficients $\langle f, \psi_{j,k,m} \rangle$ for a function $f \in L^2(\mathbb{R}^3)$ given by

$$f(x) = \sum_{m \in \mathbb{Z}^3} f_{J,m} 2^{J \cdot 3/2} (\phi^1 \otimes \phi^1 \otimes \phi^1)(2^J x - m) \quad (13)$$

as follows:

Definition 16. Let $f_J \in \ell^2(\mathbb{Z}^3)$ be the scaling coefficients given in (13), and retain the notions from this section. Then the *digital shearlet transform* associated with $\Psi(\psi; \alpha, c)$ is defined by

$$DST_{\psi}^{3D} f(j, k, m) := (f_J * \overline{\psi_{j,k}^d})(\tilde{m}) \quad \text{for } j = 0, \dots, J-1,$$

where the sampling constants c_1^j and c_2^j are chosen so that

$$\tilde{m} := (2^{J-j} c_1^j m_1, 2^{J-\frac{\alpha_j}{2} j} c_2^j m_2, 2^{J-\frac{\alpha_j}{2} j} c_2^j m_3) \in \mathbb{Z}^3,$$

and the discrete-time Fourier transforms of the 3D digital shearlet filters $\psi_{j,k}^d$ are defined by

$$\Psi_{j,k}^d(\xi) := G_{J-j}(\xi_1) \Phi_{j,k_1}^d(\xi_1, \xi_2) \Phi_{j,k_2}^d(\xi_1, \xi_3)$$

with Φ_{j,k_1}^d and Φ_{j,k_2}^d being the discrete-time Fourier transforms of

$$\phi_{j,k_1,(n_1,n_2)}^d := \left(S_{k_1 2^{-d\alpha_j}}^d (h_{J-\frac{\alpha_j}{2} j} *_{x_2} p_j) \right)_{(n_1,n_2)}$$

and

$$\phi_{j,k_2,(n_1,n_3)}^d := \left(S_{k_2 2^{-d\alpha_j}}^d (h_{J-\frac{\alpha_j}{2} j} *_{x_3} p_j) \right)_{(n_1,n_3)},$$

respectively.

Similar to the 2D situation, the definition of the full 3D digital shearlet transform $DST_{\phi, \psi, \tilde{\psi}, \check{\psi}}^{3D} f(j, k, m, \iota)$ is then canonical.

Applications of Shearlets

This section is devoted to applications of shearlet systems and the associated transforms. We will foremost exploit the fact that shearlets provide optimal sparse approximations of functions which are governed by anisotropic features (section “[Sparse Approximation](#)”), alongside with a faithful implementation (section “[Digital Shearlet Systems](#)”). Due to their high spatial localization and their equal

treatment of different directions, we will focus on compactly supported cone-adapted discrete shearlet systems (section “[Compactly Supported Shearlets](#)”). We wish to remark that the problem settings we present in this section such as image inpainting can also be handled in the 3D setting, i.e., video inpainting, by similar means.

The main areas of application of shearlets are inverse problems from imaging sciences. Indeed, images are typically governed by anisotropic features such as edges, which also the human visual cortex is particularly sensitive to recognize. The traditional (model-based) approach to solving an inverse problem using the fact that the original image is sparsely approximated by a representation system, here shearlets, is by sparse regularization. In the sequel, we will discuss the inpainting (Grohs and Kutyniok 2014; King et al. 2014) and the separation problem (Kutyniok and Lim 2012; Donoho and Kutyniok 2013; Kutyniok 2014) as exemplary problem instances. For a more extensive survey about applications of shearlets using pure model-based approaches, we refer to Easley and Labate (2012) and Kutyniok et al. (2016).

Due to the increasing complexity of problems in imaging, pure model-based methods are often today not sufficient anymore. At the same time, we witness the tremendous success of data-driven methodologies such as deep neural networks for various problem classes, in particular, in imaging sciences. However, entirely replacing physical knowledge about a problem by learned insights is usually not a sensible strategy. The type of approaches which intuitively lead to an optimal combination of the model-based and data-driven realm pursues the strategy to use model-based methods as far as they are reliable and data-driven methods where it is necessary. This concept also circumvents the problem that as of now methodologies such as deep learning act as a black box without any comprehensive theoretical underpinning. In the sequel, the approach “Learning the Invisible” to the limited-angle computed tomography problem (Bubba et al. 2019) shall serve as an example. The classical problem of edge detection, even wavefront set detection, will show another possibility to optimally combine shearlets with deep learning approaches as it is done in DeNSE (Deep Network Shearlet Edge Extractor), leading to superior performance over model-based methods (Andrade-Loarca et al. 2019, 2020).

Sparse Regularization Using Shearlets

Given an ill-posed inverse problem $Tf = g$, where $T : \mathcal{H} \rightarrow \mathcal{H}$ is a linear, bounded operator and $g \in \mathcal{H}$, classical Tikhonov regularization aims to solve this problem by minimizing the functional

$$\|Tf - g\|^2 + \beta \cdot \|f\|^2,$$

with β being the regularization parameter. However, the regularization term $\|f\|^2$ might not be appropriate for each inverse problem, and other prior information of f is known and should be incorporated. For functions in $L^2(\mathbb{R}^2)$ governed by

anisotropic features such as images, shearlet systems provide optimal sparse approximations. The generalization of Tikhonov regularization introduced in Daubechies et al. (2004) exploits such information by suggesting to minimize

$$\|Tf - g\|^2 + \beta \cdot \|(\langle \cdot, \psi_\lambda \rangle)_{\lambda \in \Lambda}\|_{\ell_1},$$

with $(\psi_\lambda)_{\lambda \in \Lambda}$ being a shearlet system, instead. We remark that the concept of sparse regularization is closely related to, and in fact might also be seen as belonging to, the area of compressed sensing (Davenport et al. 2012).

We now discuss two different special situations in which this conceptual approach can be applied.

Image Separation

Images are typically a composition of morphologically distinct components. The problem of image separation, which is a highly ill-posed inverse problem, aims to decompose the image into those components. To be mathematically precise, assuming just two components, the problem can be modeled as follows: Let $f_1, f_2 \in L^2(\mathbb{R}^2)$ and $g = f_1 + f_2$; we aim to recover f_1 and f_2 from g . One possible setting is the separation of curve-like and point-like objects, which, for example, appears in neurobiological imaging in the form of spines (point-like objects) and dendrites (curve-like objects) or astronomical imaging in the form of stars (point-like objects) and filaments (curve-like objects). For further examples, we refer to Starck et al. (2010).

This problem can only be solved by assuming prior information on the components. The approach of sparse regularization assumes that each component f_1 and f_2 can be sparsified by a representation system $(\psi_\lambda^1)_{\lambda \in \Lambda}$ and $(\psi_\lambda^2)_{\lambda \in \Lambda}$, respectively. This leads to the following minimization problem:

$$\arg \min_{u_1, u_2} \|(\langle u_1, \psi_\lambda^1 \rangle)_{\lambda \in \Lambda}\|_{\ell_1} + \|(\langle u_2, \psi_\lambda^2 \rangle)_{\lambda \in \Lambda}\|_{\ell_1} \quad \text{subject to} \quad g = u_1 + u_2, \tag{14}$$

where we chose the constrained form of the optimization problem, for which also the theoretical results are formulated. Let us now consider the situation that f_1 are point-like features and f_2 are curve-like features. In this case, we would choose $(\psi_\lambda^1)_{\lambda \in \Lambda}$ to be a wavelet system and $(\psi_\lambda^2)_{\lambda \in \Lambda}$ to be a shearlet system. For an illustration, we refer to Fig. 6.

To explain the associated theoretical results, assume for f_1 and f_2 models for point-like and curve-like features, namely,

$$f_1 := \sum_{i=1}^P |x - x_i|^{-3/2} \quad \text{and} \quad f_2 := \int \delta_{\tau(t)} dt,$$

where $x_i \in \mathbb{R}^2$ and $\tau : [0, 1] \rightarrow \mathbb{R}^2$ are closed curves. To aim for an asymptotic analysis, let $(F_j)_j$ be a sequence of filters such as wavelet filters satisfying

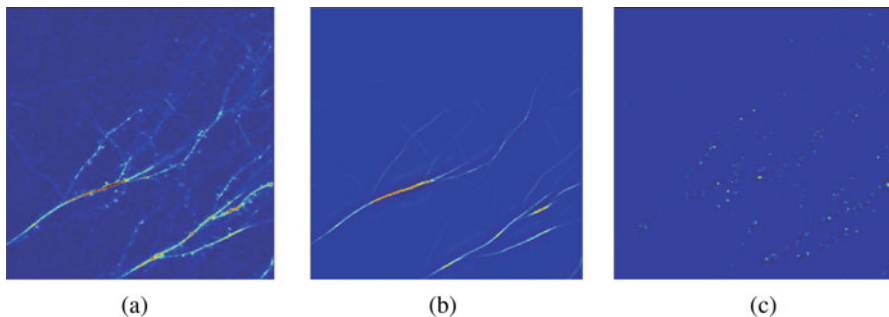


Fig. 6 Separation of spines and dendrites in neurobiological imaging (Kutyniok and Lim 2012) using ShearLab3D to solve (14). (a) Original image. (b) Extracted dendrites (curve-like objects). (c) Extracted spines (point-like objects)

$$g = \sum_j F_j * (F_j * g), \quad \text{for all } g \in L^2(\mathbb{R}^2). \tag{15}$$

This leads to a scale-dependent decomposition. Consider then the accordingly filtered components $(f_{i,j})_j := (f_i * F_j)_j$, $i = 1, 2$ as well as the image at scale j , i.e., $g_j := f_{1,j} + f_{2,j}$. The following result analyzes the microlocal structure of the problem and shows that at all sufficiently fine scales, nearly perfect separation is achieved. The key reason for the success of the separation approach is the morphological difference between the point and curve structures, which is mirrored in the difference between the associated sparsifying systems.

Theorem 8 (Donoho and Kutyniok 2009, 2013). *Retaining the notation from this subsection and letting $\widetilde{f}_{1,j}$, $\widetilde{f}_{2,j}$ denote the solution of (14) for the separation problem $g_j = f_{1,j} + f_{2,j}$, we have*

$$\frac{\|f_{1,j} - \widetilde{f}_{1,j}\|_{L^2} + \|f_{2,j} - \widetilde{f}_{2,j}\|_{L^2}}{\|f_{1,j}\|_{L^2} + \|f_{2,j}\|_{L^2}} \rightarrow 0, \quad j \rightarrow \infty.$$

A stronger result concerning recovery of the wavefront sets of the models for point-like and curve-like features using a thresholding algorithm was derived in Kutyniok (2013, 2014) studies the separation of cartoon and texture using as sparsifying systems a shearlet and a Gabor system. Finally, for similar results in the general Hilbert space setting, we refer to Donoho and Kutyniok (2013).

Image inpainting

Image inpainting aims to recover missing or deteriorated parts of an image. It is thus a special case of a data recovery problem; and the approach we discuss can be generalized to this setting as well. The problem can be formulated as follows:

Let $f \in L^2(\mathbb{R}^2)$ and a (measurable) mask $M \subset \mathbb{R}^2$; we aim to recover f from $g := f \cdot 1_{\mathbb{R}^2 \setminus M}$.

Let now $(\psi_\lambda)_{\lambda \in \Lambda}$ be a shearlet system. Sparse regularization using shearlets assumes the following model for the solution, where – similar as in the previous subsection – we choose the constrained form of the optimization problem:

$$\min_u \|(\langle u, \psi_\lambda \rangle)_{\lambda \in \Lambda}\|_{\ell_1} \quad \text{subject to} \quad g = u \cdot 1_{\mathbb{R}^2 \setminus M}. \tag{16}$$

Figure 7 shows some numerical experiments. For further examples as well as comparison to other state-of-the-art approaches, we refer to Kutyniok et al. (2016).

Theoretical results have been achieved in the case that f is a distribution with a curvilinear singularity, i.e.:

$$f := \int_{-\rho}^{\rho} w(t) \delta_{\tau(t)} dt,$$

where $\tau : [-1, 1] \rightarrow \mathbb{R}^2$ is a C^2 -curve, $\rho < 1$, and $w : [-\rho, \rho] \rightarrow \mathbb{R}_0^+$ is a “bump” function. The mask is then defined as a vertical strip intersecting the curve, with a flexible width:

$$M_h = \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| \leq h\}, \quad h > 0.$$

Again aiming for an asymptotic analysis, let $(F_j)_j$ be a sequence of filters (cf. (15)), which leads to a scale-dependent decomposition, and consider the filtered image $f_j := f * F_j$ as well as the filtered observed image $g_j := (f \cdot 1_{\mathbb{R}^2 \setminus M_{h_j}}) * F_j$, where we also make the width of the mask dependent on the scale j . The following result shows that at all sufficiently fine scales, nearly perfect inpainting is achieved in case the shearlets are asymptotically larger than the width of the mask.

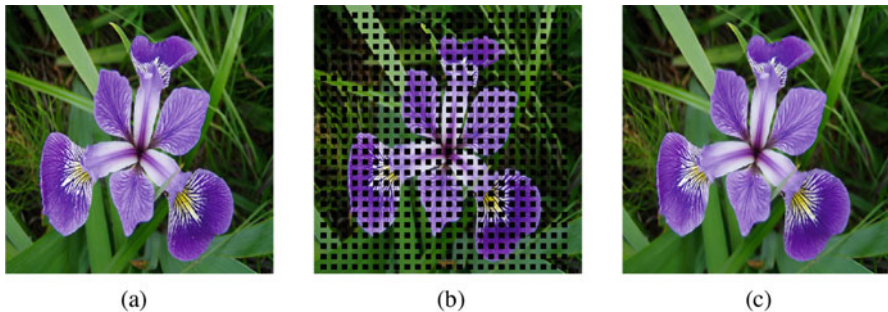


Fig. 7 Numerical experiments using ShearLab3D to solve (16). (a) Original image. (b) Masked image. (c) Inpainted image

Theorem 9 (King et al. 2014). *Retaining the notation from this subsection and letting \tilde{f}_j denote the solution of (16) for the inpainting problem $g_j = (f \cdot 1_{\mathbb{R}^2 \setminus M_{h_j}}) * F_j$, if $h_j = o(2^{-j/2})$ as $j \rightarrow \infty$, we have*

$$\frac{\|\tilde{f}_j - f_j\|_{L^2}}{\|f_j\|_{L^2}} \rightarrow 0, \quad j \rightarrow \infty.$$

A similar result holds for wavelet inpainting, then with the sufficient condition that $h_j = o(2^{-j})$ as $j \rightarrow \infty$ according to the smaller width of a wavelet element. An extension to inpainting using universal shearlet systems can be found in Genzel and Kutyniok (2014). For similar results in the general Hilbert space setting, we refer to Donoho and Kutyniok (2013) and Genzel and Kutyniok (2014).

Shearlets Meet Deep Learning

Deep learning approaches have recently swept the area of inverse problems, predominantly from imaging, the main reason being that no physical model for images exists, consequently making data-driven methods very effective. A standard feed-forward *deep neural network* consists of affine-linear maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \rightarrow \mathbb{R}^{N_\ell}$, $\ell = 1, \dots, L$, i.e., $W_\ell(x) = A_\ell x + b_\ell$, where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$, as well as a (nonlinear) univariate function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ called *activation function*, and realizes the map $\mathcal{NN}_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{N_L}$

$$\mathcal{NN}_\theta(x) = W_L \sigma(W_{L-1} \sigma(\dots \sigma(W_1(x))))$$

with σ being applied componentwise and θ denoting all parameters of the neural network, i.e., the weight matrices A_ℓ and biases b_ℓ . In applications, the activation function is typically chosen as the ReLU (Rectified Linear Unit) given by $\sigma(x) := \max\{0, x\}$. Corresponding to the depiction as a graph, L is referred to as the number of layers. Given samples $(x_i, f(x_i))_{i=1}^m$ of a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^{N_L}$, learning algorithms such as stochastic gradient descent learn θ according to minimizing a certain empirical risk functional.

For an introduction and overview, also concerning the various types of neural networks, we refer to Goodfellow et al. (2017).

Convolutional neural networks, in which convolutions are performed in each layer, are the state-of-the-art for imaging applications. The network architecture typically utilized for solving inverse problems is the *U-Net* as introduced in Ronneberger et al. (2015), which can be regarded as an autoencoder with additional skip connections to allow the transportation of additional information across the compressed layers.

The most basic approach to solving an inverse problem $Tf = g$ by deep learning is to train a neural network Φ on samples $(Tf_i, f_i)_{i=1}^m$, thereby pursuing a pure data-driven method while entirely discarding physical knowledge. Another elementary

approach, which was suggested in Jin et al. (2017), first recovers an approximation of f from g by standard model-based approaches followed by a convolutional neural network, which acts as a denoiser. More sophisticated types of approaches aim to insert deep neural networks in iterative reconstruction schemes, for instance, by replacing certain steps such as a denoising step by a neural network, which was pioneered in Gregor and LeCun (2010), or replacing some of the proximal operators by networks (see, e.g., Meinhardt et al. (2017) and Adler and Öktem (2018)). For an overview of deep learning approaches to inverse problems, we refer to Adler and Öktem (2017) and McCann et al. (2017).

In contrast to the previously discussed approaches, we will now present two exemplary algorithms which combine the model-based realm represented by shearlets with the data-driven realm of deep neural networks following the philosophy of using model-based methods as far as they are reliable and data-driven methods where it is necessary. This conceptual type of approach not only avoids that deep neural networks affect the entire data set during inversion, which presumably causes instabilities (Gottschling et al. 2020), but also allows a better interpretation of the results.

Limited-Angle Computed Tomography

Computed tomography (CT) is one of the main imaging technologies for medical diagnosis. A CT scanner samples the *Radon transform*

$$\mathcal{R}f(\phi, s) = \int_{L(\phi, s)} f(x) dS(x),$$

where $L(\phi, s) = \{x \in \mathbb{R}^2 : x_1 \cos(\phi) + x_2 \sin(\phi) = s\}$, $\phi \in [-\pi/2, \pi/2)$, and $s \in \mathbb{R}$ (Natterer 2001). The inverse problem of reconstructing f from its Radon transform $g := \mathcal{R}f$ becomes even more challenging when only partial data is available. One instance of this problem complex is *limited-angle computed tomography*, where $\mathcal{R}f(\cdot, s)$ is only sampled on $[-\phi, \phi] \subset [-\pi/2, \pi/2)$. Examples include breast tomosynthesis, dental CT, and electron tomography. Due to the large missing part in the measured data – in contrast to, for instance, low-dose CT – model-based approaches only provide crude reconstructions, since no model-based priors exist which model a human body sufficiently accurately.

Depending on the missing angle, it is known which information about the wavefront set of the original image is contained in the measured data, hence in this sense what is “visible” (Quinto 1993). This allows to view the problem of limited-angle computed tomography as an inpainting problem of the wavefront set. Due to the sensitivity of shearlets to the wavefront set (Theorem 2), it is suggestive to exploit this system in this problem setting.

The approach “Learning the Invisible” (Bubba et al. 2019) pursues this strategy, by first reconstructing the image using sparse regularization with shearlets as sparsifying system, followed by surgically precisely learning the invisible data

corresponding to the missing part of the wavefront set by a deep learning approach. The algorithm can be outlined as follows:

- *Step 1: Reconstruct the Visible.* Solve

$$f^* := \arg \min_{f \geq 0} \|\mathcal{R}f - g\|_2^2 + \|SH_{\phi, \psi, \tilde{\psi}} f\|_{1, w},$$

with $SH_{\phi, \psi, \tilde{\psi}}$ being a shearlet transform and $\|\cdot\|_{1, w}$ a suitably chosen weighted ℓ_1 norm. The wavefront set can then be approximately assessed via a sparsity prior on shearlets in the following sense, where \mathcal{I}_{inv} corresponds to the “invisible” shearlet coefficients and \mathcal{I}_{vis} to the “visible” coefficients:

- For $(j, k, m, \iota) \in \mathcal{I}_{\text{inv}}$: $SH_{\phi, \psi, \tilde{\psi}} f^*(j, k, m, \iota) \approx 0$.
- For $(j, k, m, \iota) \in \mathcal{I}_{\text{vis}}$: $SH_{\phi, \psi, \tilde{\psi}} f^*(j, k, m, \iota)$ is reliable and near perfect.
- *Step 2: Learn the Invisible.* Apply a neural network \mathcal{NN}_θ with a U-Net-like CNN architecture of 40 layers coined *PhantomNet* (Bubba et al. 2019), which is trained using training data $(f_i^*, f_i^{\text{gt}})_{i=1}^m$ (“gt” = “groundtruth”) by minimizing

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^m \|\mathcal{NN}_\theta(SH_{\phi, \psi, \tilde{\psi}} f_i^*) - SH_{\phi, \psi, \tilde{\psi}} f_i^{\text{gt}}|_{\mathcal{I}_{\text{inv}}}\|_{w, 2}^2,$$

and compute

$$\mathcal{NN}_\theta : SH_{\phi, \psi, \tilde{\psi}} f^*|_{\mathcal{I}_{\text{vis}}} \longrightarrow F \left(\overset{!}{\approx} SH_{\phi, \psi, \tilde{\psi}} f^{\text{gt}}|_{\mathcal{I}_{\text{inv}}} \right).$$

- *Step 3: Combine.* Compute the reconstruction

$$f_{\text{L}t\text{I}} = SH_{\phi, \psi, \tilde{\psi}}^{-1} \left(SH_{\phi, \psi, \tilde{\psi}} f^*|_{\mathcal{I}_{\text{vis}}} + F \right).$$

Figure 8 shows numerical results, which prove superiority not only over the model-based approach but even over the pure deep learning approach from Gu and Ye (2017).

Wavefront Set Detection

Edge detection is a widely studied problem, which aims to detect singularity points in an image. As argued before, edges carry most of the information of an image; in addition, it is believed that rough sketching involving edge detection is actually the first of the operations of the human visual cortex. Various approaches to edge detection have been suggested with maybe the most famous one being the Canny edge detector (Canny 1986).

However, sometimes not only the detection of the edge but also its directionality in the sense of detecting the wavefront set is required. One example is – also related to the previous subsection – tomographic imaging. In fact, the wavefront set of an

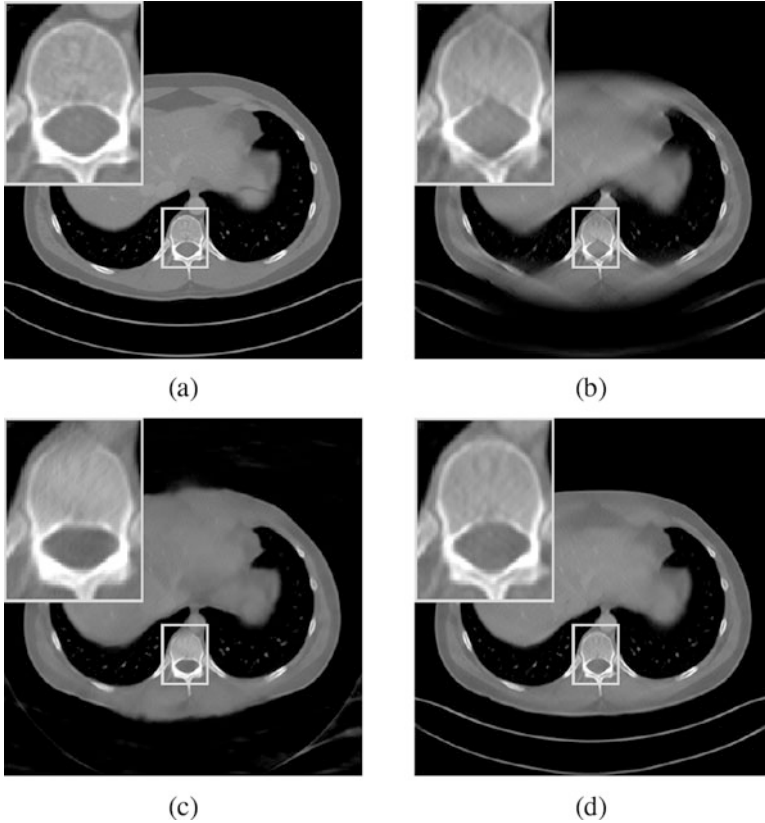


Fig. 8 Numerical experiments from Bubba et al. (2019) using data from Mayo-60° with a missing wedge of 60°, where RE stands for relative error and HaarPSI is the Haar wavelet-based perceptual similarity index for image quality assessment (Reisenhofer et al. 2018). (a) Original image. (b) f^* (RE: 0.19, HaarPSI: 0.43). (c) Result from Gu and Ye (2017) (RE: 0.22, HaarPSI: 0.40). (d) f_{LTI} (RE: 0.09, HaarPSI: 0.76)

image can be related to the wavefront set of its transformed version such as its Radon transform by (microlocal) canonical relations. Being able to detect the wavefront set of the Radon transform, say, allows to compute an approximation of the wavefront set of the original image by a (microlocal) canonical relation and use it as a prior for reconstruction (Andrade-Loarca et al. 2020).

Cone-adapted continuous shearlet systems are able to resolve wavefront sets (Theorem 2). But algorithms following this model such as Yi et al. (2009) and Reisenhofer et al. (2015) often suffer from the fact that real-world scenarios are highly complex and the theoretical analysis only provides an asymptotic estimate.

In the sequel, we will discuss an approach coined DeNSE (Deep Network Shearlet Edge Extractor) (Andrade-Loarca et al. 2019), which again follows the philosophy to use a model-based approach as far as it is reliable and use a deep

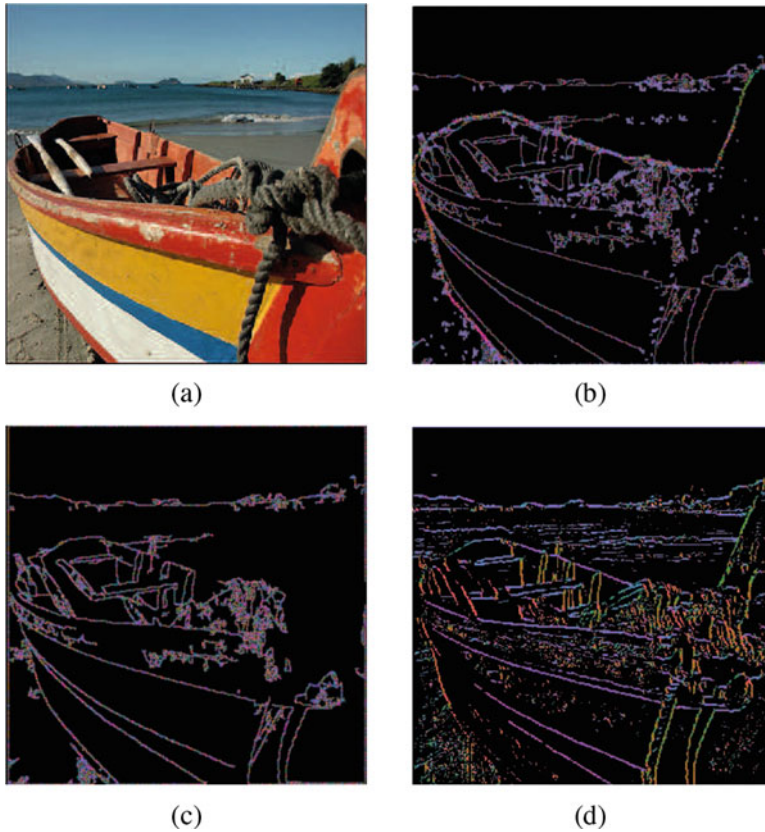


Fig. 9 Numerical experiments from Andrade-Loarca et al. (2019), where the color coding indicates the detected direction. (a) Original image. (b) Result from Yi et al. (2009). (c) Result from Reisenhofer et al. (2015). (d) Result using DeNSE. Copyright ©2019 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved

neural network where it is necessary. More precisely, it first computes a shearlet transform whose ability to detect wavefront sets is subsequently improved by deep learning when operating in shearlet domain. The algorithm can be outlined as follows:

- *Step 1: Reveal Directionality in the Shearlet Domain.* For a given test image $f \in \mathbb{R}^{M \times M}$, compute the digital shearlet transform of f with 49 shearlet generators, i.e.,

$$\left(DST_{\phi, \psi, \tilde{\psi}}^{2D} f(j, k, m, \iota) \right)_{j, k, m \in [1, M]^2, \iota \in \{-1, 0, 1\}}.$$

- *Step 2: Shearlet Transform.* For every location $m^* = (m_1^*, m_2^*) \in [11, M - 10]^2$, apply a neural network classifier consisting of four convolutional layers plus one fully connected layer to the associated patch

$$\left(DST_{\phi, \psi, \tilde{\psi}}^{2D} f(j, k, m, \iota) \right)_{j, k, m \in [m_1^* - 10, m_1^* + 10] \times [m_2^* - 10, m_2^* + 10], \iota \in \{-1, 0, 1\}}$$

If the network predicts the presence of an edge with direction ϑ , then (m^*, ϑ) is detected as an element of the wavefront set of f .

For an example of the effectiveness of this hybrid approach, we refer to Fig. 9.

Conclusion

The area of applied harmonic analysis provides representation systems for data processing, aiming for both decomposition and expansion of data/functions. Shearlet systems are specifically designed for the setting of multivariate functions and exist as continuous, discrete, and digital systems. While the continuous version allows a precise resolution of wavelet fronts, the discrete version provides optimally sparse approximations of cartoon-like functions as a model class of functions being governed by anisotropic features, and the digital version yields faithful implementations. Shearlet systems can be extended to higher dimensions as well as also to more general universal shearlets and a-molecules. Shearlet systems are typically used for sparse regularization of inverse problems such as feature extraction and inpainting, for which both theoretical and numerical results are available. Recent applications combine the shearlet transform with deep neural networks in a smart way targeting problems such as limited-angle computed tomography and wavefront set detection.

Acknowledgments G.K. would like to thank Hector Andrade-Loarca for producing several of the figures.

References

- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**, 124007 (2017)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE T. Med. Imaging* **37**, 1322–1332 (2018)
- Andrade-Loarca, H., Kutyniok, G., Öktem, O.: Shearlets as feature extractor for semantic edge detection: the model-based and data-driven realm. *Proc. R. Soc. A.* **476**(2243), 20190841 (2020). <https://royalsocietypublishing.org/toc/rspa/2020/476/2243>
- Andrade-Loarca, H., Kutyniok, G., Öktem, O., Petersen, P.: Extraction of digital wavefront sets using applied harmonic analysis and deep neural networks. *SIAM J. Imaging Sci.* **12**, 1936–1966 (2019)

- Antoine, J.P., Carrette, P., Murenzi, R., Piette, B.: Image analysis with two-dimensional continuous wavelet transform. *Sig. Process.* **31**, 241–272 (1993)
- Bamberger, R.H., Smith, M.J.T.: A filter bank for the directional decomposition of images: theory and design. *IEEE Trans. Sig. Process.* **40**, 882–893 (1992)
- Bodmann, B.G., Labate, D., Pahari, B.R.: Smooth projections and the construction of smooth Parseval frames of shearlets. *Adv. Comput. Math.* **45**, 3241–3264 (2019)
- Bubba, T.A., Kutyniok, G., Lassas, M., März, M., Samek, W., Siltanen, S., Srinivasan, V.: Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.* **35**, 064002 (2019). <https://iopscience.iop.org/article/10.1088/1361-6420/ab10ca>
- Candès, E.J., Donoho, D.L.: New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. Pure Appl. Math.* **56**, 216–266 (2004)
- Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 679–698 (1986)
- Casazza, P.G., Kutyniok, G., Philipp, F.: Introduction to finite frame theory. In: *Finite Frames: Theory and Applications*, pp. 1–53. Birkhäuser, Boston (2012)
- Christensen, O.: *An Introduction to Frames and Riesz Bases*. Birkhäuser, Boston (2003)
- Cohen, A.: *Numerical Analysis of Wavelet Methods*. Studies in Mathematics and Its Applications, vol. 32. JAI Press, Greenwich (2003)
- Dahlke, S., Kutyniok, G., Maass, P., Sagiv, C., Stark, H.-G., Teschke, G.: The uncertainty principle associated with the continuous shearlet transform. *Int. J. Wavelets Multiresolut. Inf. Process.* **6**, 157–181 (2008)
- Dahlke, S., Kutyniok, G., Steidl, G., Teschke, G.: Shearlet coorbit spaces and associated banach frames. *Appl. Comput. Harmon. Anal.* **27**, 195–214 (2009)
- Dahlke, S., Steidl, G., Teschke, G.: The continuous shearlet transform in arbitrary space dimensions. *J. Fourier Anal. Appl.* **16**, 340–354 (2010)
- Dahlke, S., Steidl, G., Teschke, G.: Shearlet coorbit spaces: compactly supported analyzing shearlets, traces and embeddings. *J. Fourier Anal. Appl.* **17**, 1232–1255 (2011)
- Dahlke, S., Häuser, S., Steidl, G., Teschke, G.: Shearlet coorbit spaces: traces and embeddings in higher dimensions. *Monatsh. Math.* **169**, 15–32 (2013)
- Daubechies, I.: *Ten Lectures on Wavelets*. SIAM, Philadelphia (1992)
- Daubechies, I., DeFrise, M., De Mo, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **57**, 1413–1457 (2004)
- Davenport, M., Duarte, M., Eldar, Y., Kutyniok, G.: Introduction to compressed sensing. In: *Compressed Sensing: Theory and Applications*, pp. 1–64. Cambridge University Press (2012)
- Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.* **14**, 2091–2106 (2005)
- Donoho, D.L.: Sparse components of images and optimal atomic decomposition. *Constr. Approx.* **17**, 353–382 (2001)
- Donoho, D.L., Kutyniok, G.: Geometric separation using a wavelet-shearlet dictionary. *SampTA'09, Marseille. Proceedings* (2009)
- Donoho, D.L., Kutyniok, G.: Microlocal analysis of the geometric separation problem. *Commun. Pure Appl. Math.* **66**, 1–47 (2013)
- Easley, G., Labate, D.: Image processing using shearlets. In: *Shearlets: Multiscale Analysis for Multivariate Data*, pp. 283–325. Birkhäuser, Boston (2012)
- Easley, G., Labate, D., Lim, W.-Q.: Sparse directional image representation using the discrete shearlet transform. *Appl. Comput. Harmon. Anal.* **25**, 25–46 (2008)
- Genzel, M., Kutyniok, G.: Asymptotic analysis of inpainting via universal shearlet systems. *SIAM J. Imaging Sci.* **7**, 2301–2339 (2014)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge (2017)
- Gottschling, N., Antun, V., Adcock, B., Hansen, A.C.: The troublesome kernel: why deep learning for inverse problems is typically unstable. preprint, arXiv:2001.01258 (2020)

- Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: International Conference on Machine Learning (ICML), pp. 399–406 (2010)
- Grohs, P.: Continuous Shearlet frames and Resolution of the Wavefront Set. *Monatsh. Math.* **164**, 393–426 (2011a)
- Grohs, P.: Continuous shearlet tight frames. *J. Fourier Anal. Appl.* **17**, 506–518 (2011b)
- Grohs, P.: Bandlimited shearlet frames with nice duals. *J. Comput. Appl. Math.* **142**, 139–151 (2013)
- Grohs, P., Kutyniok, G.: Parabolic molecules. *Found. Comput. Math.* **14**, 299–337 (2014)
- Grohs, P., Keiper, S., Kutyniok, G., Schäfer, M.: α -molecules. *Appl. Comput. Harmon. Anal.* **42**, 297–336 (2016a)
- Grohs, P., Keiper, S., Kutyniok, G., Schäfer, M.: Cartoon approximation with α -curvelets. *J. Fourier Anal. Appl.* **22**, 1235–1293 (2016b)
- Gu, J., Ye, J.C.: Multi-scale wavelet domain residual learning for limited-angle CT reconstruction. In: *Procs Fully3D*, pp. 443–447 (2017)
- Guo, K., Labate, D.: Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.* **39**, 298–318 (2007)
- Guo, K., Labate, D.: The construction of smooth parseval frames of shearlets. *Math. Model. Nat. Phenom.* **8**, 82–105 (2013)
- Guo, K., Kutyniok, G., Labate, D.: Sparse multidimensional representations using anisotropic dilation and shear operators. In: *Wavelets and Splines*, Athens, 2005, pp. 189–201. Nashboro Press, Nashville (2006)
- Guo, K., Labate, D., Lim, W.-Q.: Edge analysis and identification using the continuous shearlet transform. *Appl. Comput. Harmon. Anal.* **27**, 24–46 (2009)
- Häuser, S., Steidl, G.: Convex multiclass segmentation with shearlet regularization. *Int. J. Comput. Math.* **90**, 62–81 (2013)
- Hörmander, L.: The analysis of linear partial differential operators. I. Distribution theory and Fourier analysis. Springer, Berlin (2003)
- Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Proc.* **26**, 4509–4522 (2017)
- King, E.J., Kutyniok, G., Zhuang, X.: Analysis of inpainting via clustered sparsity and microlocal analysis. *J. Math. Imaging Vis.* **48**, 205–234 (2014)
- Kittipoom, P., Kutyniok, G., Lim, W.-Q.: Irregular shearlet frames: geometry and approximation properties. *J. Fourier Anal. Appl.* **17**, 604–639 (2011)
- Kittipoom, P., Kutyniok, G., Lim, W.-Q.: Construction of compactly supported shearlet frames. *Constr. Approx.* **35**, 21–72 (2012)
- Kutyniok, G.: Clustered sparsity and separation of cartoon and texture. *SIAM J. Imaging Sci.* **6**, 848–874 (2013)
- Kutyniok, G.: Geometric separation by single-pass alternating thresholding. *Appl. Comput. Harmon. Anal.* **36**, 23–50 (2014)
- Kutyniok, G., Labate, D.: Resolution of the wavefront set using continuous shearlets. *Trans. Am. Math. Soc.* **361**, 2719–2754 (2009)
- Kutyniok, G., Labate, D.: Introduction to shearlets. In: *Shearlets: Multiscale Analysis for Multivariate Data*, pp. 1–3. Birkhäuser, Boston (2012)
- Kutyniok, G., Lim, W.-Q.: Compactly supported shearlets are optimally sparse. *J. Approx. Theory* **163**, 1564–1589 (2011)
- Kutyniok, G., Lim, W.-Q.: Image separation using wavelets and shearlets. In: *Curves and Surfaces*, Avignon, 2010. *Lecture Notes in Computer Science*, vol. 6920, pp. 416–430. Springer (2012)
- Kutyniok, G., Lim, W.-Q.: Optimal compressive imaging of Fourier data. *SIAM J. Imaging Sci.* **11**, 507–546 (2018)
- Kutyniok, G., Petersen, P.: Classification of edges using compactly supported shearlets. *Appl. Comput. Harmon. Anal.* **42**, 245–293 (2017)
- Kutyniok, G., Lemvig, J., Lim, W.-Q.: Optimally sparse approximations of 3D functions by compactly supported shearlet frames. *SIAM J. Math. Anal.* **44**, 2962–3017 (2012)

- Kutyniok, G., Lim, W.-Q., Reisenhofer, R.: ShearLab 3D: faithful digital shearlet transforms based on compactly supported shearlets. *ACM Trans. Math. Softw.* **42**, 5 (2016)
- Labate, D., Lim, W.-Q., Kutyniok, G., Weiss, G.: Sparse multidimensional representation using shearlets. In: *Wavelets XI, Proceedings of SPIE, Bellingham*, vol. 5914, pp. 254–262 (2005)
- Labate, D., Mantovani, L., Negi, P.S.: Shearlet smoothness spaces. *J. Fourier Anal. Appl.* **19**, 577–611 (2013)
- Le Pennec, E.L., Mallat, S.: Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.* **14**, 423–438 (2005)
- Lessig, C., Petersen, P., Schäfer, M.: Bendlets: a second-order shearlet transform with bent elements. *Appl. Comput. Harmon. Anal.* **46**, 384–399 (2019)
- Lim, W.-Q.: The discrete shearlet transform: a new directional transform and compactly supported shearlet frames. *IEEE Trans. Image Proc.* **19**, 1166–1180 (2010)
- Lim, W.-Q.: Nonseparable shearlet transform. *IEEE Trans. Image Proc.* **22**, 2056–2065 (2013)
- Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic Press, San Diego (1998)
- McCann, M.T., Jin, K.H., Unser, M.: Convolutional neural networks for inverse problems in imaging: a review. *IEEE Signal Proc. Mag.* **34**, 85–95 (2017)
- Meinhardt, T., Möller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: using denoising networks for regularizing inverse imaging problems. In: *International Conference on Computer Vision (ICCV)* (2017)
- Natterer, F.: *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2001)
- Quinto, E.T.: Singularities of the X-ray transform and limited data tomography in \mathbb{R}^2 and \mathbb{R}^3 . *SIAM J. Math. Anal.* **24**, 1215–1225 (1993)
- Reisenhofer, R., Kiefer, J., King, E.J.: Shearlet-based detection of flame fronts. *Exp. Fluids* **57**, 11 (2015)
- Reisenhofer, R., Bosse, S., Kutyniok, G., Wiegand, T.: A Haar wavelet-based perceptual similarity index for image quality assessment. *Sig. Process. Image* **61**, 33–43 (2018)
- Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. LNCS, vol. 9351, pp. 234–241. Springer (2015)
- Simoncelli, E.P., Freeman, W.T., Adelson, E.H., Heeger, D.J.: Shiftable multiscale transforms. *IEEE Trans. Inform. Theory* **38**, 587–607 (1992)
- Starck, J.-L., Murtagh, F., Fadili, J.: *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, Cambridge (2010)
- Yi, S., Labate, D., Easley, G.R., Krim, H.: A shearlet approach to edge analysis and detection. *IEEE Trans. Image Process.* **18**, 929–941 (2009)



Sebastian Lunz

Contents

Introduction	1134
Shallow Learned Regularizers	1136
Bilevel Learning	1136
Dictionary Learning	1137
Deep Regularizers	1138
Regularization Properties of Learned Regularizers	1138
Adversarial Regularization	1141
Total Deep Variation	1145
Summary and Outlook	1149
Conclusion	1149
Outlook	1151
References	1152

Abstract

In the past years, there has been a surge of interest in methods to solve inverse problems that are based on neural networks and deep learning. A variety of approaches have been proposed, showing improvements in reconstruction quality over existing methods. Among those, a class of algorithms builds on the well-established variational framework, training a neural network as a regularization functional. Those approaches come with the advantage of a theoretical understanding and a stability theory that is built on existing results for variational regularization. We discuss various approaches for learning a

S. Lunz (✉)

Department of Applied Mathematics and Theoretical Physics, University of Cambridge,
Cambridge, UK

e-mail: sl767@cam.ac.uk

regularization functional, aiming at giving an overview at the multiple directions investigated by the research community.

Keywords

Inverse problems · Variational regularization · Deep learning

Introduction

We consider an inverse problem of the form

$$y = Ax + \epsilon, \quad (1)$$

where $x \in \mathcal{X}$ is an image we wish to reconstruct from measurements $y \in \mathcal{Y}$, the operator $A : \mathcal{X} \rightarrow \mathcal{Y}$ is linear, and $\epsilon \in \mathcal{Y}$ is random noise. A well-established framework for recovering x is via solving a variational problem of the form

$$\arg \min_x \mathcal{D}(Ax, y) + \lambda \mathcal{R}(x), \quad (2)$$

where $\mathcal{D} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a distance functional, typically chosen to be the ℓ^2 distance if the noise ϵ is Gaussian. The regularization functional \mathcal{R} is chosen such that minimization is well-posed despite the pseudo-inverse A^\dagger possibly being unbounded. A classical choice of \mathcal{R} is the Tikhonov regularization functional $\mathcal{R}(x) := \|x\|_2^2$. This allows deducing various stability and convergence results on the reconstruction (see, e.g., Engl et al. 1996).

Taking the viewpoint of Bayesian statistics, we can interpret a solution to (2) as a maximum a posteriori likelihood estimator via

$$\arg \max_x \log p(x|y) = \arg \min_x -\log p(x|y) - \log p(x). \quad (3)$$

The expression $\log p(x|y)$ is captured by the data term $\mathcal{D}(Ax, y)$, whereas the regularization functional can be viewed as an approximation to the log prior. This viewpoint motivates investigating priors beyond their ability to stabilize reconstruction, explaining the success of widely used handcrafted priors such as total variation (TV) that capture the distinct properties of the distribution of images, such as sharp edges, more closely than Tikhonov-type regularization.

While TV has enjoyed great success in the past decades, its representation of the behavior of images remains limited, assuming them to be piecewise constant. As this is not true for many images, TV-based regularization is known to introduce staircasing artefacts into reconstructions. To overcome these drawbacks, the research community has shifted their focus on learning priors from data directly, with the goal of obtaining a more realistic and detailed image representation. More precisely, one aims at utilizing a training set $\{(\tilde{x}^i, y^i)\}$ of ground truth images \tilde{x}^i

and associated measurements y^i to learn powerful characterization of images from data directly. We want to note at this point that the setting $\{\{\tilde{x}^i, y^i\}\}$ corresponds to a supervised training setting and that some algorithms require less structure in the training data, as, for example, dictionary learning (section “[Dictionary Learning](#)”) or the adversarial regularizers we discuss later (section “[Adversarial Regularization](#)”). Well-established approaches for learning priors from data include dictionary learning and bilevel learning, as outlined in section “[Shallow Learned Regularizers](#)”. Recently, attention has shifted to methods based on deep neural networks (Kobler et al. 2017; Adler and Öktem 2017, 2018; Jin et al. 2017; Li et al. 2020; Lunz et al. 2018; Kobler et al. 2020). The majority of approaches is based on a direct parametrization of the reconstruction operator $\Psi_\Theta(\cdot, A) : \mathcal{Y} \rightarrow \mathcal{X}$ that is trained using a loss function $\ell : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and empirical risk minimization

$$\min_{\Theta} \sum_i \ell(\Psi_\Theta(y^i, A), \tilde{x}^i). \quad (4)$$

For those methods, the trained network $\Psi_\Theta(\cdot, A)$ can be applied directly to new measurements at inference. On the other hand, approaches based on learning a regularization functional \mathcal{R}_Θ typically separate between the training procedure of \mathcal{R}_Θ and the reconstruction step, using a variational functional of the form (2) or a similar functional for reconstruction. While those methods in general perform slightly worse than methods based on a direct parametrization that are trained *end-to-end* (Adler and Öktem 2018), they often allow for stability and convergence guarantees and enable a statistical interpretation of the learned functional. In this survey, we will in particular discuss Network Tikhonov (NETT) in section “[Regularization Properties of Learned Regularizers](#)”, adversarial regularizers in section “[Adversarial Regularization](#)”, and total deep variation in section “[Total Deep Variation](#)”.

Some hybrid approaches invoke a variational problem (or an early stopped version of it) but aim at parametrizing the gradient of the regularization functional instead of the functional directly (Kobler et al. 2017; Romano et al. 2017). While these methods have shown very good reconstruction results, we will omit them in our summary, focusing instead on methods that parametrize a regularization functional directly. In particular, approaches like regularization by denoising (RED) cannot always guarantee that the learned gradient is in fact the gradient of some functional.

Finally, deep image priors (Ulyanov et al. 2018) use the network architecture itself, without prior training, as a regularization term. These methods however are crucially reliant on early stopping, and we will not discuss them in detail here, but instead, refer to Ulyanov et al. (2018) for details.

Outline In this summary, we first give a brief overview over classical approaches at learning regularization functionals that do not make use of deep neural networks in section “[Shallow Learned Regularizers](#)”. We then discuss three approaches for using neural networks as regularization functionals in detail in section “[Deep Regularizers](#)”: Network Tikhonov in section “[Regularization Properties of Learned](#)

[Regularizers](#)”, adversarial regularizers in section [“Adversarial Regularization”](#), and total deep variation in section [“Total Deep Variation”](#). We finish this review by giving a short summary and outline of potential for future research in section [“Summary and Outlook”](#).

Shallow Learned Regularizers

In this section, we review some approaches for learning a regularization functional that do not make use of neural networks. We in particular discuss bilevel learning as a technique for parameter optimization in regularization functionals and dictionary learning as a prominent unsupervised approach.

Bilevel Learning

Given a training set of the form $\{(\tilde{x}^i, y^i)\}$ of some images \tilde{x}^i and associated measurements y^i , the bilevel problem of finding the optimal parameters Θ is given by

$$\begin{cases} \hat{\Theta} \in \arg \min_{\Theta} \sum_i [\ell(x_{\Theta}^i, \tilde{x}^i)] \\ x_{\Theta}^i := \arg \min_{x^i} \mathcal{D}(Ax^i, y^i) + \mathcal{R}_{\Theta}(x^i). \end{cases} \quad (5)$$

The generic framework of (5) has been used in various contexts to learn a regularization functional \mathcal{R}_{Θ} . A prominent example is learning TV-type regularizers that consist of one or multiple regularization functionals based on the ℓ^1 norm of the gradient or smoothed versions thereof (Kunisch and Pock 2013; Calatroni et al. 2012). More complex regularization functionals, such as the field of experts (FoE) model (Roth and Black 2005), have also been trained using bilevel learning (Chen et al. 2013). In this setting, a linear combination of filters is learned from data.

Deriving sharp optimality conditions for bilevel learning generally requires the lower-level problem in (5) to be sufficiently regular. Under sufficient smoothness assumptions on the inner problem, optimality conditions can be established, and the problem (5) can be solved utilizing suitable techniques from PDE-constrained optimization.

In general, solving (5) is hard, with the problem being non-convex in Θ even in simple scenarios such as the Operator $A = Id$ being the identity (Arridge et al. 2019), making it challenging to scale bilevel techniques to highly parametric regularization functionals such as those given by neural networks. However, the concept of empirical risk minimization, i.e., of using a term of the form

$$\hat{\Theta} \in \arg \min_{\Theta} \sum_i [\ell(x_{\Theta}^i, \tilde{x}^i)]$$

is wildly used to train neural networks, and we will see an approach that utilizes a term of this form to train a deep regularization functional in the chapter on total deep variation (Kobler et al. 2020).

Dictionary Learning

Dictionary learning is based on the concept that the model parameter has a sparse representation in a some dictionary D . Approaches for dictionary learning (Aharon et al. 2006; Dabov et al. 2007; Xu et al. 2012) can be classified by the strategy taken to learn the dictionary D , which can be defined a priori in an analytical form, can be learned before reconstruction from data, or can be generated at reconstruction time, where the latter is mostly used in patch-based approaches.

A common approach in this context is sparse dictionary learning, aiming at learning a dictionary S from a collection of samples $\tilde{x}_i \in \mathcal{X}$ by minimizing the functional

$$\arg \min_{D, \xi} \sum_i \ell_{\mathcal{X}}(\tilde{x}_i, D\xi_i) + \mu \|\xi\|_1, \quad (6)$$

where $D : \mathcal{E} \rightarrow \mathcal{X}$ is a matrix containing the atoms of the dictionary in its columns and ξ_i is the representation of \tilde{x}_i in the dictionary D . The distance on image space $\ell_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ can, for example, be chosen to be ℓ_2 . The ℓ_1 penalty term $\|\xi\|_1$ is chosen as a convex relaxation of a sparsity constraint on the representation ξ_i that limits $\|\xi_i\|_0 < s$ in its non-relaxed form. This formulation allows to learn a dictionary D that can represent each image x_i sparsely. Once learned, it can be used as a sparsity penalty during reconstruction, for example, by solving the problem

$$\arg \min_x \|AD\xi - y\| + \lambda \|\xi\|_1 \quad (7)$$

at reconstruction time, leading to the reconstruction $x = D\xi$. A drawback of this approach is that the sparsity level s , parametrized by μ in the relaxed formulation, needs to be chosen beforehand. This can be challenging as a too low sparsity level will not allow the dictionary to capture details of the images \tilde{x}_i , while a too high sparsity level will lead to a that does not act as an efficient regularizer at reconstruction time.

Finally, we note that when learning a dictionary of sparse representations from a large body of training samples, only *unsupervised* training data samples are required. To be more precise, we require access to a collection of \tilde{x}^i of true images only, without requiring any pairing to corresponding measurements y^i . This makes training data more readily accessible in this context. We will later see how other learned distribution-based approaches, in particular the adversarial regularizer discussed in section “[Adversarial Regularization](#)”, inherit this property.

Deep Regularizers

In this section, we move from shallow to deep regularization functionals, presenting three recent approaches for training a deep neural network as a regularization functional. These works are motivated by the established theory for variational regularization outlined above, some building on well-posedness and stability results, some on the statistical viewpoint of inverse problems. In terms of training strategy, some approaches build on the paradigm of empirical risk minimization previously seen in the context of bilevel learning, while some take an unsupervised approach to the problem, much like dictionary learning. We highlight that these approaches put an emphasis on statistical understanding, cross-modality flexibility, stability, and convergence results, separating them from the majority of deep learning approaches that directly parametrize a reconstruction operator. While those obtain state-of-the-art results, they offer little room for theoretical understanding.

Regularization Properties of Learned Regularizers

In the network Tikhonov (NETT) paper (Li et al. 2020), the authors propose one of the earliest approaches at learning a regularization functional from data using tools from deep learning. The authors put a strong emphasis on deducing stability results for the resulting algorithm that resemble the classical theory of variational regularization (Engl et al. 1996).

The authors study the inverse problem associated with (1) in the general setting of $(\mathcal{X}, \|\cdot\|)$ and $(\mathcal{Y}, \|\cdot\|)$ being reflexive Banach spaces with domain D . We denote by δ the noise level such that the noise ϵ satisfies $\|\epsilon\| \leq \delta$. The authors restrict their study to regularization functional of the form

$$\mathcal{R}_\Theta(x) = \phi(\Psi_\Theta(x)), \quad (8)$$

where $\phi : \mathbb{X}_L \rightarrow [0, \infty]$ is a scalar functional and $\Psi_\Theta : \mathcal{X} \rightarrow \mathbb{X}_L$ is a neural network of depth L , with parameters Θ . An example of a regularization functional of this form is given by a neural network $\Psi_\Theta : \mathcal{X} \rightarrow \mathcal{X}$ that maps an input image to some other element in image space, which is then mapped to a scalar via the ℓ_2 norm, $\Phi = \|\cdot\|_2$. The network Ψ_Θ is as usual given by a concatenation of affine functions and pointwise nonlinear activation functions that we denote by σ . Given this regularization function, an image x can be reconstructed from measurements y by minimizing the variational functional

$$\mathcal{J}_{\lambda, y_\delta}(x) := \mathcal{D}(A(x), y_\delta) + \lambda \mathcal{R}_\Theta(x) \rightarrow \min_{x \in D} \quad (9)$$

where $\mathcal{D} : Y \times Y \rightarrow [0, \infty]$ is the data consistency term.

A key contribution of the authors is the result that, under certain assumptions, reconstructions via (9) provide a stable solution scheme for (1). In addition to the

well-posedness and weak convergence, the authors provide a complete analysis of norm-convergence and various convergence rates results, introducing the absolute Bregman distance as a new generalization of the standard Bregman distance from the convex to the non-convex setting. In the following, we report their key results.

To start, we discuss convergence of NETT regularization. To this end, the authors make the following assumptions.

Assumption 1.

- Network regularizer \mathcal{R} :
 - the regularizer is defined by (8);
 - The linear part of the affine layers in Ψ_Θ is bounded;
 - The activation functions σ are weakly continuous;
 - The functional ϕ is weakly lower semi-continuous.
- Data consistency term \mathcal{D} :
 - For some $\tau > 1$ we have $\forall y_0, y_1, y_2 \in Y : \mathcal{D}(y_0, y_1) \leq \tau \mathcal{D}(y_0, y_2) + \tau \mathcal{D}(y_2, y_1)$;
 - $\forall y_0, y_1 \in Y : \mathcal{D}(y_0, y_1) = 0 \iff y_0 = y_1$;
 - $\forall (y_k)_{k \in \mathbb{N}} \in Y^{\mathbb{N}} : y_k \rightarrow y \implies \mathcal{D}(y_k, y) \rightarrow 0$;
 - The functional $(x, y) \mapsto \mathcal{D}(A(x), y)$ is sequentially lower semi-continuous.
- Coercivity condition:
 - $\mathcal{R}_\Theta(\cdot)$ is coercive, that is $\mathcal{R}_\Theta(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$.

The conditions on the network regularizer guarantee the lower semicontinuity of the regularizer. Both those conditions and the assumptions on the data consistency term are not very restrictive and are satisfied by most natural choices of consistency term (such as the ℓ^2 distance) and network architectures. We hence point the reader's attention to the coercivity condition, which is not straightforward and will be violated by standard network architectures without introducing further restrictions. In particular, in the following chapter on adversarial regularizers, we will see that the authors allow a class of networks that can violate this coercivity assumption and they hence rely on the data term for providing the coercivity that is crucial for theoretical stability guarantees. The authors of Li et al. (2020) describe several ways to obtain coercivity by tuning the architecture of the network. The proposed approaches include skip and residual connections as well as layer-wise coercivity constraints using, for example, leaky ReLU or max-pooling.

We now state a key result of the paper that can be deduced under the above assumptions, demonstrating that they are sufficiently powerful to obtain results similar to the classical stability theory for variational problems with convex regularization functionals.

Theorem 1 (Well-posedness of CNN-regularization (Thm 2.6 Li et al. 2020)).
Let Assumption 1 be satisfied. Then the following assertions hold true:

- *Existence:* For all $y \in Y$ and $\lambda > 0$, there exists a minimizer of $\mathcal{F}_{\lambda,y}$;
- *Stability:* If $y_k \rightarrow y$ and $x_k \in \arg \min \mathcal{F}_{\lambda,y_k}$, then weak accumulation points of $(x_k)_{k \in \mathbb{N}}$ exist and are minimizers of $\mathcal{F}_{\lambda,y}$;
- *Convergence:* Let $x \in \mathcal{X}$, $y := A(x)$, $(y_k)_{k \in \mathbb{N}}$ satisfy $\mathcal{D}(y_k, y), \mathcal{D}(y, y_k) \leq \delta_k$ for some sequence $(\delta_k)_{k \in \mathbb{N}} \in (0, \infty)^{\mathbb{N}}$ with $\delta_k \rightarrow 0$, suppose $x_k \in \arg \min_x \mathcal{F}_{\lambda(\delta_k),y_k}(x)$, and let the parameter choice $\lambda: (0, \infty) \rightarrow (0, \infty)$ satisfy

$$\lim_{\delta \rightarrow 0} \lambda(\delta) = \lim_{\delta \rightarrow 0} \frac{\delta}{\lambda(\delta)} = 0 \tag{10}$$

Then the following holds:

- Weak accumulation points of $(x_k)_{k \in \mathbb{N}}$ are $\mathcal{R}_{\Theta}(\cdot)$ -minimizing solutions of $A(x) = y$;
- $(x_k)_{k \in \mathbb{N}}$ has at least one weak accumulation point x_+ ;
- Any weakly convergent subsequence $(x_{k(n)})_{n \in \mathbb{N}}$ satisfies $\mathcal{R}_{\Theta}(x_{k(n)}) \rightarrow \mathcal{R}_{\Theta}(x_+)$;
- If the $\mathcal{R}_{\Theta}(\cdot)$ -minimizing solution of $A(x) = y$ is unique, then $x_k \rightarrow x_+$ (weak convergence).

This theorem establishes the classical results of existence and stability of solutions along with convergence of solutions to the true image as the noise level $\delta \rightarrow 0$ for reconstructions via the variational problem (9). The authors strengthen the convergence results by introducing the notion of total nonlinearity as an additional assumption. We refer to Theorem 2.11 in the paper for details as well as proof of the theorem.

Finally, the authors also establish convergence rates in the absolute Bregmann distance. We refer the reader to Section 3 in Li et al. (2020) for further details on the resulting theorems as well as on the conditions necessary to obtain convergence rates in the absolute Bregman distance.

To summarize, we note that the combination of theoretical results deduced in Li et al. (2020) forms the most extensive theoretical analysis of a learned regularization functional conducted so far, including stability and convergence results as well as convergence rates. However, in order for the theorems to apply, one requires various constraints on the network architecture that need to be imposed either by network design or during training. Enforcing those might potentially be harmful in terms of model performance, but comes with the benefit of guaranteed stability and convergence as shown above.

Training scheme and results While the main emphasis of the paper is on an extensive stability and convergence theory, the authors also propose an algorithm for training a regularization functional. In particular, they choose a parametrization of the form

$$\mathcal{R}_{\Theta}(x) = \sum_i \|\Psi_{\Theta,i}(x)\|_q^q,$$

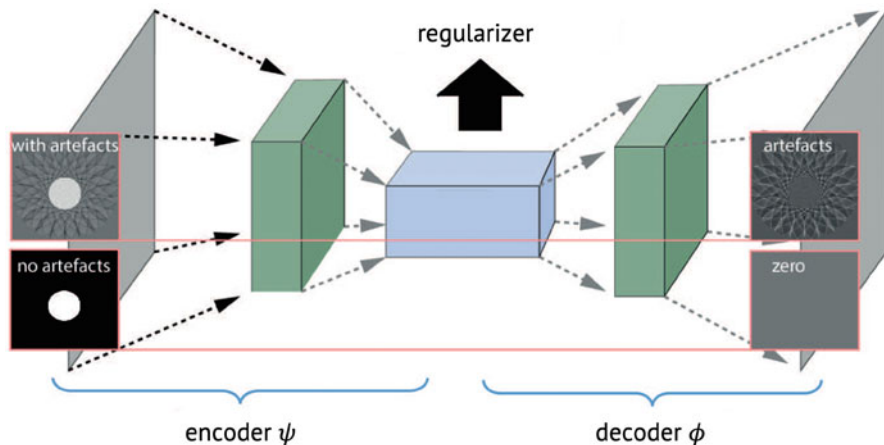


Fig. 1 Training setup in NETT. (Taken from Li et al. 2020)

where $\Psi_{\theta,i}(x)$ denotes the i the component of $\Psi_{\theta}(x)$. In order to train Ψ_{θ} , the authors propose an encoder-decoder-based architecture that invokes a decoder network Φ in addition to the encoder Ψ_{θ} . The joint architecture is trained to detect the characteristic artefacts in unregularized reconstructions, as shown in Fig. 1. The heuristic motivation behinds is that the resulting network is able to decompose the parts of a given reconstruction that are part of the underlying images and the ones that are reconstruction artefacts only. By penalizing the ℓ_q norm of the noise part only, typical noise patterns are suppressed during reconstruction without introducing artefacts in the underlying image. Note the similarity to adversarial training as discussed in the next section on adversarial regularizers (Lunz et al. 2018).

The authors employ subgradient descent for solving the minimization problem (9) and show results for photoacoustic tomography (PAT), as seen in Fig. 2.

Note that the authors and further researchers have published a variety of extension papers based on the NETT theory discussed here. These papers include discussions on improved training schemes and architectures as well as on further fields of applications (Obmann et al. 2020a,b). The NETT paper (Li et al. 2020) can be viewed as the theoretical foundation and first result in this direction.

Adversarial Regularization

The paper “adversarial regularizers” (Lunz et al. 2018) introduces a regime for learning regularization functionals, training the functional to reduce the *distributional* distance between reconstructions and true images. While there are similarities between the training regimes in this paper and in the previously discussed NETT (Li et al. 2020) approach, the authors of the adversarial regularizer paper focus their

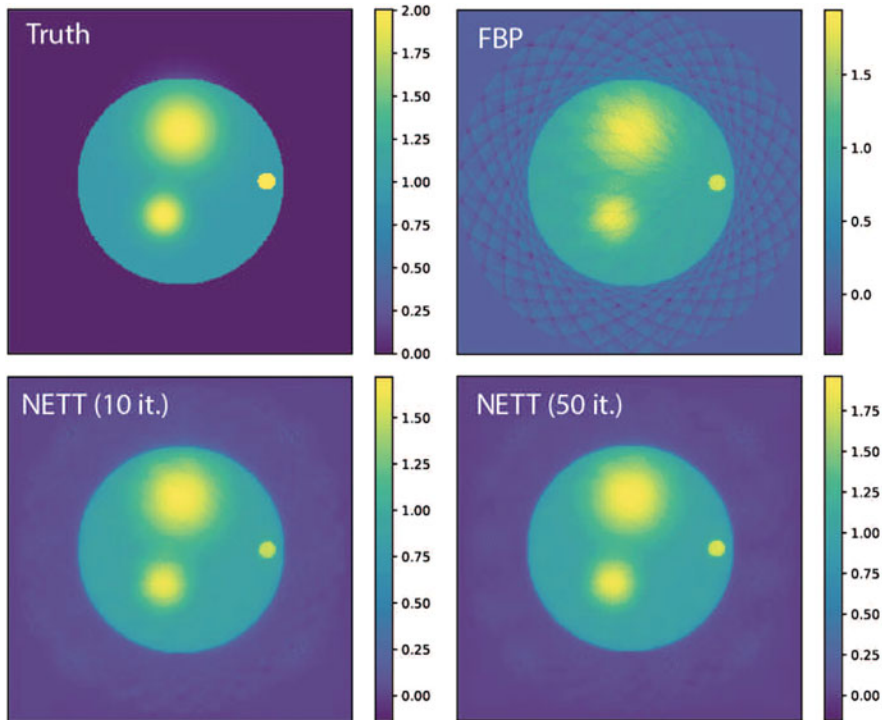


Fig. 2 Results for photoacoustic tomography reconstruction using the NETT approach on the Shepp-Logan phantom. (Taken from Li et al. 2020)

viewpoint on a statistical and distributional understanding of the learned regularization functional in contrast to the clear emphasis on convergence results in Li et al. (2020). The main contribution of the paper is a training technique the authors entitle adversarial training. Using this approach, the authors are able to train very complex regularization functionals. While this training regime is not necessarily limited to regularization functionals given by neural networks, it is particularly appealing when training complex functionals such as those parametrized by neural networks.

The authors rely on two distributions, given by their empirical counterparts of $\tilde{x}_i \in \mathcal{X}$ independent samples from the distribution of ground truth images \mathbb{P}_r and by $y_i \in \mathcal{Y}$ independent samples from the distribution of measurements \mathbb{P}_Y .

The authors then consider the mapping of the distribution \mathbb{P}_Y to a distribution on image space by applying via a pseudo-inverse A_δ^\dagger , yielding the distribution $\mathbb{P}_n = (A_\delta^\dagger)_\# \mathbb{P}_Y$ of distorted reconstructions. Here, $\#$ denotes the push-forward of measures, i.e., $A_\delta^\dagger Y \sim (A_\delta^\dagger)_\# \mathbb{P}_Y$ for $Y \sim \mathbb{P}_Y$. Samples drawn from \mathbb{P}_n are corrupted with noise that depends both on the noise model e and on the operator A .

The authors argue that a good regularization functional \mathcal{R}_Θ is able to tell apart the distributions \mathbb{P}_r and \mathbb{P}_n . The authors use this as a motivation to choose the loss functional for training a neural network Ψ_Θ that directly parametrizes the regularization functional $\mathcal{R}_\Theta = \Psi_\Theta$ as

$$\mathbb{E}_{X \sim \mathbb{P}_r} [\Psi_\Theta(X)] - \mathbb{E}_{X \sim \mathbb{P}_n} [\Psi_\Theta(X)] + \mu \cdot \mathbb{E} \left[\left(\|\nabla_x \Psi_\Theta(X)\| - 1 \right)_+^2 \right], \quad (11)$$

where the last term in the loss functional serves to enforce the trained network Ψ_Θ to be Lipschitz continuous with constant one.

Written using the empirical distributions instead, the training loss (11) reads as

$$\sum_i \Psi_\Theta(\tilde{x}_i) - \sum_i \Psi_\Theta(A^\dagger y_i) + \mu \sum_i \left(\|\nabla_x \Psi_\Theta(\xi_i)\| - 1 \right)_+^2,$$

where the points ξ are chosen randomly on the straight line between \tilde{x}_i and $A^\dagger y$.

The authors make this choice of penalty term for its connection to the Wasserstein distance between the distributions \mathbb{P}_r and \mathbb{P}_n that allows them to deduce the following theorem on the gradient flow over a perfectly trained regularization functional. Here, perfectly trained refers to the functional being 1-Lipschitz and perfectly minimizing the Wasserstein distance in the Kantorovich duality formulation

$$\text{Wass}(\mathbb{P}_r, \mathbb{P}_n) = \sup_{f \in 1\text{-Lip}} \mathbb{E}_{X \sim \mathbb{P}_n} [f(X)] - \mathbb{E}_{X \sim \mathbb{P}_r} [f(X)]. \quad (12)$$

Consider the distribution $\mathbb{P}_\eta := (g_\eta)_\# \mathbb{P}_n$ of samples obtained after a single gradient descent over Ψ_Θ of step of size η , starting from noisy reconstructions.

$$g_\eta(x) := x - \eta \cdot \nabla_x \Psi_\Theta(x). \quad (13)$$

The authors show the following theorem.

Theorem 2 (Wasserstein distance descent (Thm 1 Lutz et al. 2018)). *Assume that $\eta \mapsto \text{Wass}(\mathbb{P}_r, \mathbb{P}_\eta)$ admits a left and a right derivative at $\eta = 0$ and that they are equal. Then*

$$\frac{d}{d\eta} \text{Wass}(\mathbb{P}_r, \mathbb{P}_\eta)|_{\eta=0} = -\mathbb{E}_{X \sim \mathbb{P}_n} \left[\|\nabla_x \Psi_\Theta(X)\|^2 \right].$$

The authors strengthen this result to

$$\frac{d}{d\eta} [\Psi_\Theta(g_\eta(X))]|_{\eta=0} = -1 \quad (14)$$

under weak assumptions.

The authors are hence able to show that the regularization functional trained via (11) can in fact optimally reduce the Wasserstein distance between reconstructions and ground truth images, at least at the initial step of the gradient descent scheme. The authors extend their analysis by deducing an explicit form of the regularization functional in the specific scenario of the true distribution being concentrated along a manifold $\mathcal{M} \subset \mathcal{X}$.

Assumption 2. Denote by

$$P_{\mathcal{M}} : D \rightarrow \mathcal{M}, \quad x \rightarrow \arg \min_{y \in \mathcal{M}} \|x - y\| \tag{15}$$

the data manifold projection, where D denotes the set of points for which such a projection exists. We assume $\mathbb{P}_n(D) = 1$. This can be guaranteed under weak assumptions on \mathcal{M} and \mathbb{P}_n . We make the assumption that the measures \mathbb{P}_r and \mathbb{P}_n satisfy

$$(P_{\mathcal{M}})_{\#}(\mathbb{P}_n) = \mathbb{P}_r \tag{16}$$

i.e., for every measurable set $A \subset \mathcal{X}$, we have $\mathbb{P}_n(P_{\mathcal{M}}^{-1}(A)) = \mathbb{P}_r(A)$

The authors motivate this as a low-noise assumption under which it is guaranteed that the distortions of the true data present in the distribution of pseudo-inverses \mathbb{P}_n are sufficiently well behaved to recover the distribution of true images from noisy ones by projecting back onto the manifold. Under this assumption, the authors prove the following theorem.

Theorem 3 (Data Manifold Distance (Thm 2 Lunz et al. 2018)). *Under Assumption 2, a maximizer to the functional*

$$\sup_{f \in 1\text{-Lip}} \mathbb{E}_{X \sim \mathbb{P}_n} f(X) - \mathbb{E}_{X \sim \mathbb{P}_r} f(X) \tag{17}$$

is given by the distance function to the data manifold

$$d_{\mathcal{M}}(x) := \min_{y \in \mathcal{M}} \|x - y\| \tag{18}$$

The authors motivate the theorem as a consistency result, demonstrating that the approach yields reasonable regularization functionals in the particular setting of the theorem.

The paper also contains stability result with a similar flavor, the NETT paper in Theorem 1. The analysis is however less exhaustive and requires stronger assumptions on the operator A , making it less readily applicable to all inverse problems than Theorem 1. On a technical level, the key difference is that the NETT paper develops assumptions that ensure that the learned regularization functional

is itself coercive, whereas the adversarial regularizer relies on the coercivity of the data term. The latter makes use of the additional 1-Lipschitz property of the regularization functional \mathcal{R}_Θ to ensure that a coercive regularization functional yields a coercive variational functional even if the regularization functional is not bounded from below. In the following, we state the results shown for adversarial regularizers in Lunz et al. (2018) and refer to the paper for the proof and further details.

Theorem 4 (Stability (Thm 3 Lunz et al. 2018)). *Let y_n be a sequence in Y with $y_n \rightarrow y$ in the norm topology and denote by x_n a sequence of minimizers of the functional*

$$\arg \min_{x \in X} \|Ax - y_n\|^2 + \lambda \mathcal{R}_\Theta(x)$$

Under appropriate assumptions on the operator A (see Appendix of Lunz et al. 2018), x_n has a weakly convergent subsequence, and the limit x is a minimizer of $\|Ax - y\|^2 + \lambda \mathcal{R}_\Theta(x)$.

Computational Results The authors show results for the discussed algorithm for denoising and computed tomography reconstruction. They show improved results compared to classical approaches such as total variation (Engl et al. 1996), but do not match results obtained with end-to-end trained algorithms such as post-processing approach for computed tomography (Jin et al. 2017) or a DnCNN (Zhang et al. 2017) for denoising. The results in Fig. 3 show results on denoising, whereas Fig. 4 contains results for computed tomography reconstruction.

Total Deep Variation

The recent paper Kobler et al. (2020) follows the paradigm of *end-to-end* learning in order to obtain a regularization functional, using a distance functional between reconstruction and ground truth as training objective. In general, unrolling methods

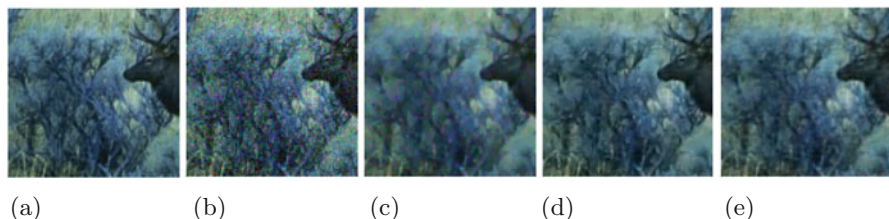


Fig. 3 Denoising results for the adversarial regularizer on the Berkeley Segmentation dataset (BSDS500). (Taken from Lunz et al. 2018). (a) Ground Truth. (b) Noisy Image. (c) TV. (d) Denoising N.N. (e) Adversarial Reg.

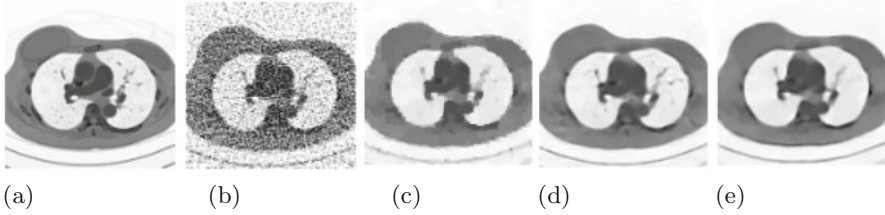


Fig. 4 Reconstruction from simulated CT measurements on the LIDC dataset using adversarial regularizers. (Taken from Lunz et al. 2018). (a) Ground Truth. (b) FBP. (c) TV. (d) Post-Processing. (e) Adversarial Reg.

such as Adler and Öktem (2017), Meinhardt et al. (2017), and Kobler et al. (2017) recover an image x_T from measurements y by applying

$$x_{n+1} = x_n - \lambda \Psi_{\Theta}(A^t(Ax_n - y), x_n), \quad (19)$$

where the iteration is typically initialized with a pseudo-inverse x_0 and stopped after a fixed predefined number of steps N . The parameters Θ are trained by minimizing a loss functional

$$\sum_i \ell(x_N^i, x_T^i) \quad (20)$$

over the parameters Θ for a collection of samples $\{x_T^i, y^i\}$ and a notion of distance ℓ that is typically chosen to be the ℓ^2 distance. Various approaches differ in their choice of parametrization of Ψ_{Θ} , ranging from architectures that do not further restrict the mapping properties of Ψ_{Θ} to those that explicitly separate out a gradient terms obtained from the data term and the image prior, leading to the form

$$\Psi_{\Theta}(A^t(Ax_n - y), x_n) = A^t(Ax_n - y) + \mu \Phi_{\Theta}(x_n). \quad (21)$$

While these methods have shown to yield high-quality reconstructions, they cannot readily be understood using the viewpoint of variational regularization, as the regularization or image prior is implicitly contained in the mapping properties of the network Ψ_{Θ} . Even if parameterized as in (21), the network parametrizes the gradients of an implicit regularization functional rather than the functional directly.

An additional challenge in bridging the gap between unrolling based methods and a variational methods lies in the fixed choice of iterations N that is typically small and prohibits viewing x_N as the result of a minimization of a variational problem.

The authors of Kobler et al. (2020) bridge these problems by introducing two novel contributions: firstly, instead of parametrizing the gradient of the regularization functional, the functional itself is parametrized directly. While this makes training slightly more challenging, requiring double backpropagation for minimization, it yields a true regularization functional that can be interpreted as

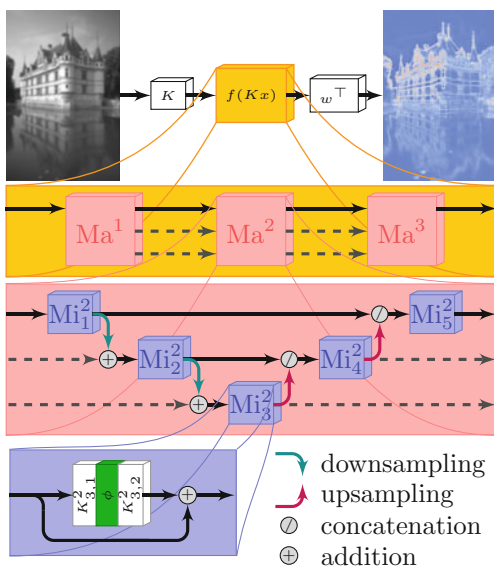
a prior on image distribution. Secondly, instead of fixing the number of gradient steps a priori, the authors introduce an optimal stopping time that allows for a flexible number of gradient descent iterations on the variational functional. While this still leaves a gap to classical regularization functionals that do not necessarily require a stopping criterion, the flexibility in the number of iterations makes the *total deep variation* approach the closest candidate for a generic method to yield a deep regularization functional that is trained by differentiating through the minimization of the corresponding variational functional.

Architecture For an image $x \in \mathbb{R}^{nC}$, where n denotes the number of pixels and C the number of channels, the authors parametrize a regularization functional \mathcal{R} of the form

$$\mathcal{R}_\Theta(x) = \sum_{i=1}^n r(x, \Theta)_i, \quad r(x, \Theta) = \omega^T \mathcal{N}(Kx) \in \mathbb{R}^n. \quad (22)$$

Here, K denotes a zero-mean convolution kernel, ω is a learned weight vector contracting over channels but not over the spatial component, and \mathcal{N} is a multi-scale neural network that is inspired by a UNet (Ronneberger et al. 2015) architecture. The authors employ a smooth log-student-t-distribution of the form $\Phi(x) = \frac{1}{2\mu} \log(1 + \mu x^2)$ as activation function, leading to a smooth regularization functional. This is advantageous for the double backpropagation used for minimization as discussed later (Fig. 5).

Fig. 5 Architecture of the regularization functional for TDV. (Taken from Kobler et al. 2020)



Training procedure We assume to be given a training set $\{(\tilde{x}^i, y^i, x_0^i)\}$ of ground truth image $\tilde{x}^i \in \mathcal{X}$, measurement $y^i \in \mathcal{Y}$ and an initial guess, such as a pseudo-inverse of y_i , $x_0^i \in \mathcal{X}$. The authors cast the training process as an optimal control problem, introducing an optimal time horizon T . Using a fixed time discretization level $S \in \mathbb{N}$, their sampled objective function on the training set $\{(\tilde{x}^i, y^i, x_0^i)\}$ reads as

$$\inf_{T \in [0, T_{Max}]} \left\{ \frac{1}{N} \sum_{i=1}^N l(x_s^i - \tilde{x}^i) \right\}, \quad (23)$$

subject to the state equation

$$x_s^{i+1} = x_s^i - \frac{T}{S} A^t (Ax_{s+1}^i - y^i) - \frac{T}{S} \nabla \mathcal{R}_\Theta(x_s^i), \quad (24)$$

where $l : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ denotes a loss functional in (23), which is typically chosen as either the ℓ^2 or ℓ^1 loss. An equivalent formulation of the state equation that is solved for x_s^{i+1} reads as

$$x_s^{i+1} = \left(\text{Id} + \frac{T}{S} A^t A \right)^{-1} \left(x_s^i + \frac{T}{S} \left(A^t y^i - \nabla \mathcal{R}_\Theta(x_s^i) \right) \right) \quad (25)$$

The training objective is simultaneously minimized for the time horizon T and the parameters Θ that determine the form of the regularization functional. The stochastic ADAM optimizer is used for minimization. The zero-mean constraint on the regularization functional is enforced projection after every minimization step. Note that differentiating (23) with respect to (T, Θ) involves derivatives of the regularization functional $\mathcal{R}_\Theta(x)$ with respect to both Θ and x . These terms are handled in a numerically efficient way using the *double backpropagation* algorithm. The algorithm can be applied in this context as the architecture and activation functions used have been chosen to be C^2 . The application of double backpropagation separates this work from earlier attempts at learning a regularization functional with a loss functional of the form (20).

The authors also derive various theorems to characterize the solutions of (23).

Theorem 5 (Existence of a solution (Thm 2.1 Kobler et al. 2020)). *The time continuous version of (23) alongside its corresponding state equation (24) admits a solution in the sense that the infimum is attained.*

The authors provide a characterization of the optimal solution in terms of the adjoint state in Theorem 3.1 in Kobler et al. (2020). They also include a sensitivity analysis of the results with respect to changes in the model parameters (T, Θ) , bounding changes in the reconstruction by the differences in model parameters and some experiment specific quantities like the Lipschitz norm of the regularization

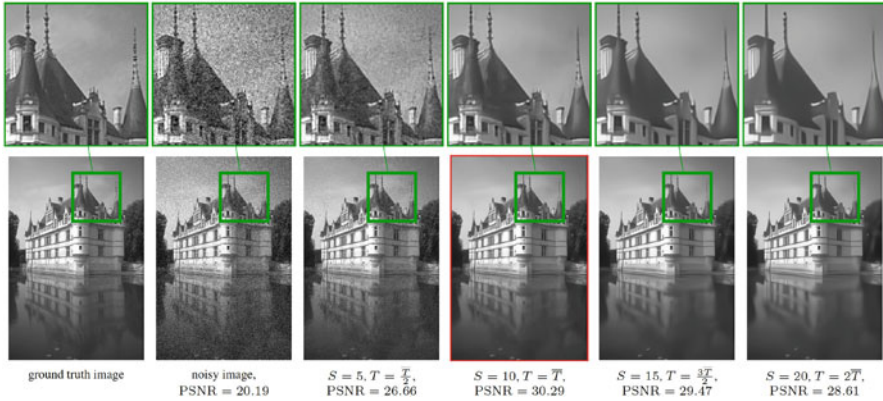


Fig. 6 TDV results for denoising with various choices for the time horizon. (Taken from Kobler et al. 2020)

functional \mathcal{R}_θ , the norm of the gradient of the regularization functional, and various others. Details can be found in Theorem 3.2 in the paper.

Results The authors show results for their TDV approach on a variety of inverse problems. For denoising, the approach is able to outperform approaches like BM3D (Dabov et al. 2007) as well as some end-to-end trained approaches like DnCNN (Zhang et al. 2017), but slightly underperforms compared to FOCNet (Jia et al. 2019). The latter has roughly one hundred times more parameter than the TDV approach. Results for denoising are shown in Fig. 6 for various choices of time discretization S and time horizon T . As expected, choosing the time horizon lower than the learned optimal parameter leads to under-regularization, while choosing it higher leads to over-regularization.

This chapter also discusses applications of the approach to medical imaging, demonstrating that a prior trained for computed tomography reconstruction of abdominal CT images can be readily applied for MRI reconstruction of knee images – a task that differs both in the imaging modality and in the characteristics of the images occurring. This shows that TDV generalizes well between different tasks and image characteristics. Results for MRI reconstruction can be seen in Fig. 7.

Summary and Outlook

Conclusion

We have discussed various approaches for training neural networks as regularization functionals that have been proposed in the past years. Network Tychonov (NETT) focuses on deriving stability and convergence results for regularization functionals

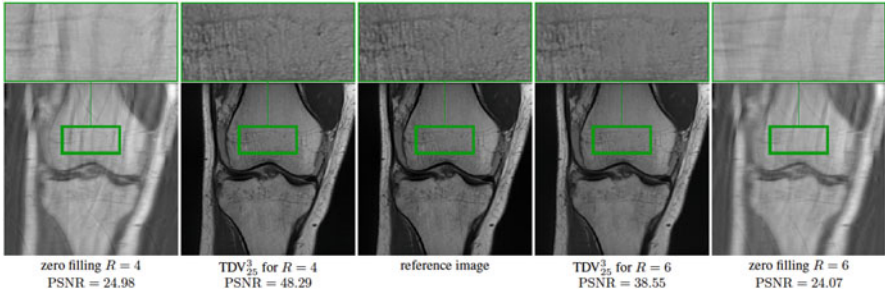


Fig. 7 TDV results for MRI reconstruction. (Taken from Kobler et al. 2020)

Table 1 Comparison of learned regularization approaches discussed in this survey

	Stability theory	Data structure	Performance
NETT	Fully developed, applies to wide variety of inverse problems	Paired, unpaired extensions	No comparison to SOTA in original publication, follow-up work (Obmann et al. 2020b)
Adv. Reg.	Stability theory only applicable to some inverse problems	Unpaired training data suffices	Strong improvements on classical approaches (TV), slightly under SOTA for supervised reconstruction
TDV	No equivalent results for classical variational stability derived in paper	Fundamentally reliant on paired training data	Essentially SOTA performance, generalizability between tasks demonstrated

based on neural networks, allowing to deduce guarantees on the behavior of the resulting algorithm. The main contribution of adversarial regularizers and total deep variation lies in the proposal of novel schemes for training regularization functionals. The first introduces an approach that is based on training the network to tell apart ground truth images from noisy reconstructions, yielding an algorithm that can be trained in an unsupervised manner. The latter investigates the idea of supervised training of regularization functionals, made possible by the use of double backpropagation and the introduction of an optimal stopping time. We now turn to comparing the algorithms presented in terms of their results on stability, the structure of training data needed, and the performance demonstrated. This discussion is summarized in Table 1.

Stability Results The NETT paper contains an extensive stability analysis that is applicable to a wide variety of inverse problems. On a technical level, the theory does not make any assumptions on the data term being coercive and ensures coercivity by discussing sufficient conditions for the learned regularization

functional to be coercive. This in particular allows the application of the theory to ill-posed inverse problems. The adversarial regularizer paper on the other hand makes strong assumptions on the properties of the forward operator, which can be violated in the context of ill-posed inverse problems. Most of the theoretical analysis in the paper focuses on discussing the effects of the learned regularization functional on the distribution of reconstructions instead of focusing on an instance-level stability theory. For the total deep variation approach, the authors include a discussion in terms of optimal control theory as well as stability with respect to changes in the training dataset, but do not derive stability results that are equivalent to the classical stability theory for inverse problems.

Training Data Both the NETT and the TDV approach rely on paired training data consisting of measurements and their corresponding ground truth images. While the first one can be extended to an unpaired setting when changing the training scheme (Obmann et al. 2020b), TDV is fundamentally dependent on paired data. Looking at marginals of distributions only, the adversarial regularizer approach can naturally handle unpaired training data.

Performance The authors of the NETT paper compare to backprojection only; an assessment on how the method compares to the state of the art is hence difficult. More extensive comparisons are included in the authors' more recent follow-up publications (Obmann et al. 2020a,b). The adversarial regularizer has been demonstrated to clearly outperform classical regularization techniques like total variation regularization, but does not quite reach the performance of state-of-the-art reconstruction methods that are trained with supervised data. The authors of TDV report results that are essentially state of the art and also demonstrate generalizability of the learned regularization functional between different imaging tasks, a property not yet investigated in the other papers discussed.

Outlook

In future work, combining the viewpoints of NETT and adversarial regularizers or NETT and total deep variation could be an interesting direction to explore. This could yield an algorithm that is provably stable while still being built on the training heuristics proposed in adversarial regularizers and total deep variation, respectively. As an example, we are recently working on introducing convexity constraints on the adversarial regularizer, resulting in an algorithm with better stability and convergence guarantees.

Finally, building a regularization functionals that approximate the prior on the image distribution more directly and more closely than the approaches discussed in this survey is another possible line of research. Notable algorithms based on

generative models and in particular flow-based probabilistic models are being discussed within the research community for their potential to learn the image prior distribution without the need for any information on the operator or the specific noise distribution used.

References

- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**(12), 124007 (2017)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006)
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Calatroni, L., Cao, C., De Los Reyes, J.C., Schönlieb, C.-B., Valkonen, T.: Bilevel approaches for learning of variational imaging models. *RADON Book Series* **18** (2012)
- Chen, Y., Pock, T., Ranftl, R., Bischof, H.: Revisiting loss-specific training of filter-based MRFs for image restoration. In: *German Conference on Pattern Recognition*, pp. 271–281. Springer (2013)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*, vol. 375. Springer Science & Business Media, Dordrecht (1996)
- Jia, X., Liu, S., Feng, X., Zhang, L.: Focnet: a fractional optimal control network for image denoising. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6054–6063 (2019)
- Jin, K.H., McCann, M., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **26**(9), 4509–4522 (2017)
- Kobler, E., Klatzer, T., Hammernik, K., Pock, T.: Variational networks: connecting variational methods and deep learning. In: *German Conference on Pattern Recognition*, pp. 281–293. Springer (2017)
- Kobler, E., Effland, A., Kunisch, K., Pock, T.: Total deep variation for linear inverse problems (2020)
- Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. *SIAM J. Imaging Sci.* **6**(2), 938–983 (2013)
- Li, H., Schwab, J., Antholzer, S., Haltmeier, M.: NETT: solving inverse problems with deep neural networks. *Inverse Probl.* **36**, 065005 (2020)
- Lunz, S., Öktem, O., Schönlieb, C.-B.: Adversarial regularizers in inverse problems. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 31*, pp. 8507–8516. Curran Associates, Inc., Red Hook (2018)
- Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: using denoising networks for regularizing inverse imaging problems. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1781–1790 (2017)
- Obmann, D., Schwab, J., Haltmeier, M.: Deep synthesis regularization of inverse problems. *arXiv preprint arXiv:2002.00155* (2020a)
- Obmann, D., Nguyen, L., Schwab, J., Haltmeier, M.: Sparse aNETT for solving inverse problems with deep learning. *arXiv preprint arXiv:2004.09565* (2020b)
- Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (red). *SIAM J. Imaging Sci.* **10**(4), 1804–1844 (2017)

- Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
- Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 860–867. IEEE (2005)
- Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9446–9454 (2018)
- Xu, Q., Yu, H., Mou, X., Zhang, L., Hsieh, J., Wang, G.: Low-dose x-ray CT reconstruction via dictionary learning. *IEEE Trans. Med. Imaging* **31**(9), 1682–1697 (2012)
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)



Filter Design for Image Decomposition and Applications to Forensics

32

Robin Richter, Duy H. Thai, Carsten Gottschlich,
and Stephan F. Huckemann

Contents

Introduction	1156
Applications and Challenges for Automated Image Decomposition	1157
Diffusion Methods	1160
Fourier and Wavelet Methods	1160
Variational Problems	1162
Non-linear Spectral Decompositions	1164
Texture Information	1165
Machine Learning	1166
Adaptive Balancing	1167
Adapting the Data-Fidelity-Norm	1168
Connection to the G -Norm	1168
Other Choices of M	1169
Connections with Machine Learning	1169
Solving via the ADMM/AL-Algorithm	1170
Interpretation via a Feasibility Problem	1171
A General Learning Problem	1175
Filter Design Using Factor Families	1177
Conclusion	1178
References	1179

R. Richter (✉) · S. F. Huckemann (✉)
Felix-Bernstein-Institute for Mathematical Statistics in the Biosciences, University of Göttingen,
Göttingen, Germany
e-mail: robin.richter@mathematik.uni-goettingen.de; huckeman@math.uni-goettingen.de

D. H. Thai (✉)
Department of Mathematics, Colorado State University, Fort Collins, CO, USA
e-mail: duy.hoang-thai@mathematik.uni-goettingen.de

C. Gottschlich (✉)
Institute for Mathematical Stochastics, University of Göttingen, Göttingen, Germany
e-mail: gottschl@math.uni-goettingen.de

Abstract

Employing image filters in image processing applications, essentially matrix convolution operators, has been an active field of research since a long time, and it is so very much still today. In the first part, we give a brief overview of imaging methods with emphasis on applications in fingerprint recognition and shoeprint forensics. In the second part, we propose a generalized discrete scheme for image decomposition that encompasses many of the existing methods. Due to its generality, it has the potential to learn, for specific use cases, a highly flexible set of imaging filters that are related to one another by rather general conditions.

Keywords

Image decomposition · Variational methods · Texture · Forensics · Fingerprint recognition

Introduction

Image decomposition is one of the first and crucial steps in image analysis, be it decomposition into *signal* and *noise*, *foreground* and *background*, or more refined, such as decompositions into *cartoon*, *texture*, and *noise*. Often, under theoretical technical assumptions, precise objective functions are employed to this end; however, for specific applications, they are not a priori available. The latter is the case, for instance, when at crime scenes, latent fingerprints or shoeprints are to be compared to print scans taken from suspects at hand who are released immediately after. For expert comparison taking place afterwards, the quality of the scanned prints is decisive. This quality, however, can only be defined indirectly, for example, that improved quality is proportional to improved (lowered) error rates. In this application scenario, other image processing steps surface as well, namely, *image enhancement*, for example, of latent prints, and *image compression* to significant features, for example, in large databases.

In this chapter we give, guided by examples from forensics, a brief overview of image decomposition methods from the past to the present with emphasis on a unified viewpoint for some current challenges.

In acoustic signal processing, digital filter design has first been inspired by analog electric filtering circuits, and this has also inspired filter design for images. Images, however, have fundamentally different features than acoustic signals. While for the former Fourier decomposition was highly effective, image analysis required different types of analysis, for example, *Haar wavelet frames* for sharp edge modeling (Daubechies 1992; Mallat 2008). Other popular approaches are given by diffusion equations (Perona and Malik 1990; Weickert 1998) or minimization problems (Mumford and Shah 1989; Scherzer et al. 2009). This has led to the development of entirely new mathematical frameworks, often connected with one another (Steidl et al. 2004; Burger et al. 2016).

In this context, Chambolle and Pock (2016) give an overview of a multitude of optimization algorithms for a multitude of proposed minimization problems. Such reconstruction methods often *balance* between *data fidelity* and, possibly, several *reconstruction regularity* objectives. In this context, the assertion of unique optima and the development of convergent algorithms has spurred an abundance of publications, in particular when, as is often the case in realistic applications, linearity is relaxed and modeled by additional constraints. This has led to the development/application of iterative algorithms for *saddle points* of associated Lagrange functionals which are *augmented* to obtain strict convexity, which results in additional robustness (e.g., Bertsekas 1982; Eckstein and Bertsekas 1992; Wu and Tai 2010).

The advent of *machine learning* allowed to train modified regularization filters in view of specific application tasks, given larger databases for training and testing. When only moderately sized databases are available, as is the case, for example, in academic forensic sciences, learning methods improve by drawing on prior structure information at hand. For instance, in view of fingerprint analysis, it is a priori known that the object of interest comprises a fringe pattern, sometimes forking, of nearly constant frequency that follows a smooth orientation field, featuring only three types of singularities (Maltoni et al. 2009).

Considering minimization problems with a global balancing parameter, as a learning model, however, comes at a price such as the well-known *loss of contrast* dilemma: Removing highly oscillating structures while preserving steps of small intensity differences between otherwise flat structures cannot be simultaneously achieved (e.g., Figure 2 of Strong and Chan 2003). As a workaround, adaptive balancing filters have been introduced, localizing in the spatial or frequency domain (Osher et al. 2003; Buades et al. 2010; Bredies et al. 2013). In conclusion of this chapter, in generalization, we introduce a flexible, discrete learning model featuring a general *alternating direction method of multipliers* (ADMM) inspired algorithm based on a *feasibility problem*. This framework draws flexibility from decoupling involved families of filters from one another only requiring rather general regularity conditions.

It includes several of the abovementioned methods as special cases. For specific application scenarios at hand, suitable filter families can be learned. In application to forensics, we illustrate how to employ the new model for shoeprint decompositions. For shoeprint analysis, as detailed, challenges are much higher than for fingerprint analysis (which are still high), and scientifically based shoeprint image analysis is still in its very beginnings.

Applications and Challenges for Automated Image Decomposition

Very often, images contain an object of interest (or several) within a *region of interest* (ROI), for example, the area covered by a latent fingerprint or shoeprint in *forensics* (cf. Fig. 1), a tumor within an organ in *medical imaging*, faces observed

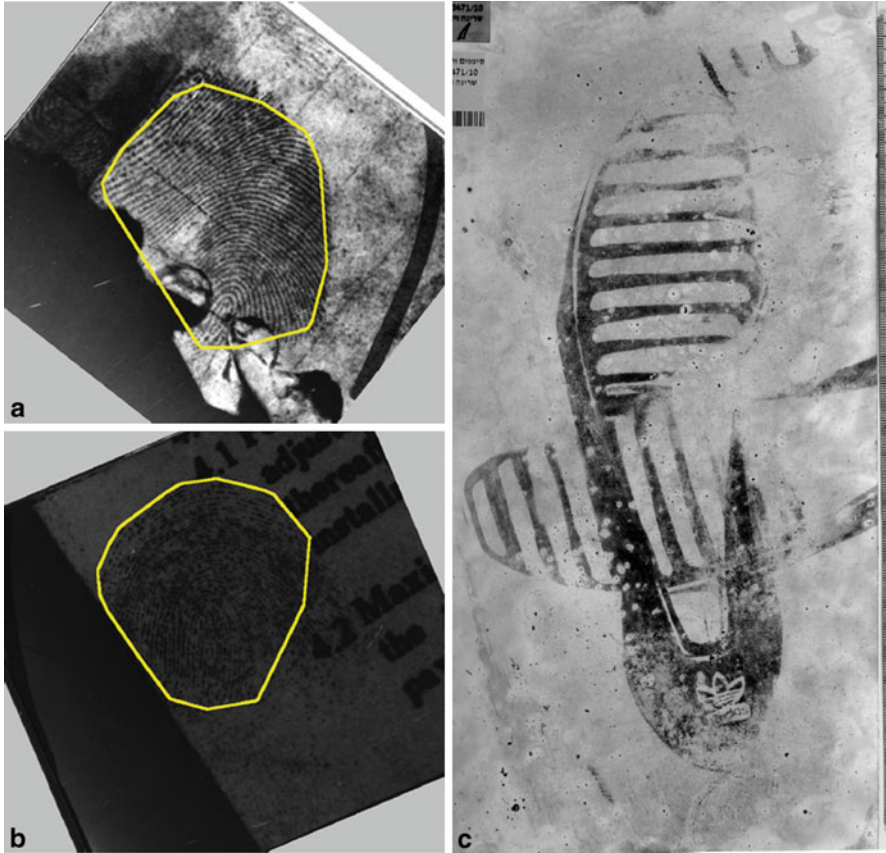


Fig. 1 Latent fingerprint images from the NIST special database 27 (left) (cf. Garris and McCabe (2000) with boundary (drawn in yellow) of the estimated region of interest by the DG3PD of Thai and Gottschlich (2016a) and from Wiesner et al. (2020b) two overlapping shoeprints with similar shoe pattern elements (right). A natural question is: Are those from the same shoe?

by *surveillance cameras* or by *web searches*, or structures of buildings in *satellite images*, to name just a few. In many applications, upon closer inspection, certain parts or features of these objects are of concern, e.g., texture information of dotting material in the *material sciences*, connectivity structure in *brain imaging*, or minutiae loci in fingerprints (see Fig. 2) for smartphone user *authentication and identification*. Extracting this kind of information out of often high-dimensional input images $F \in \mathbb{R}^{n \times m}$ is especially challenging when the inputs can consist of heterogeneous images, such as fingerprint images taken at a crime scene that are hard to model.

Notably, since all images are based on individual pixels, in this chapter, we consider only the discrete case, viewing images as matrices.

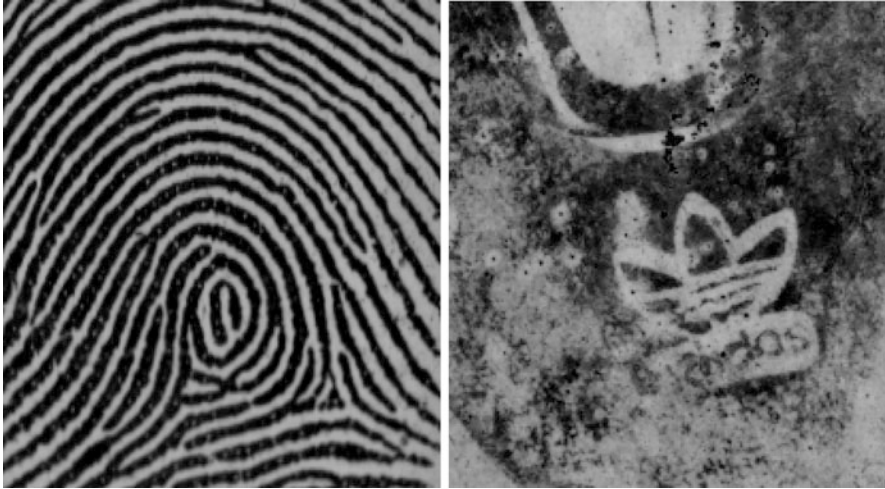


Fig. 2 Fingerprint ridge lines of very good quality following an *orientation field*, ending or forking at *minutiae* (left, from Turroni et al. 2011). Shoeprint (detail from Fig. 1) with sole *pattern* and pattern damages called *accidentals* (right). Here the black dots with circular white halo due to sand grains and the dark black clusters due to dirt need to be discriminated from true wear effects, for instance, on the left side of the brand's logo

These challenges have led to a surge of decomposition methods aimed at automatic removal of *noise* and/or *texture*, returning a piecewise constant or smooth *cartoon* component $U \in \mathbb{R}^{n \times m}$, possibly a second *texture* component $V \in \mathbb{R}^{n \times m}$, and a *noise* component $\varepsilon \in \mathbb{R}^{n \times m}$. Such decompositions can simplify the extraction of information as in applications one is often interested either in large-scale information (e.g., edges of buildings in aerial photographs) or small-scale information (e.g., fringe patterns in fingerprints). Well-known side effects of such methods are *loss of contrast* or artifacts such as *ringing* and *staircasing*.

In addition to image enhancement, decomposition of overlapping structures in the ROI is a frequent challenge in forensics; see Fig. 1. Moreover, structure at different scale is of high importance, e.g., shoe pattern *elements* identifying a shoe brand (cf. Fig. 1) and damages to the pattern due to wear or other damaging effects, called *accidentals*, identifying an individual shoe; cf. Fig. 2. In fingerprint analysis, the ridge line structure with its *orientation field* is the coarse structure to be identified, and *minutiae* (ending or forking ridge lines) convey the microstructure identifying individuals; cf. Fig. 2.

Automated fingerprint comparison utilizes minutiae loci and possibly the ridge line structure (orientation field); cf. (Maltoni et al. 2009). These are extracted by identifying a ROI. Bad quality images can be enhanced, or, while fingerprint scans are taken, bad quality scans can be rejected; cf. (NFI 2015; Yao et al. 2016; Richter et al. 2019).

For shoeprint analysis, due to the larger challenges given by the huge diversity of shoe element patterns and accidental structures, automated comparison is still in its very beginnings, e.g., Wiesner et al. (2020a,b).

Diffusion Methods

Solving the heat equation with initial conditions given by the image at hand, and following it over time, is one of the oldest smoothing methods. Over time, first smaller structures are smoothed, and then also bigger structures disappear, until, after infinite time, no information remains. This calls for smart choices of stopping times, and, in order to preserve specific structures for a longer time, alterations of the diffusion differential equation. For instance, Perona and Malik (1990), and subsequently Alvarez et al. (1992), impede diffusion along image gradients by *anisotropic nonlinear* diffusion, thus steering diffusion along rather constant image intensity regions.

In fingerprint images, among others, as detailed above, estimation of orientation fields is of high importance. Due to small interr ridge distances in fingerprints, in low-quality fingerprint images, however, image gradients are heavily influenced by noise and cannot be relied on. To this end, Perona (1998) applied orientation diffusion to estimate a smooth orientation field. Such separately estimated orientation fields (for alternate methods, e.g., Bazen and Gerez 2002) have been used by Gottschlich and Schönlieb (2012) for fingerprint enhancement (cf. Fig. 3 for this and related methods):

- (1) Orientation field (OF) estimation
- (2) *Oriented diffusion*
- (3) Contrast enhancement

An overview of structure tensor-based diffusion methods is given in Weickert (1998); for more broad structure-based image analysis with application in face and fingerprint recognition, see Bigun (2006).

Notably, solving the heat equation can be viewed as applying a low-pass Gauss filter, and anisotropic diffusion has been shown to be strongly connected to $TV-l^2$ minimization and Haar-wavelet soft-thresholding, e.g., Steidl et al. (2004), linking to spectral and wavelet methods briefly discussed in the next section and minimization methods in the next but one section.

Fourier and Wavelet Methods

In the context of image processing, Fourier, wavelet, curvelet, and similar transformations map an image from an image domain into a spectral, wavelet, etc. domain, apply some form of thresholding, and map the result under the inverse transformation back to the image domain, giving a *filtered* image. Such methods may serve all ends of noise removal, cartoon and texture identification, image

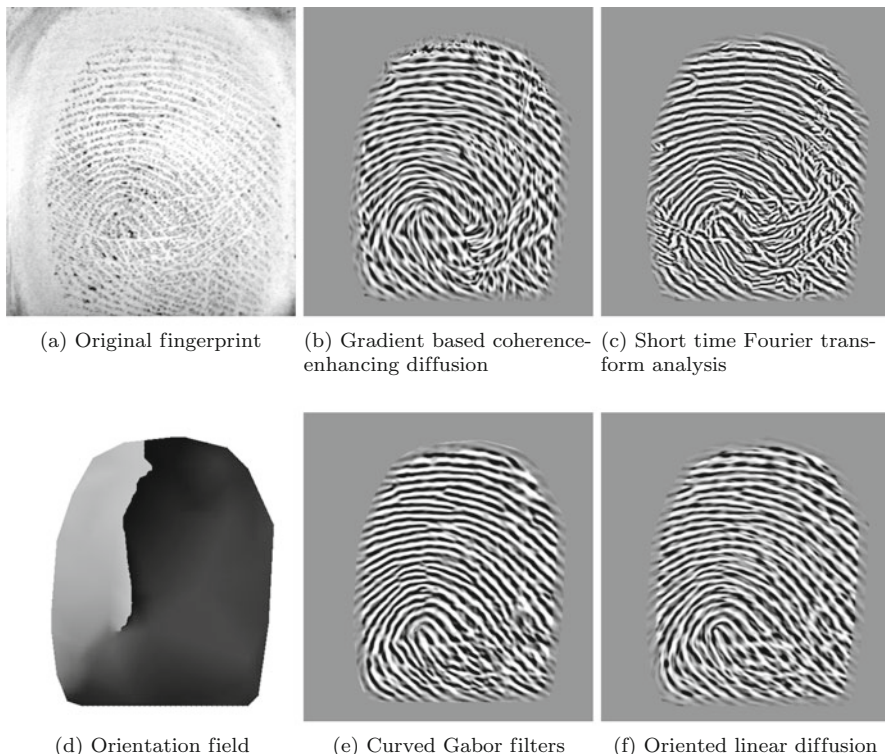


Fig. 3 A low-quality fingerprint (a) from Turrone et al. (2011) and the corresponding orientation field (d), where orientations in degrees are encoded as gray values between 0 and 179, with 0 denoting the x-axis' direction angle and angles increase clock-wise. Compared enhancement methods are (b) gradients-based coherence-enhancing diffusion filtering according to Weickert (1999), (c) STFT analysis by Chikkerur et al. (2007), (e) curved Gabor filters by Gottschlich (2012), and (f) oriented diffusion filtering by Gottschlich and Schönlieb (2012)

enhancement, and image compression. In extension of the Fourier transform, wavelet transforms also include local information, and thus draw strength from multiresolution analysis. Very popular is the Haar wavelet which is a special case of the Daubechies wavelet, for which many multiresolution filter banks are available. For an overview, cf. Daubechies (1992); Chui (1992); Mallat (2008). Curvelets have been introduced by Candès et al. (2006), and Ma and Plonka (2010) give a concise survey.

Particularly fingerprint images, due to their periodic fringe pattern, and also shoeprint images with repeating element patterns, are well suited to Fourier and wavelet methods. *Wavelet scalar quantization* (WSQ) has been used for fingerprint image compression by Hopper et al. (1993). Chikkerur et al. (2007) and Bartůněk et al. (2013) use the *short time Fourier transform* (STFT) for image enhancement; cf. Fig. 3. Gragnaniello et al. (2014) apply a three-level wavelet transform to fingerprint images from which in subsequent processing steps, features are derived

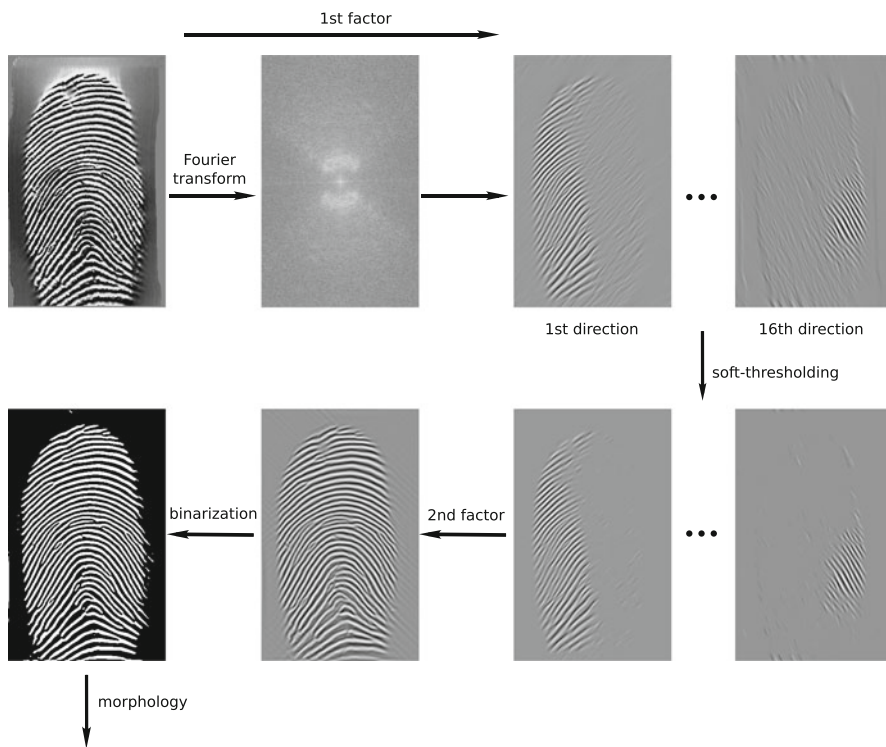


Fig. 4 Factorized directional bandpass (FDB) method from Thai et al. (2016): Soft-thresholding the result of 16 directional filters in the Fourier domain (first factor), binarizing the reconstructing (second factor) in the image domain, and morphological operations lead to identification of the ROI used in Fig. 5

to discriminate between real and *spoof* fingers. Spoof fingerprints are artificial fingerprints created from gelatin or latex; say, cf. Maltoni et al. (2009). *Factorized directional bandpass* (FDB) filters have been built by Thai et al. (2016) using the directional Hilbert transform of a Butterworth bandpass (DHBB) filter and soft-thresholding; cf. Fig. 4. Curiously, thresholding can be viewed as testing with statistical significance for the presence of non-zero filter response coefficients; cf. (Donoho and Johnstone 1994; Frick et al. 2012). The FDB filters have been optimized for texture extraction from fingerprint images with the purpose of segmentation; see Fig. 5.

Variational Problems

Variational problems have played an important role in imaging over the last decades; see, for example, Scherzer et al. (2009) and Aubert and Kornprobst (2006). As for anisotropic diffusion the emphasis lies on computing image approximations

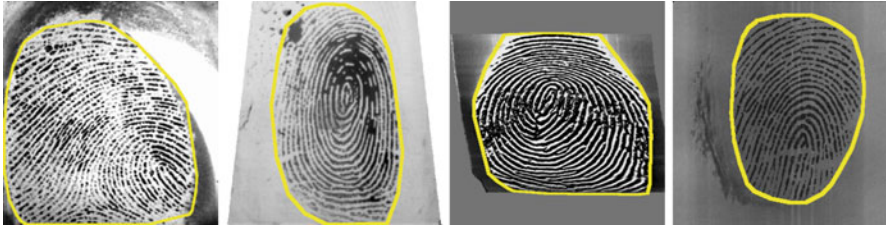


Fig. 5 Four examples of estimated fingerprint segmentation by FDB from Thai et al. (2016)

that keep sharp edges (discontinuities) while removing uninformative noise and/or texture. The, possibly, most influential model is the *Rudin-Osher-Fatemi* (ROF) model of Rudin et al. (1992). In its unconstrained form it is below given as Problem 1 (cf. Chambolle and Lions 1997; they use Neumann boundary conditions, however, as discussed below). It can be seen as a convex recast of the more intrigued *Mumford-Shah* (MS) model (Mumford and Shah 1989) that involves a non-convex term (Hausdorff measure of discontinuities); and for this the MS model is difficult to minimize numerically. Instead, the ROF model includes a total-variation term as regularizer that, when discretized, is given by an ℓ^1 -norm of the discrete gradient. Note that instead of the Neumann boundary conditions that are often enforced in the continuous domain (e.g., in the classical Rudin et al. 1992), we, being in the discrete domain, prefer periodic boundary conditions. Then, the discrete gradient comes in handy as a circular convolution operator denoted by $\underline{\mathcal{C}}_D$; cf. Definition 1 in section “Adaptive Balancing”.

Problem 1 (Discrete ROF Model). Let $F \in \mathbb{R}^{n \times m}$ be an input image and $\mu \in \mathbb{R}_+$. The (isotropic) $TV - \ell^2$ -model is given by

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}_{\text{ROF}}(U) := \left| \underline{\mathcal{C}}_D(U) \right|_{1,2} + \frac{\mu}{2} \|U - F\|^2, \\ \text{over} \quad & U \in \mathbb{R}^{n \times m}, \end{aligned} \tag{1}$$

where $|\cdot|_{1,2}$ is the ℓ^1 -norm of the ℓ^2 -norms of the gradients at each pixel and $\|\cdot\|$ is the usual Euclidean norm.

Solving (1) via steepest descent has led Andreu et al. (2001) to consider a corresponding partial differential equation (PDE) with weak solutions coined as *total variation (TV) flow*.

Meanwhile, alleviating for the systematic loss of contrast in the classical ROF-model, Osher et al. (2005) propose iterative Bregman iterations beginning with the ROF solution, passing near a putative noise free version and eventually converging in an *inverse scale-space flow* to the original noisy image.

An extension using higher order derivatives has led to the *total generalized variation (TGV)* model in Bredies et al. (2010) with more detail in Papafitsoros and

Bredies (2015). For a detailed overview of total variation in imaging, see Caselles et al. (2015) and the chapter in this book. In the context of relating different imaging techniques to one another, (Steidl et al. 2004), among others, link the balancing parameter μ of (1) to the stopping time in the anisotropic diffusion model discussed in section “Diffusion Methods”. In the following we also consider the general regularization problem given for an input image $F \in \mathbb{R}^{n \times m}$ by

$$\begin{aligned} & \text{minimize} && \mathcal{R}(U) + \mu \mathcal{D}(F, U), \\ & \text{over} && U \in \mathbb{R}^{n \times m}, \end{aligned} \tag{2}$$

with $\mathcal{R} : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ the *regularization term*, $\mathcal{D} : \mathbb{R}^{n \times m} \times \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ the *data-fidelity term*, and $\mu \in \mathbb{R}_+$ a *balancing parameter*.

Non-linear Spectral Decompositions

In analogy to filtering of (linearly) transformed coefficients – discussed in section “Fourier and Wavelet Methods” – a non-linear scale-space approach, the *TV transform*, has been developed by Gilboa (2014). The basis of the TV transform (the rescaled second derivative in the distributional sense of the TV flow) is the definition of so-called eigenfunctions for the TV flow – corresponding to functions f such that αf minimizes the continuous analog of (1) for some $\alpha \in \mathbb{R}$. The TV transform of a general image is then obtained by decomposing into such eigenfunctions (atoms), which for the TV flow are simply disks. Upon observing that the phenomenon of loss of contrast is rooted in the fact that no stopped TV flow is an ideal low-pass filter because the disks loose height, Gilboa (2014) proposes genuine low- and band-pass filters with respect to his TV transform.

The concept of the non-linear TV transform has been generalized to non-linear spectral decompositions for one-homogeneous functionals in Burger et al. (2016). To this end, the notion of the “eigenfunction” introduced above was extended to (2) with any one-homogeneous convex regularization term \mathcal{R} and \mathcal{D} being ℓ^2 -fidelity. It is not clear, however, whether finite linear combinations of eigenfunctions (also called singular vectors) are decomposable into their respective atoms; corresponding circumstances are addressed in Schmidt et al. (2018). Moreover, Burger et al. (2016) defined the non-linear spectral decomposition not only on the basis of the “forward” scale-space flow (as Gilboa 2014 with the TV flow) but also on the basis of the regularization model (2) – with ℓ^2 -fidelity – and the inverse scale-space flow, as introduced for the ROF model by Osher et al. (2005) and discussed above.

On the application side, the TV transform has been used for color image denoising in Moeller et al. (2015), texture decomposition into different scales in Horesh and Gilboa (2016), segmentation in Zeune et al. (2017), and image manipulation and image fusion in Hait and Gilboa (2018).

Texture Information

A typical example for texture is the fringe pattern in fingerprint applications. Decomposition methods as the ROF model into a cartoon U and a texture/noise part $V := U - F$ are well suited to obtain a binary ROI that segments an image into *foreground* (e.g., fingerprint) and *background*. Among the many descendants of the ROF model, there are the *decompositions into three parts*: cartoon, texture, and noise (e.g., Aujol and Chambolle 2005; Shen 2005), which are particularly useful in fingerprint analysis. One decisive step is introducing the theory of the G -space from Meyer (2001), a space particularly designed to feature small corresponding G -norms for oscillating functions. In general, for a function $f : \Omega \rightarrow \mathbb{R}$ from a bounded image domain Ω , the G -norm is given by

$$\|f\|_G := \inf\{\|g\|_{L^\infty(\Omega, \mathbb{R}^2)} : g = (g_1, g_2); f = \partial_1 g_1(x) + \partial_2 g_2\}, \quad (3)$$

where $\|g\|_{L^\infty(\Omega, \mathbb{R}^2)} = \text{ess sup}_{x \in \Omega} \left(x \mapsto \sqrt{g_1(x)^2 + g_2(x)^2} \right)$. Due to its indirect definition via the g 's, solving a minimization problem involving the G -norm is rather hard. There is quite a body of literature devoted to analyzing the G -space (and its related E - and F -space also introduced in Meyer 2001) and proposing approximations or simplifications to a $TV - G$ model (see, e.g., Vese and Osher 2003; Le and Vese 2005; Aujol et al. 2005). In section “[Adaptive Balancing](#)” a more detailed inspection of some of these approaches will be given.

For ROI extraction in fingerprint images, the *global three parts decomposition* (G3PD) model has been proposed by Thai and Gottschlich (2016b). It decomposes an image into cartoon, texture, and noise, using an anisotropic total variation regularizer for the cartoon U , a curvelet- ℓ^1 -norm plus an ℓ^1 -norm on the texture part V , while the ℓ^∞ -norm of the curvelet coefficients of $\varepsilon = F - U - V$ is bounded. The model involves the following objective function

$$\mathcal{J}_{\text{G3PD}}(U, V) := \left| \underline{\mathcal{C}}_D(U) \right|_{1,1} + \mu_1 |\mathcal{C}(V)|_1 + \mu_2 |V|_1,$$

for $U, V \in \mathbb{R}^{n \times m}$ and $\mathcal{C}(V)$ being the curvelet decomposition (Candès et al. 2006) of V , which is to be minimized under the constraints

$$|\mathcal{C}(\varepsilon)|_\infty \leq \delta, \quad F = U + V + \varepsilon,$$

where \mathcal{C} is the same curvelet transform of Candès et al. (2006) and $\mu_1, \mu_2, \delta \in \mathbb{R}$. Due to orientation sensitivity of the curvelet transform, G3PD is well suited to capture the fringe pattern of a fingerprint in the texture component; see Fig. 6. In automated practice, when applied to images, not containing other small-scale information featuring similar frequencies as the fingerprint pattern, parameters can be well tuned to specific sensors, such that ROIs are reliably extracted. On crime

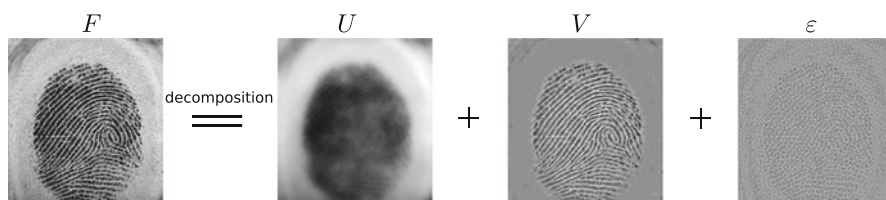


Fig. 6 Decomposition by G3PD of a fingerprint image F from Thai and Gottschlich (2016b) into three parts: cartoon (U), texture (V), noise (ε)

scene images, however, ideal parameter choices often vary substantially over images with different background, calling for more flexibility of the model and specific learning methods.

Machine Learning

We have thus far reported how for specific tasks at hand (e.g., segmentation, enhancement) specific tools have been designed, often using elaborate parameter tuning. In fact such ideal parameters often vary over varying use cases (e.g., G3PD requires different parameter choices when large regions contain small-scale patterns not related to the fingerprint at hand, as is often the case in real crime scene images). This calls for designing more flexible models and learning methods to incorporate heterogeneous use cases. Notably, when abundant data are available, nearly any machine learning method off the shelf usually works well. The less data are available, however, the more a priori structure must be built into learning methods. This is, for instance, the case in academic forensic research. For example, supervised learning models involving second order minutiae structure have resulted in a highly discriminatory test for separating real fingerprints from synthetic images where training set, validation set, and test set have summed up only 110 fingers (and 8 impressions per finger) per class; cf. Gottschlich and Huckemann (2014).

The very small size of data sets in fingerprint recognition and forensic applications stands in stark contrast to databases for image classification and visual object recognition like ImageNet which contains more than 14 million images (<http://image-net.org/about-stats>). Very large data sets enable fully automatic end-to-end learning by neural networks (Bengio 2009), whereas a very small number of training examples pose a huge additional machine learning challenge for biometric and forensic research. For the task of fingerprint quality estimation using image decomposition, Richter et al. (2019) proposed a new robust biometric quality validation scheme (RBQ VS) based on repeated random subsampling cross-validation to deal with problematic lack of a preferable number of training and test images. For fingerprint alteration detection, even fewer examples are available for training and testing and Gottschlich et al. (2015) also resort to cross-validation in order to compare different approaches. Biometric and forensic applications could profit

immensely from research on, e.g., deep learning (LeCun et al. 2015), evolutionary algorithms (Kennedy and Shi 2001), Bayesian learning (Neal 2012), support vector machines (Schoelkopf and Smola 2002), or random forests (Breiman 2001) if only larger data sets were available.

Regarding the aforementioned imaging approaches (cf. sections “[Diffusion Methods](#)”, “[Fourier and Wavelet Methods](#)”, “[Variational Problems](#)”), there have been a multitude of machine learning extensions proposed in the literature. For example, anisotropic diffusion has been learned by De los Reyes and Schönlieb (2013), and Chen and Pock (2017) have learned reaction diffusion models, while (Grossmann et al. 2020) learn TV transform filters. Arridge et al. (2019) give a survey on solving ill-posed inverse problems based on deep learning, with domain-specific knowledge contained in physical–analytical models.

Adaptive Balancing

Augmenting the ℓ^1 -regularization model in (1), obtaining a more general training model can be achieved by making the balancing parameter μ in the spatial or in the frequency domain adaptive. In fact, the former corresponds to bilevel minimization problems that choose the balancing parameter (or a more general balancing function) via its own minimization (e.g., Bredies et al. 2013; Calatroni et al. 2017), while the latter relates to various approaches to model texture following intuition from Meyer (2001). In the following we report on this connection for the discrete case and propose a way of extending the model class even beyond.

Definition 1. Define the *matrix-family convolution* in the following circular way by

$$\underline{\mathfrak{C}}_{\underline{B}}(U) := (B_1 * U, B_2 * U, \dots, B_p * U),$$

where $B * U$ is the usual circular convolution of matrices (e.g., Mallat 2008) with components given by

$$(B * U)[r, s] = \sum_{k=0}^{n-1} \sum_{\ell=0}^{m-1} B[k, \ell] U[r - k, s - \ell],$$

where k is taken modulo n and ℓ is taken modulo m . Moreover, let us denote by Γ_P the space of matrix-families $(\mathbb{R}^{n \times m})^P$.

Then the (forward) discrete gradient (with periodic boundary conditions) is given by the matrix-family convolution

$$\underline{\mathfrak{C}}_{\underline{D}} : \mathbb{R}^{n \times m} \rightarrow \Gamma_2, \quad U \mapsto (\mathfrak{C}_{D_1}(U), \mathfrak{C}_{D_2}(U))^T,$$

where

$$D_1[k, \ell] := \begin{cases} -1 & \text{if } [k, \ell] = [0, 0] \\ 1 & \text{if } [k, \ell] = [1, 0] \\ 0 & \text{else} \end{cases}, \quad D_2[k, \ell] := \begin{cases} -1 & \text{if } [k, \ell] = [0, 0] \\ 1 & \text{if } [k, \ell] = [0, 1] \\ 0 & \text{else} \end{cases}.$$

Adapting the Data-Fidelity-Norm

As mentioned in section “[Variational Problems](#)”, following Meyer (2001) there has been much research devoted to change the data-fidelity norm towards making it more adaptive to capture oscillating patterns. In the discrete setting, many of these models can be brought into the form of the general *TV-Hilbert model* proposed in Aujol and Gilboa (2006). Absorbing the balancing parameter in M , one can interpret the *TV-Hilbert model* as *adaptive TV-regularization* minimizing the functional

$$\mathcal{J}_M(U) := \left| \underline{\mathcal{C}}_D(U) \right|_{1,2} + \frac{1}{2} \left| \mathcal{C}_M(F - U) \right|^2, \tag{4}$$

with the discrete gradient $\underline{\mathcal{C}}_D$ and the new *balancing filter* $M \in \mathbb{R}^{n \times m}$ featuring $\widehat{M} \in \mathbb{R}_+^{n \times m}$ (where \widehat{M} is the discrete Fourier transform). Notably, Aujol and Gilboa (2006) also allow operators more general than the circular convolution operator \mathcal{C}_M above. Of course, the ROF model is a special case of (4) by choosing \mathcal{C}_M as a multiplication with the balancing parameter μ . Let us ponder first on a connection of (4) with the G -norm given in (3) and secondly on some literature considering (4).

Connection to the G -Norm

The *Osher-Solè-Vese (OSV) model* considered in Osher et al. (2003) serves as an example of the connection between the $TV - G$ model and minimizing the functional in (4). To sketch the underlying ideas, let u, f be functions defined on a bounded domain $\Omega \subset \mathbb{R}^2$. Recall that the G -norm of $f - u$ is defined as the infimum of L^∞ -norms over all $g \in L^\infty(\Omega, \mathbb{R}^2)$ such that $\text{div}(g) = f - u$. In Vese and Osher (2003) the G -norm is approximated by introducing g as a variable and replacing the G -norm of $f - u$ by an L^2 penalization $f - u - \text{div}(g)$ plus an L^p -norm on g with $1 \leq p < \infty$. Building on this model in Vese and Osher (2003), the OSV model in Osher et al. (2003) simplifies by assuming the existence of $g \in L^2(\Omega^2)$ with $\text{div}(g) = f - u$ and $g = \nabla P$ for some scalar-valued function $P \in H^1(\Omega)$. Hence,

$$f - u = \text{div}(g) = \Delta P.$$

Plugging the above into the model of Vese and Osher (2003), one obtains (cf. Equation (2.1) in Osher et al. 2003) for $\lambda \in \mathbb{R}_+$ the objective function

$$\int_{\Omega} |\nabla u| + \lambda \int_{\Omega} \left| \nabla (\Delta^{-1})(f - u) \right|^2.$$

Assuming that $\mathfrak{C}_{\tilde{M}}$ is an appropriate discrete version of the pseudo-differential operator Δ^{-1} , then discretizing the OSV model leads to the matrix-convolution operator $\underline{\mathfrak{C}}_D \mathfrak{C}_{\tilde{M}}$. Since $(\underline{\mathfrak{C}}_D \mathfrak{C}_{\tilde{M}})^* \underline{\mathfrak{C}}_D \mathfrak{C}_{\tilde{M}}$ is self-adjoint and thus has real and nonnegative eigenvalues, it allows for a unique positive-semidefinite square-root \mathfrak{C}_M which is the one from (4) and λ is set to 1, as it can be absorbed by M .

Other Choices of M

In Aujol et al. (2006), for a matrix-family convolution $\underline{\mathfrak{C}}_B$ with Gabor wavelet frames, *Gabor wavelet filters* of form $\mathfrak{C}_M = \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B$ have been proposed for (4).

Garnett et al. (2007) use Besov norms of the Besov spaces $\dot{B}_{p,q}^\alpha$ for $1 \leq p, 1 \leq q < \infty$ and $\alpha \in \mathbb{R}$ to approximate the G -norm. For $p = q = 2$ specific filters K_α (see Definition 5 of Garnett et al. 2007) are associated to the Besov spaces $\dot{B}_{2,2}^\alpha$. Then a discrete version of the Besov space norm on $F - U$ is given by

$$\left| \mathfrak{C}_{K_\alpha}(U - F) \right|^2,$$

cf. (56) in Garnett et al. (2007).

A model proposed by Buades et al. (2010) considers a special \mathfrak{C}_M in (4), defined in the frequency domain by the continuous filter

$$\widehat{L}_\sigma(\xi) := \frac{1}{1 + (2\pi\sigma|\xi|)^4}.$$

Discretization of the above then yields \mathfrak{C}_M .

Connections with Machine Learning

Yang et al. (2016) consider ℓ^1 -regularization in the spirit of machine learning approaches: They implement a general learning problem based on (2) with ℓ^2 -data-fidelity term. Upon closer inspection one can show that the learning architecture is constructed in such a way that it also learns over adaptive balancing parameters (Richter et al. 2020). In the remainder of this chapter we ponder on the connection of adaptive balancing and intersection point problems arising from an ADMM/AL algorithm solving (5) on which (Yang et al. 2016) build. To this end, denote the larger class of *adaptive ℓ^1 -regularizations* by

$$\mathcal{J}_{B,M}(U) := \left| \underline{\mathfrak{C}}_B(U) \right|_{1,\kappa} + \frac{\mu}{2} \left| \mathfrak{C}_M(F - U) \right|^2. \tag{5}$$

Here $\underline{B} \in \Gamma_P$ is a suitable matrix-family convolution, $\kappa \in \{1, 2\}$, $\mu \in \mathbb{R}_+$ and $M \in \mathbb{R}^{n \times m}$ is the balancing filter.

Solving via the ADMM/AL-Algorithm

The advantage of the functional $\mathcal{J}_{B,M}$ given in (5) lies in its convexity, in the smoothness of its data-fidelity term, and in the norm of its regularizer, being well understood. In the following we focus on the *alternating directions method of multipliers* (ADMM) in the context of *augmented Lagrangian* (AL) approaches. While the convergence of ADMM/AL to the exact solution is often slower when compared to other methods, its convergence to a neighborhood, when given bad starting values, is rather satisfactory. The method of multipliers has been introduced by Powell (1969) and Hestenes (1969). For a general result on the setup and convergence of ADMM/AL algorithms in the context of minimization via the augmented Lagrangian, see Theorem 8 of Eckstein and Bertsekas (1992) and references therein.

There have been various other algorithms proposed for minimizing functionals such as $\mathcal{J}_{B,M}$ from (5). The original ROF model, a special case of (4), was solved by Rudin et al. (1992) via a rather slow gradient descent algorithm. Popular later approaches include projection algorithms (Chambolle 2004; Aujol and Gilboa 2006), the use of Bregman distances (Goldstein and Osher 2009), graph-cut methods (Darbon and Sigelle 2006a,b), and forward-backward splitting (Chambolle and Pock 2011). For an in-depth overview, we refer to Chambolle and Pock (2016) and Goldstein et al. (2014).

The functional $\mathcal{J}_{B,M}$ of (5) contains the non-linear regularization term $|\underline{\mathcal{C}}_B(U)|_{1,\kappa}$ which cannot be minimized simply by differentiation with respect to U . For this reason a new additional variable \underline{W} is introduced, taking the place of $\underline{\mathcal{C}}_B(U)$. This yields the constrained problem

$$\begin{aligned} &\text{minimize } \tilde{\mathcal{J}}_{B,M}(U, \underline{W}) := |\underline{W}|_{1,\kappa} + \frac{\mu}{2} \left\| \underline{\mathcal{C}}_M(F - U) \right\|^2, \\ &\text{such that } \underline{\mathcal{C}}_B(U) = \underline{W}, \quad \text{over } U \in \mathbb{R}^{n \times m} \text{ and } \underline{W} \in \Gamma_P, \end{aligned} \tag{6}$$

which is equivalent to minimizing $\mathcal{J}_{B,M}$. Problem (6) can now be solved by computing the saddle point of the augmented Lagrangian functional \mathcal{J}_{AL} given below for $\beta \in \mathbb{R}_+$ and Lagrangian multiplier $\underline{\lambda} \in \Gamma_P$ (e.g., Bertsekas 1982),

$$\begin{aligned} \mathcal{J}_{AL}(U, \underline{W}, \underline{\lambda}) := & |\underline{W}|_{1,\kappa} + \frac{\mu}{2} \left\| \underline{\mathcal{C}}_M(F - U) \right\|^2 \\ & + \frac{\beta}{2} \left\| \underline{W} - \underline{\mathcal{C}}_B(U) \right\|^2 + \left\langle \underline{\lambda}, \underline{W} - \underline{\mathcal{C}}_B(U) \right\rangle. \end{aligned} \tag{7}$$

Algorithm 1 ADMM/AL for adaptive ℓ^1 -regularization (one step)**Input:** $F \in \mathbb{R}^{n \times m}$.**Input Filters:** $M \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$.**Customizable Parameters:** $\mu \in \mathbb{R}_+$, $\kappa \in \{1, 2\}$.**Initialization:** $U^{(0)} = F \in \mathbb{R}^{n \times m}$, $\underline{\lambda}^{(1)} = \underline{0} \in \Gamma_P$.**for** $\tau = 1, 2, \dots$ **do**

$$\underline{W}^{(\tau)} = \arg \min_{\underline{W} \in \Gamma_P} \left(\mathcal{J}_{\text{AL}} \left(U^{(\tau-1)}, \underline{W}; \underline{\lambda}^{(\tau)} \right) \right),$$

$$U^{(\tau)} = \arg \min_{U \in \mathbb{R}^{n \times m}} \left(\mathcal{J}_{\text{AL}} \left(U, \underline{W}^{(\tau)}; \underline{\lambda}^{(\tau)} \right) \right),$$

$$\underline{\lambda}^{(\tau+1)} = \underline{\lambda}^{(\tau)} + \beta \left(\underline{W}^{(\tau)} - \underline{\mathfrak{C}}_{\underline{B}} \left(U^{(\tau)} \right) \right).$$

end for

Notably, a saddle-point of (7) does not depend on the choice of β . To solve for the saddle-point, Algorithm 1 alternates between minimizing \mathcal{J}_{AL} for \underline{W} and U (one iteration of an ADMM algorithm), while updating in each iteration the Lagrangian multiplier $\underline{\lambda}$ via a gradient step.

Interpretation via a Feasibility Problem

We now show that Algorithm 1, which converges to the saddle-point of \mathcal{J}_{AL} , solves a special case of a broader feasibility problem (Problem 3). Before, we state a (seemingly) different feasibility problem directly derived from the above updating rules.

Problem 2. Given $F, M \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$, $\mu \in \mathbb{R}_+$, $\kappa \in \{1, 2\}$, and $\beta \in \mathbb{R}_+$, with discrete Fourier transform $\widehat{M} \in \mathbb{R}_+^{n \times m}$, find a point $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ in the intersection of the following three sets

$$\begin{aligned} \Omega_1^\kappa &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{W} = \underset{\underline{W} \in \Gamma_P}{\operatorname{argmin}} \mathcal{J}_{\text{AL}} \left(U, \underline{W}, \underline{\lambda} \right) \right\}, \\ \Omega_2 &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : U = \underset{\tilde{U} \in \mathbb{R}^{n \times m}}{\operatorname{argmin}} \mathcal{J}_{\text{AL}} \left(\tilde{U}, \underline{W}, \underline{\lambda} \right) \right\}, \\ \Omega_C &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{\mathfrak{C}}_{\underline{B}}(U) = \underline{W} \right\}. \end{aligned} \quad (8)$$

To prepare for the proof of equivalence of the above feasibility problem and the one stated further below (Problem 3), let us first compute the minimizers of \mathcal{J}_{AL}

with respect to \underline{W} and U , using standard variational calculus (from, e.g., Boyd and Vandenberghe 2004; Bauschke and Combettes 2011).

- For given $U \in \mathbb{R}^{n \times m}$, $\underline{B}, \underline{\lambda} \in \Gamma_P, \kappa \in \{1, 2\}$, as well as $\beta \in \mathbb{R}_+$, the unique minimizer of

$$\mathcal{J}_1(W) := |W|_{1,\kappa} + \frac{\beta}{2} \|W - \underline{\mathfrak{C}}_{\underline{B}}(U)\|^2 + \langle \underline{\lambda}, W \rangle,$$

is given by

$$\underline{W}^\dagger = \mathbf{S}_\kappa \left(\underline{\mathfrak{C}}_{\underline{B}}(U) - \frac{1}{\beta} \underline{\lambda}; \frac{1}{\beta} \right), \tag{9}$$

where $\mathbf{S}_\kappa : \Gamma_P \rightarrow \Gamma_P$ is the isotropic ($\kappa = 2$) or anisotropic ($\kappa = 1$) *soft-shrinkage function*.

- For given $F \in \mathbb{R}^{n \times m}$, $\underline{B}, \underline{W}, \underline{\lambda} \in \Gamma_P$, and $\beta \in \mathbb{R}_+$, the unique minimizer of

$$\mathcal{J}_2(U) := \frac{\mu}{2} \|\mathfrak{C}_M(F - U)\|^2 + \frac{\beta}{2} \|\underline{W} - \underline{\mathfrak{C}}_{\underline{B}}(U)\|^2 - \langle \underline{\lambda}, \underline{\mathfrak{C}}_{\underline{B}}(U) \rangle,$$

is given by

$$\begin{aligned} U^\dagger &= \mu \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}} \right)^{-1} \mathfrak{C}_M^* \mathfrak{C}_M(F) \\ &+ \beta \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}} \right)^{-1} \underline{\mathfrak{C}}_{\underline{B}}^* \left(\underline{W} + \frac{1}{\beta} \underline{\lambda} \right), \end{aligned} \tag{10}$$

given that $\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}}$ is invertible, which is the case because by $\widehat{M} \in \mathbb{R}_+^{n \times m}$ we have $\ker(\mathfrak{C}_M) = \{0\}$.

Abbreviating the two operators in (10), we introduce $A \in \mathbb{R}^{n \times m}$ and $\widetilde{\underline{B}} \in \Gamma_P$ such that

$$\mathfrak{C}_A := \mu \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}} \right)^{-1} \mathfrak{C}_M^* \mathfrak{C}_M, \tag{11}$$

and

$$\underline{\mathfrak{C}}_{\widetilde{\underline{B}}}^* := \beta \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}} \right)^{-1} \underline{\mathfrak{C}}_{\underline{B}}^*.$$

This gives the above anticipated feasibility problem. As before, for any matrix A, \widehat{A} denotes its discrete Fourier transform.

Problem 3. Consider arbitrary $F, Y \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$, $\kappa \in \{1, 2\}$, and $\nu \in \mathbb{R}_+$, such that the following two conditions are satisfied

1. $\widehat{Y} \in \mathbb{R}_+^{n \times m}$,
2. $\mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B$ has all eigenvalues in $[0, 1)$.

Moreover, define $\widetilde{B} \in \Gamma_P$ via

$$\widetilde{B}_p[k, \ell] = \widehat{Y}[k, \ell] \widehat{B}_p[k, \ell],$$

for all $0 \leq k \leq n-1$ and $0 \leq \ell \leq m-1$ and $1 \leq p \leq P$ and $A \in \mathbb{R}^{n \times m}$ as the matrix corresponding to the matrix-convolution given by

$$\mathfrak{C}_A = \mathfrak{E} - \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B,$$

where \mathfrak{E} is the identity operator on $\mathbb{R}^{n \times m}$. Find a point $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ in the intersection of the following three sets

$$\begin{aligned} \Omega_1^\kappa &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{W} = \mathfrak{S}_\kappa \left(\underline{\mathfrak{C}}_B(U) - \frac{1}{\nu} \underline{\lambda}; \frac{1}{\nu} \right) \right\}, \\ \Omega_2^G &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : U = \mathfrak{C}_A(F) + \underline{\mathfrak{C}}_B^* \left(\underline{W} + \frac{1}{\nu} \underline{\lambda} \right) \right\}, \\ \Omega_C &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{\mathfrak{C}}_B(U) = \underline{W} \right\}. \end{aligned} \quad (12)$$

Replacing M , we have introduced in Problem 3 a new matrix Y balancing now the interplay of the matrix-families \underline{B} and \widetilde{B} . It turns out that this balancing Y corresponds in the following way to the adaptive balancing filter M of (5), guaranteeing the equivalence of Problems 2 and 3.

Theorem 1. Let $F \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$ and $\kappa \in \{1, 2\}$. For given $M \in \mathbb{R}^{n \times m}$ such that $\widehat{M} \in \mathbb{R}_+^{n \times m}$, let $\mu \in \mathbb{R}_+$ and let $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ be a solution of Problem 2. Then, letting $\nu = \mu = \beta$, and defining $Y \in \mathbb{R}^{n \times m}$ via

$$\mathfrak{C}_Y = \beta \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right)^{-1},$$

we have that $\widehat{Y} \in \mathbb{R}_+^{n \times m}$, that $\mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B$ has all eigenvalues in $[0, 1)$, and that $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger)$ is a solution of Problem 3.

Vice versa, let $Y \in \mathbb{R}^{n \times m}$, such that $\widehat{Y} \in \mathbb{R}_+^{n \times m}$ and $\mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B$ has all eigenvalues in $[0, 1)$, let $\nu \in \mathbb{R}_+$, and let $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ be a solution of Problem 3. Then, defining $\mu = 1$ and \mathfrak{C}_M as the unique positive semi-definite square root of

$$\mathfrak{C}_{\tilde{M}} = \nu \mathfrak{C}_Y^{-1} \left(\mathfrak{C} - \mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} \right) = \nu \mathfrak{C}_Y^{-1} - \nu \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}}$$

(existing due to the eigenvalues of $\mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}}$ being strictly less than 1), then $\widehat{M} \in \mathbb{R}_+^{n \times m}$ and $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger)$ is a solution of Problem 2.

Proof. Let, as in the assertion, $F, M \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$, $\mu \in \mathbb{R}_+$ and $\kappa \in \{1, 2\}$ be given with $\widehat{M} \in \mathbb{R}_+^{n \times m}$, and let $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ be a solution of Problem 2. Recall that the solution does not depend on the choice of $\beta \in \mathbb{R}_+$ for \mathcal{J}_{AL} . Hence, w.l.o.g., we can set $\beta = \mu$. Moreover setting $\nu = \mu$ we have at once by (9) that the definitions of Ω_1^κ of Problem 2 and of Ω_1^κ Problem 3 coincide. Since Ω_C is the same for both problems, we are left to show that $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Omega_2^G$.

Defining $Y \in \mathbb{R}^{n \times m}$ as in the assertion, we have at once that $\widehat{Y} \in \mathbb{R}_+^{n \times m}$. Moreover, since matrix convolution operators are diagonalized by the discrete Fourier transform, the eigenvalues of $\mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}}$ are given by

$$\beta \left(\mu \widehat{M}[k, \ell]^2 + \beta \sum_{p=1}^P \left| \widehat{B}_p[k, \ell] \right|^2 \right)^{-1} \sum_{p=1}^P \left| \widehat{B}_p[k, \ell] \right|^2 \in [0, 1),$$

because $\widehat{M} \in \mathbb{R}_+^{n \times m}$.

Last, by (10) we have that

$$\begin{aligned} U^\dagger &= \mu \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} \right)^{-1} \mathfrak{C}_M^* \mathfrak{C}_M(F) \\ &\quad + \beta \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} \right)^{-1} \mathfrak{C}_{\underline{B}}^* \left(\underline{W}^\dagger + \frac{1}{\beta} \underline{\lambda}^\dagger \right) \\ &= \frac{\mu}{\beta} \mathfrak{C}_Y \mathfrak{C}_M^* \mathfrak{C}_M(F) + \mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \left(\underline{W}^\dagger + \frac{1}{\beta} \underline{\lambda}^\dagger \right) \\ &= \mathfrak{C}_A(F) + \mathfrak{C}_{\underline{B}}^* \left(\underline{W}^\dagger + \frac{1}{\nu} \underline{\lambda}^\dagger \right), \end{aligned}$$

where the last equality holds true due to $\nu = \mu = \beta$ and

$$\begin{aligned} \mathfrak{C}_A &= \mathfrak{C} - \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} = \mathfrak{C} - \mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} = \mathfrak{C}_Y (\mathfrak{C}_Y^{-1} - \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}}) \\ &= \mathfrak{C}_Y \left(\frac{\mu}{\beta} \mathfrak{C}_M^* \mathfrak{C}_M + \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} - \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}} \right) = \mathfrak{C}_Y \mathfrak{C}_M^* \mathfrak{C}_M. \end{aligned}$$

Hence, $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Omega_2^G$ yielding that $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger)$ is a solution of Problem 3.

Vice versa, let now $F, Y \in \mathbb{R}^{n \times m}$, $\underline{B} \in \Gamma_P$, $\nu \in \mathbb{R}_+$, and $\kappa \in \{1, 2\}$, with $\widehat{Y} \in \mathbb{R}_+^{n \times m}$, and suppose that $\mathfrak{C}_Y \mathfrak{C}_{\underline{B}}^* \mathfrak{C}_{\underline{B}}$ has all eigenvalues in $[0, 1)$. Further, let $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ be a solution of Problem 3.

Choose $\mu = 1$ and $M \in \mathbb{R}^{n \times m}$ as in the assertion. Since $\mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B$ has all eigenvalues in $[0, 1)$ and $\widehat{Y} \in \mathbb{R}_+^{n \times m}$, we have that $\mathfrak{C}_M^* \mathfrak{C}_M = \beta \mathfrak{C}_Y^{-1} (\mathfrak{C} - \mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B)$ has all eigenvalues in \mathbb{R}_+ . In consequence, by definition of the unique positive semi-definite square root, all eigenvalues of \mathfrak{C}_M are positive, yielding $\widehat{M} \in \mathbb{R}_+^{n \times m}$.

Next, let \mathcal{J}_{AL} be defined via $\beta = \nu$, then again via (9) the spaces Ω_1^κ and $\Omega_1^{\kappa'}$ defined in the two Problems 2 and 3 coincide, and the space Ω_C is the same anyway. Moreover, we have

$$\begin{aligned}
 U^\dagger &= \mathfrak{C}_A(F) + \underline{\mathfrak{C}}_B^* \left(\underline{W}^\dagger + \frac{1}{\nu} \underline{\lambda}^\dagger \right) \\
 &= (\mathfrak{C} - \mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B)(F) + \mathfrak{C}_Y \underline{\mathfrak{C}}_B^* \left(\underline{W}^\dagger + \frac{1}{\beta} \underline{\lambda}^\dagger \right) \\
 &= \left(\mathfrak{C} - \beta \left(\mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right)^{-1} \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right) (F) \\
 &\quad + \beta \left(\mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right)^{-1} \underline{\mathfrak{C}}_B^* \left(\underline{W}^\dagger + \frac{1}{\beta} \underline{\lambda}^\dagger \right) \\
 &= \mu \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right)^{-1} \mathfrak{C}_M^* \mathfrak{C}_M (F) \\
 &\quad + \beta \left(\mu \mathfrak{C}_M^* \mathfrak{C}_M + \beta \underline{\mathfrak{C}}_B^* \underline{\mathfrak{C}}_B \right)^{-1} \underline{\mathfrak{C}}_B^* \left(\underline{W}^\dagger + \frac{1}{\beta} \underline{\lambda}^\dagger \right).
 \end{aligned}$$

Hence $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger)$ is by (10) a minimizer of \mathcal{J}_{AL} for fixed \underline{W}^\dagger and $\underline{\lambda}^\dagger$ over $U \in \mathbb{R}^{n \times m}$, i.e., $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Omega_2$. Thus, $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger)$ solves Problem 2 for M and μ , as defined. \square

A General Learning Problem

The filter $Y \in \mathbb{R}^{n \times m}$ introduced in Problem 3 had to satisfy two properties. In order to generalize beyond these, we formalize them as relations between \underline{B} and $\widetilde{\underline{B}}$ and add already a relaxed version, which comes first.

Definition 2. Let $(\underline{B}, \widetilde{\underline{B}}) \in \Gamma_{2P}$.

- We say that $(\underline{B}, \widetilde{\underline{B}})$ factor weakly if for all $1 \leq p \leq P$ and $0 \leq k \leq n-1$ and $0 \leq \ell \leq m-1$ we have

$$\widehat{\widetilde{B}}_p[k, \ell] = \widehat{Y}_p[k, \ell] \widehat{B}_p[k, \ell],$$

for some $\underline{Y} = (Y_p)_{p=1}^P \in \Gamma_P$ with $\widehat{Y}_p \in \mathbb{R}_+^{n \times m}$ for all $1 \leq p \leq P$, called *factor matrix-family*.

- We say that $(\underline{B}, \widetilde{\underline{B}})$ factor strongly if for all $1 \leq p \leq P$ and $0 \leq k \leq n - 1$ and $0 \leq \ell \leq m - 1$ we have

$$\widehat{\underline{B}}_p[k, \ell] = \widehat{Y}[k, \ell] \widehat{\underline{B}}_p[k, \ell],$$

for some $Y \in \mathbb{R}^{n \times m}$ with $\widehat{Y} \in \mathbb{R}_+^{n \times m}$, called factor matrix.

- We say that $(\underline{B}, \widetilde{\underline{B}})$ satisfy the contraction and positive semidefinite condition (CPC) if

$$0 \leq \sum_{p=1}^P \overline{\widehat{\underline{B}}_p[k, \ell] \widehat{\underline{B}}_p[k, \ell]} < 1, \quad \text{for all } 0 \leq k \leq n - 1 \text{ and } 0 \leq \ell \leq m - 1.$$

Relaxing the feasibility Problem 3 from strongly factoring to weakly factoring, we obtain a more general problem. Moreover, we let the filter $A \in \mathbb{R}^{n \times m}$ be flexible as well.

Problem 4. Given $F \in \mathbb{R}^{n \times m}$, $\kappa \in \{1, 2\}$ and $\beta \in \mathbb{R}_+$, as well as input filters $(A, \underline{B}, \widetilde{\underline{B}}) \in \Gamma_{1+2P}$, find a point $(U^\dagger, \underline{W}^\dagger, \underline{\lambda}^\dagger) \in \Gamma_{1+2P}$ in the intersection of the following three sets

$$\begin{aligned} \Omega''_1^\kappa &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{W} = \mathbf{S}_\kappa \left(\underline{\mathfrak{C}}_{\underline{B}}(U) - \frac{1}{\beta} \underline{\lambda}; \frac{1}{\beta} \right) \right\}, \\ \Omega''_2^G &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : U = \mathfrak{C}_A(F) + \underline{\mathfrak{C}}_{\underline{B}}^* \left(\underline{W} + \frac{1}{\beta} \underline{\lambda} \right) \right\}, \\ \Omega_C &:= \left\{ (U, \underline{W}, \underline{\lambda}) \in \Gamma_{1+2P} : \underline{\mathfrak{C}}_{\underline{B}}(U) = \underline{W} \right\}. \end{aligned} \tag{13}$$

Generalizing in a similar manner Algorithm 1, we obtain the following Algorithm 2.

Notably, weakly factoring families allow for at least $(P - 1) \lceil \frac{mn}{2} \rceil$ new trainable parameters while keeping the eigenvalues of $\underline{\mathfrak{C}}_{\underline{B}}^* \underline{\mathfrak{C}}_{\underline{B}}$ real and positive. We have the following result on existence of a solution.

Theorem 2 (Richter 2019; Richter et al. 2020). Let $F \in \mathbb{R}^{n \times m}$, $\kappa \in \{1, 2\}$, $\beta \in \mathbb{R}_+$ and let $(A, \underline{B}, \widetilde{\underline{B}}) \in \Gamma_{1+2P}$ be input filters, with weakly factoring $(\underline{B}, \widetilde{\underline{B}})$ satisfying the (CPC). Then Problem 4 has a solution.

Uniqueness of the solution and convergence of Algorithm 2 to it, say, by showing that Algorithm 2 is again an ADMM/AL algorithm for Problem 4 remains an open problem. In practice, in all numerical experiments conducted by Richter (2019) and Richter et al. (2020) convergence has been observed.

Algorithm 2**Input:** $F \in \mathbb{R}^{n \times m}$.**Input Filters:** $(A, B, \tilde{B}) \in \Gamma_{1+2P}$.**Customizable Parameters:** $\beta \in \mathbb{R}_+, \kappa \in \{1, 2\}$.**Initialization:** $U^{(0)} = F \in \mathbb{R}^{n \times m}, \underline{\lambda}^{(1)} = \underline{0} \in \Gamma_P$.**for** $\tau = 1, 2, \dots$ **do**

$$\begin{aligned} \underline{W}^{(\tau)} &= \mathbf{S}_\kappa \left(\underline{\mathfrak{C}}_B \left(U^{(\tau-1)} \right) - \frac{1}{\beta} \underline{\lambda}^{(\tau)}; \frac{1}{\beta} \right), \\ U^{(\tau)} &= \mathfrak{C}_A(F) + \underline{\mathfrak{C}}_B^* \left(\underline{W}^{(\tau)} + \frac{1}{\beta} \underline{\lambda}^{(\tau)} \right), \\ \underline{\lambda}^{(\tau+1)} &= \underline{\lambda}^{(\tau)} + \beta \left(\underline{W}^{(\tau)} - \underline{\mathfrak{C}}_B(U^{(\tau)}) \right). \end{aligned}$$

end for**Filter Design Using Factor Families**

We conclude by reporting on weakly factoring filters proposed for Problem 4 in Richter (2019) and Richter et al. (2020). These filters lead to cartoon texture separation with desirable properties (keeping edges, removing texture and no blurring, caveat: mosaic pattern appearing); see Fig. 7. The construction is based heuristically on the filter A of the ROF model derived via (11) given by

$$\mathfrak{C}_A = \mu(\mu + \beta \underline{\mathfrak{C}}_D^* \underline{\mathfrak{C}}_D)^{-1}.$$

As $\underline{\mathfrak{C}}_D$ is a discrete gradient, the operator $\underline{\mathfrak{C}}_D^* \underline{\mathfrak{C}}_D$ is a discrete Laplace operator. The filter A can now be recast by the Laplacian B-spline ϕ defined in Van De Ville et al. (2005) given in the frequency domain by

$$\widehat{\phi}(x, y) := \left(\frac{4 \left(\sin^2 \left(\frac{x}{2} \right) + \sin^2 \left(\frac{y}{2} \right) \right) - \frac{8}{3} \left(\sin \left(\frac{x}{2} \right) \sin \left(\frac{y}{2} \right) \right)}{(x^2 + y^2)} \right)^{\frac{\kappa}{2}}. \quad (14)$$

In Van De Ville et al. (2005) the function ϕ served as a scaling function to construct bi-orthogonal wavelets. Doing a similar construction $(\underline{B}, \tilde{\underline{B}})$ can be obtained by a bi-orthogonal, directional wavelet frames construction (for the exact construction, see Richter et al. 2020, Appendix C of Richter 2019 and also Mallat 2008; Unser and Ville 2010). Note that if one were to use orthogonal wavelet frames we would be in the realm of strongly factoring, which is exactly the case for the Gabor wavelet frames proposed by Aujol and Gilboa (2006). The heuristic derivation of the new filter A , elaborated above, draws on a similar connection as in Cai et al. (2012), where the discrete gradient $\underline{\mathfrak{C}}_D$ is recast as a Haar wavelet frame, the first order cardinal B-spline (e.g., Chui 1992).

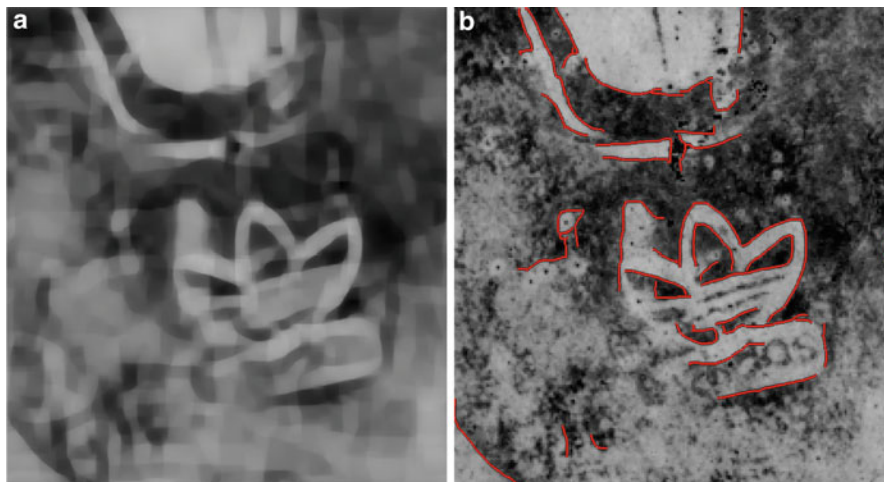


Fig. 7 Applying Algorithm 2 with filter families based on (14) in Problem 4 to the shoeprint detail (from Fig. 2), featuring sharp edges, little blurring, and minimal loss of contrast (left). From this cartoon picture, shoeprint elements are detected by a classical edge detection (Canny 1986) filter (right). For instance, the wear effect (called *accidental*, cf. section “Applications and Challenges for Automated Image Decomposition”) on the left of the brand’s logo is no longer part of the corresponding element’s edge

Conclusion

With advanced computational power and increased numbers of training images, learning methods have entered the field of image analysis and image decomposition. While in fingerprint recognition, automated methods have been around for decades, for forensics applications (latent shoeprint or fingerprint images of bad quality from crime scenes) such methods are far more difficult to design, due to the great heterogeneity of real life use case images. This calls for the development of

- (1) Highly flexible families of filters
- (2) Corresponding minimization/feasibility problems with solution guarantees
- (3) Corresponding algorithms with convergence guarantees

Additionally, since the use case is often defined only indirectly (e.g., improved quality results by improved matching rates, as in Richter et al. 2019), this calls for the development of

- (4) Learning methods based on objective functions, only indirectly available

In this chapter we have given a short survey on current research with emphasis on a recent development that seems promising in view of the above-stated goals (1)–(4).

Acknowledgments The authors thank the anonymous referee for the valuable comments and the first and last author gratefully acknowledge funding by the DFG within the RTG 2088.

References

- Nist fingerprint quality (NFIQ) (2015) <https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis>. Accessed: 2017-12-04
- Alvarez, L., Lions, P.-L., Morel, J.-M.: Image selective smoothing and edge detection by nonlinear diffusion. II. *SIAM J. Numer. Anal.* **29**(3), 845–866 (1992)
- Andreu, F., Ballester, C., Caselles, V., Mazón, J.M.: Minimizing total variation flow. *Differ. Integral Equ.* **14**(3), 321–360 (2001)
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numerica* **28**, 1–174 (2019)
- Aubert, G., Kornprobst, P.: Mathematical problems in image processing, volume 147 of *Applied Mathematical Sciences*. Springer, New York, 2nd edn. Partial differential equations and the calculus of variations, With a foreword by Olivier Faugeras (2006)
- Aujol, J.-F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vision* **22**(1), 71–88 (2005)
- Aujol, J.-F., Chambolle, A.: Dual norms and image decomposition models. *Int. J. Comput. Vis.* **63**(1), 85–104 (2005)
- Aujol, J.-F., Gilboa, G.: Constrained and SNR-based solutions for TV-Hilbert space image denoising. *J. Math. Imaging Vision* **26**(1–2), 217–237 (2006)
- Aujol, J.-F., Gilboa, G., Chan, T., Osher, S.: Structure-texture image decomposition—modeling, algorithms, and parameter selection. *Int. J. Comput. Vis.* **67**(1), 111–136 (2006)
- Bartůňek, J., Nilsson, M., Sällberg, B., Claesson, I.: Adaptive fingerprint image enhancement with emphasis on preprocessing of data. *IEEE Trans. Image Process.* **22**(2), 644–656 (2013)
- Bauschke, H.H., Combettes, P.L.: *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York. With a foreword by Hedy Attouch (2011)
- Bazen, A., Gerez, S.: Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 905–919 (2002)
- Bengio, Y.: Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2**(1), 1–127 (2009)
- Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Computer Science and Applied Mathematics. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York/London (1982)
- Bigun, J.: *Vision with Direction*. Springer, Berlin/Germany (2006)
- Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
- Bredies, K., Dong, Y., Hintermüller, M.: Spatially dependent regularization parameter selection in total generalized variation models for image restoration. *Int. J. Comput. Math.* **90**(1):109–123 (2013)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imag. Sci.* **3**(3), 492–526 (2010)
- Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
- Buades, A., Le, T., Morel, J.-M., Vese, L.: Fast cartoon + texture image filters. *IEEE Trans. Image Process.* **19**(8), 1978–1986 (2010)
- Burger, M., Gilboa, G., Moeller, M., Eckardt, L., Cremers, D.: Spectral decompositions using one-homogeneous functionals. *SIAM J. Imag. Sci.* **9**(3), 1374–1408 (2016)
- Cai, J.-F., Dong, B., Osher, S., Shen, Z.: Image restoration: total variation, wavelet frames, and beyond. *J. Am. Math. Soc.* **25**(4), 1033–1089 (2012)
- Calatroni, L., Cao, C., De Los Reyes, J.C., Schönlieb, C.-B., Valkonen, T.: Bilevel approaches for learning of variational imaging models. *Variational Meth Imaging Geometric Control* **18**(252), 2 (2017)

- Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. *Multiscale Model. Simul.* **5**(3), 861–899 (2006)
- Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(6), 679–698 (1986)
- Caselles, V., Chambolle, A., Novaga, M.: Total variation in imaging. In *Handbook of Mathematical Methods in Imaging*. Vol. 1, 2, 3. Springer, New York (2015), pp. 1455–1499
- Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vision* **20**(1–2), 89–97 (2004)
- Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numerische Mathematik* **76**(2), 167–188 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* **40**(1), 120–145 (2011)
- Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numerica* **25**, 161–319 (2016)
- Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1256–1272 (2017)
- Chikkerur, S., Cartwright, A., Govindaraju, V.: Fingerprint image enhancement using STFT analysis. *Pattern Recogn.* **40**(1), 198–211 (2007)
- Chui, C.K.: An introduction to Wavelets, Volume 1 of *Wavelet Analysis and its Applications*. Academic Press, Inc., Boston (1992)
- Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation. I. Fast and exact optimization. *J. Math. Imaging Vision* **26**(3), 261–276 (2006a)
- Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation. II. Levelable functions, convex priors and non-convex cases. *J. Math. Imaging Vision* **26**(3), 277–291 (2006b)
- Daubechies, I.: Ten lectures on wavelets, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1992)
- De los Reyes, J.C., Schönlieb, C.-B.: Image denoising: learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Probl. Imaging* **7**(4), 1183–1214 (2013)
- Donoho, D.L., Johnstone, J.M.: Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**(3), 425–455 (1994)
- Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **55**(3, Ser. A), 293–318 (1992)
- Frick, K., Marnitz, P., Munk, A., et al. Statistical multiresolution dantzig estimation in imaging: fundamental concepts and algorithmic framework. *Electron. J. Stat.* **6**, 231–268 (2012)
- Garnett, J.B., Le, T.M., Meyer, Y., Vese, L.A.: Image decompositions using bounded variation and generalized homogeneous Besov spaces. *Appl. Comput. Harmon. Anal.* **23**(1), 25–56 (2007)
- Garris, M.D., McCabe, R.M.: Nist special database 27: Fingerprint minutiae from latent and matching tenprint images. Technical Report 6534, National Institute of Standards and Technology, Gaithersburg (2000)
- Gilboa, G.: A total variation spectral framework for scale and texture analysis. *SIAM J. Imag. Sci.* **7**(4), 1937–1961 (2014)
- Goldstein, T., O’Donoghue, B., Setzer, S., Baraniuk, R.: Fast alternating direction optimization methods. *SIAM J. Imag. Sci.* **7**(3), 1588–1623 (2014)
- Goldstein, T., Osher, S.: The split Bregman method for L_1 -regularized problems. *SIAM J. Imag. Sci.* **2**(2), 323–343 (2009)
- Gottschlich, C.: Curved-region-based ridge frequency estimation and curved Gabor filters for fingerprint image enhancement. *IEEE Trans. Image Process.* **21**(4), 2220–2227 (2012)
- Gottschlich, C., Huckemann, S.: Separating the real from the synthetic: Minutiae histograms as fingerprints of fingerprints. *IET Biom.* **3**(4), 291–301 (2014)
- Gottschlich, C., Mikaelyan, A., Olsen, M., Bigun, J., Busch, C.: Improving fingerprint alteration detection. In: *Proceedings of 9th International Symposium on Image and Signal Processing and Analysis (ISPA 2015)*, pp. 83–86, Zagreb (2015)

- Gottschlich, C., Schönlieb, C.-B.: Oriented diffusion filtering for enhancing low-quality fingerprint images. *IET Biom.* **1**(2), 105–113 (2012)
- Gragnaniello, D., Poggi, G., Sansone, C., Verdoliva, L.: Wavelet-Markov local descriptor for detecting fake fingerprints. *Electron. Lett.* **50**(6), 439–441 (2014)
- Grossmann, T.G., Korolev, Y., Gilboa, G., Schönlieb, C.-B.: Deeply learned spectral total variation decomposition. arXiv preprint arXiv:2006.10004 (2020)
- Hait, E., Gilboa, G.: Spectral total-variation local scale signatures for image manipulation and fusion. *IEEE Trans. Image Process.* **28**(2), 880–895 (2018)
- Hestenes, M.R.: Multiplier and gradient methods. *J. Optim. Theory Appl.* **4**, 303–320 (1969)
- Hopper, T., Brislaw, C., Bradley, J.: WSQ gray-scale fingerprint image compression specification. Technical report, Federal Bureau of Investigation (1993)
- Horesh, D., Gilboa, G.: Separation surfaces in the spectral tv domain for texture decomposition. *IEEE Trans. Image Process.* **25**(9), 4260–4270 (2016)
- Kennedy, J.R.E., Shi, Y.: *Swarm Intelligence*. Academic, San Diego (2001)
- Le, T.M., Vese, L.A.: Image decomposition using total variation and div(BMO). *Multiscale Model. Simul.* **4**(2), 390–423 (2005)
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
- Ma, J., Plonka, G.: The curvelet transform. *IEEE Signal Process. Mag.* **27**(2), 118–133 (2010)
- Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic, San Diego (2008)
- Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: *Handbook of Fingerprint Recognition*. Springer, London (2009)
- Meyer, Y.: *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. American Mathematical Society, Boston (2001)
- Moeller, M., Diebold, J., Gilboa, G., Cremers, D.: Learning nonlinear spectral filters for color image reconstruction. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 289–297 (2015)
- Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
- Neal, R.M.: *Bayesian Learning for Neural Networks*, Vol. 118. Springer Science & Business Media (2012)
- Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.* **4**(2), 460–489 (2005)
- Osher, S., Solé, A., Vese, L.: Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Model. Simul.* **1**(3), 349–370 (2003)
- Papafitsoros, K., Bredies, K.: A study of the one dimensional total generalised variation regularization problem. *Inverse Prob. Imaging* **9**(2), 511 (2015)
- Perona, P.: Orientation diffusions. *IEEE Trans. Image Process.* **7**(3), 457–467 (1998)
- Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990)
- Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: *Optimization (Symposium, University of Keele, Keele, 1968)*, pp. 283–298. Academic, London (1969)
- Richter, R.: *Cartoon-Residual Image Decompositions with Application in Fingerprint Recognition*. Ph.D. thesis, Georg-August-University of Göttingen (2019)
- Richter, R., Gottschlich, C., Mentch, L., Thai, D., Huckemann, S.: Smudge noise for quality estimation of fingerprints and its validation. *IEEE Trans. Inf. Forensics Secur.* **14**(8), 1963–1974 (2019)
- Richter, R., Thai, D.H., Huckemann, S.: Generalized intersection algorithms with fixpoints for image decomposition learning. arXiv preprint arXiv:2010.08661 (2020)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: *Variational Methods in Imaging*. Springer (2009)
- Schmidt, M.F., Benning, M., Schönlieb, C.-B.: Inverse scale space decomposition. *Inverse Prob.* **34**(4), 1–34 (2018)

- Schoelkopf, B., Smola, A.: *Learning with Kernels*. MIT Press, Cambridge (2002)
- Shen, J.: Piecewise $H^{-1} + H^0 + H^1$ images and the Mumford-Shah-Sobolev model for segmented image decomposition. *AMRX Appl. Math. Res. Express* (4), 143–167 (2005)
- Steidl, G., Weickert, J., Brox, T., Mrázek, P., Welk, M.: On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numer. Anal.* **42**(2), 686–713 (2004)
- Strong, D., Chan, T.: Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Prob.* **19**(6), S165–S187 (2003). Special section on imaging
- Thai, D., Gottschlich, C.: Directional global three-part image decomposition. *EURASIP J. Image Video Process.* **2016**(12), 1–20 (2016a)
- Thai, D., Gottschlich, C.: Global variational method for fingerprint segmentation by three-part decomposition. *IET Biom.* **5**(2), 120–130 (2016b)
- Thai, D., Huckemann, S., Gottschlich, C.: Filter design and performance evaluation for fingerprint image segmentation. *PLoS ONE* **11**(5), e0154160 (2016)
- Turrone, F., Maltoni, D., Cappelli, R., Maio, D.: Improving fingerprint orientation extraction. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 1002–1013 (2011)
- Unser, M., Ville, D.V.D.: Wavelet steerability and the higher-order Riesz transform. *IEEE Trans. Image Process.* **19**(3), 636–652 (2010)
- Van De Ville, D., Blu, T., Unser, M.: Isotropic polyharmonic B-splines: scaling functions and wavelets. *IEEE Trans. Image Process.* **14**(11), 1798–1813 (2005)
- Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillatory patterns in image processing. *J. Sci. Comput.* **19**(1–3), 553–572 (2003)
- Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart (1998)
- Weickert, J.: Coherence-enhancing diffusion filtering. *Int. J. Comput. Vis.* **31**(2/3), 111–127 (1999)
- Wiesner, S., Kaplan-Damary, N., Eltzner, B., Huckemann, S.F.: Shoe prints: The path from practice to science. In: Banks, D., Kafadar, K., Kaye, D. (eds.) *Handbook of Forensic Statistics*, pp. 391–410. Springer (2020a)
- Wiesner, S., Shor, Y., Tsach, T., Kaplan-Damary, N., Yekutieli, Y.: Dataset of digitized racs and their rarity score analysis for strengthening shoeprint evidence. *J. Forensic Sci.* **65**(3), 762–774 (2020b)
- Wu, C., Tai, X.-C.: Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imag. Sci.* **3**(3), 300–339 (2010)
- Yang, Y., Sun, J., Li, H., Xu, Z.: Deep ADMM-net for compressive sensing MRI. In: 30th Conference on Neural Information Processing Systems (NIPS 2016), pp. 10–18 (2016)
- Yao, Z., Le Bars, J.-M., Charrier, C., Rosenberger, C.: A literature review of fingerprint quality assessment and its evaluation. *IET J. Biom.* **5**(3), 243–251 (2016)
- Zeune, L., van Dalum, G., Terstappen, L.W., van Gils, S.A., Brune, C.: Multiscale segmentation via bregman distances and nonlinear spectral analysis. *SIAM J. Imaging Sci.* **10**(1), 111–146 (2017)



Deep Learning Methods for Limited Data Problems in X-Ray Tomography

33

Johannes Schwab

Contents

Introduction	1184
Background	1185
Tomographic Image Reconstruction	1186
Deep Learning	1187
Case Examples in X-Ray CT	1190
Limited Angle Computed Tomography	1192
Reduction of Metal Artefacts	1196
Low-Dose Computed Tomography	1197
Further Methods	1198
Conclusion	1199
References	1200

Abstract

Successful medical diagnosis heavily relies on the reconstruction and analysis of images showing organs, bones, and other structures in the interior of the human body. In the last couple of years, the stored image data has increased tremendously, and also the computing power of modern GPUs experienced huge progress. Machine learning methods, and in particular deep learning methods, are on the rise to tackle advanced image reconstruction and image analysis tasks to support medical doctors in their diagnostic routines. In this chapter, we focus on the reconstruction task; especially consider tomographic imaging problems with incomplete, corrupted, or noisy data; and demonstrate how deep learning methods enable us to solve such tasks in a unified manner. We present the basic ideas of these methods assuming paired training data (supervised learning) and

J. Schwab (✉)

Department of Mathematics, University of Innsbruck, Innsbruck, Austria

e-mail: schwab@mrc-lmb.cam.ac.uk

utilizing only feed-forward networks. In particular, we illustrate the underlying concepts for missing data problems in classical computed tomography (CT), noting that most of the concepts can be transferred to other inverse imaging problems.

Keywords

Computed tomography · Deep learning · Inverse problem · Limited Data · Regularization

Introduction

Most modern medical imaging methods rely on the solution of an inverse problem, meaning that for given measured data $g \in \mathcal{Y}$ and physics-based forward model $\mathbf{R}: \mathcal{X} \rightarrow \mathcal{Y}$, the task is to estimate the cause $f \in \mathcal{X}$ for the observed measurements under the model \mathbf{R} . In an ideal setting, this amounts in solving the following task:

$$\text{Find } f \text{ from measurements } g = \mathbf{R}(f). \quad (1)$$

In tomographic imaging, the space \mathcal{X} is typically a space of functions $f: \Omega \rightarrow \mathbb{C}$, where the domain Ω denotes a subset of \mathbb{R}^2 (slice) or \mathbb{R}^3 . The corresponding model \mathbf{R} is an operator modelling the physical effects used for the tomographic modality. In computed tomography, \mathbf{R} describes the absorption of X-ray radiation in the investigated tissue (Hounsfield 1973), whereas in magnetic resonance imaging, \mathbf{R} describes the excitation and detection of radio-frequency signals of hydrogen atoms in the human tissue (Purcell et al. 1946). Tomographic imaging includes a great variety of applications in different fields, for example, electrical impedance tomography, optical tomography, positron emission tomography, seismic tomography, ultrasound tomography, and many more. In most of these applications, the forward model can be described by Radon type transforms, which use integrals over different families of one-dimensional manifolds. This can be integrals along lines as it is the case in X-ray transmission tomography (Natterer 2001), or integrals over circles in photoacoustic tomography (Beard 2011). In the following, we will present data-driven reconstruction methods based on three typical examples of ill-posedness in classical computed tomography. We assume that paired training data are available and, to make the article more readable, restrict ourselves to feed-forward neural networks. In principle, however, more complex network architectures, which are constantly being developed and improved, could be employed as well. Also the ideas illustrated on the example problems in this articles can be adopted to missing data problems in different inverse imaging applications.

Given the operator $\mathbf{R}: \mathcal{X} \rightarrow \mathcal{Y}$, in an ideal world, data would be given by $g = \mathbf{R}(f)$. However, in the real world, this is not the case, and $\mathbf{R}(f)$ is corrupted and modified by several sources. In this chapter, we consider and review three different

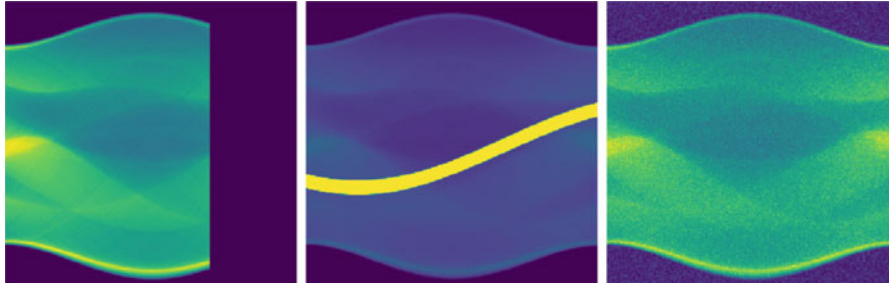


Fig. 1 Different sources of imperfect data in tomographic imaging. LEFT, incomplete data (e.g., limited angle CT); MIDDLE, corrupted data due to high-intensity region (e.g., a metal artifact); RIGHT, noisy data (e.g., low-dose CT)

frequently occurring problems in tomographic imaging, which can essentially be formulated as follows:

- Only **incomplete data** is available, meaning that only parts of the complete measurement data are given (Fig. 1).
- Partially **corrupted data** is measured. Here parts of the measurements are affected by physical effects not modelled by \mathbf{R} (Fig. 1).
- Presence of strong **noise in the data**. Physical measurements are inevitably affected by statistical uncertainty; therefore, the measured data cannot be fully described by the model \mathbf{R} (Fig. 1).

All of these scenarios typically lead to ill-posed inverse problems, where the reconstruction is either non-unique, the reconstruction process unstable, or the data not in the range of the operator \mathbf{R} . These issues can be analyzed by mathematical regularization theory (Engl et al. 1996). Incomplete data and partially corrupted data can lead to severe artifacts in the reconstruction. The noise in the data is propagating to the reconstructed image and can be severely amplified in the reconstruction process if the inverse of the forward operator is discontinuous. In all of these problems, exact direct reconstruction methods are either unavailable or lead to strong degradation of the reconstructed images. Iterative methods are extremely flexible and show good performance in all three cases, but come with very high computational cost. Deep Learning offers an alternative approach that can achieve good performance while being computationally efficient (Wang 2016).

Background

We begin with a brief description of the inverse problem in computed tomography and the three limited data problems mentioned earlier. Subsequently, we present the very basics of deep learning as well as a definition of feed-forward neural networks.

Tomographic Image Reconstruction

Analytic Reconstruction Methods

Common analytic reconstruction methods for tomographic imaging refer to numerical implementations of analytic inversion formulas and are of particular interest in application because they can be efficiently implemented. Most explicit inversion formulas for an operator \mathbf{R} are based on its adjoint operator \mathbf{R}^* or defined by some infinite series expansion. For many tomographic imaging problems, exact inversion formulas exist under the assumption that full, perfect data is available. Nevertheless, these inversion formulas only hold for specific scenarios. If the data is incomplete or not in the range of \mathbf{R} , the inversion formulas are not valid. As a consequence, the reconstructions are bad, if the data deviate from the mathematical model from which the inversion formula has been obtained. In addition, it is often challenging to incorporate existing prior knowledge into direct methods. To address these issues, iterative reconstruction methods can be used. As it turned out, deep learning constitutes a great opportunity to improve analytic image reconstruction methods (Fig. 2).

Iterative Reconstruction Methods

In contrast to direct methods, iterative methods rely on optimization tools for finding the minimizer f^* of a functional depending on the data g

$$f^* = \arg \min_{f \in \mathcal{X}} \|g - \mathbf{R}(f)\|_{\mathcal{Y}}^2.$$

Minimization problems of this type can be solved by various iterative methods. For example, assuming that \mathbf{R} is a linear operator between Hilbert spaces \mathcal{X} and \mathcal{Y} an iterative solution method is Landwebers algorithm (Landweber 1951). This algorithm is defined by the update formula

$$f_{k+1} = f_k + \mathbf{R}^*(g - \mathbf{R}(f_k)).$$

If g is in the domain of the Moore-Penrose inverse \mathbf{R}^+ defined by

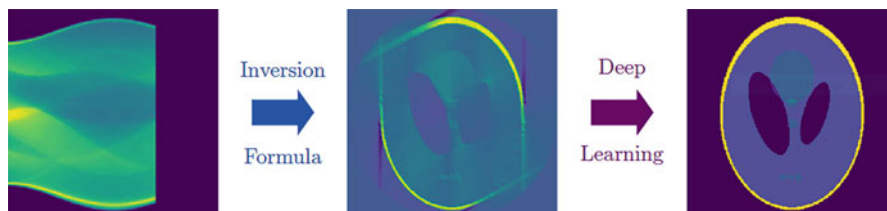


Fig. 2 Basic deep learning approach to improve analytic image reconstruction. First an analytic inversion method (derived for ideal data) is applied. In a second step, a deep learning algorithm is used to improve the initial reconstruction

$$\mathbf{R}^+ : \mathcal{Y} \supset \text{range}(\mathbf{R}) \oplus \text{range}(\mathbf{R})^\perp \rightarrow \mathcal{X}$$

$$y \mapsto \arg \min \{\|x\|_{\mathcal{X}} \mid x \in X \wedge \mathbf{R}^* \mathbf{R} x = \mathbf{R}^* y\}$$

the sequence f_k converges to the minimum norm least squares solution $\mathbf{R}^+(g)$ of the inverse problem (1) (Engl et al. 1996).

One advantage of indirect methods is that they are very flexible and one can easily add a penalty term $\mathcal{P} : \mathcal{X} \rightarrow [0, \infty]$ to obtain solutions that have specific characteristics (prior knowledge) by finding

$$f^* \in \arg \min_{f \in \mathcal{X}} \|g - \mathbf{R}(f)\|_{\mathcal{Y}}^2 + \mathcal{P}(f). \quad (2)$$

Such an approach for solving inverse problems is called variational regularization (Scherzer et al. 2009) or generalized Tikhonov regularization.

A popular choice of penalty term, also called regularizer, is the total variation (TV) of a function f , or some functional that enforces sparsity in a given basis or frame (Acar and Vogel 1994; Daubechies et al. 2004). Also penalty terms adapted to a data set of known solutions have been considered to describe signal characteristics for the class of desired solutions. For example, learning a basis or dictionary for signals to be recovered in which the reconstruction should have a sparse representation was proposed (Elad 2010). Further regularizers that are represented by deep neural networks have been proposed as well. A mathematical analysis of methods using learned regularizers has been developed in Lunz et al. (2018), Mukherjee et al. (2020), Li et al. (2020), and Obmann et al. (2020). Recently also, data-driven iterative algorithms serving to minimize (2) were introduced (Adler and Öktem 2017, 2018) and applied to various types of inverse problems (Wu et al. 2019; Guazzo 2020; Boink et al. 2019).

In the next subsection, we will provide a brief introduction to deep learning.

Deep Learning

In machine learning, the goal is to solve a given problem based on available observations. Analogous to physicists trying to explain the universe, for given observational data, one wants to find a model (or a theory in physics) that explains this data. But explaining data alone is not the most difficult challenge, since this can always be achieved with a model of sufficient complexity. For a good model, the real demand consists in enabling it to generate to new, unseen data and to make predictions. In recent years, a lot of research has been done on how such models can be calculated. An overview of common methods can be found, for example, in Goodfellow et al. (2016), Hastie et al. (2009), and LeCun et al. (2015).

Roughly speaking, machine learning tasks can be classified in Goodfellow et al. (2016).

- **Supervised learning:** Here the input to the task and the corresponding solution are known for the training set. Therefore, the training set consists of a subset of $\mathcal{A} \times \mathcal{B}$ where the training pairs are coupled by the problem to be solved.
- **Unsupervised learning:** No paired data set is available, and the training set only consists of the inputs; the solutions or even the concrete task is unknown. The training set consist of a subset of \mathcal{A} assumed to have some particular property which is to be discovered.

In this chapter, we exclusively focus on supervised learning tasks, which are described in the following. A model for solving a problem can be interpreted as an operator $\Phi: \mathcal{A} \rightarrow \mathcal{B}$. If the given data consists of input instances $(a_i)_{i=1}^N$ and the corresponding solutions $(b_i)_{i=1}^N$ fitting the data means finding an operator

$$\Phi^* = \arg \min_{\Phi: \mathcal{A} \rightarrow \mathcal{B}} \frac{1}{N} \sum_{i=1}^N \mathcal{D}(\Phi(a_i), b_i),$$

where \mathcal{D} is some similarity measure in the space \mathcal{B} . However, a model Φ^* also has to predict meaningful solutions for data different from the data used for the fitting. A model, which is unable to make predictions, is more or less useless. To achieve this, the class of admissible operators is restricted to a subset \mathcal{C} of all mappings $\Phi: \mathcal{A} \rightarrow \mathcal{B}$. In practice, additional strategies are adopted in the optimization procedure in order to restrict the class of possible solution operators.

The ultimate goal for the application is to implement a computer program, which finds a good approximation of the operator that is able to solve some specific task, as, for example, image analysis and image reconstruction tasks in medical imaging (Wang 2016). This model optimization is also termed learning of the model. For this purpose, the user has to feed the computer with experience, called training data, for example, images or measurement data.

We now introduce a popular approach of setting up such a task-solving machinery for supervised learning problems. The approach consists in parametrizing the function, which maps a given input to the solution of the problem. A particular class of such functions is called artificial neural networks (Werbos 1974).

After discretization of the spaces $\mathcal{A} := \mathbb{R}^L$ and $\mathcal{B} := \mathbb{R}^M$, a feed-forward artificial neural network is given by

$$\begin{aligned} \Phi_{\mathcal{W}}: \mathbb{R}^L &\rightarrow \mathbb{R}^M \\ a &\mapsto \Phi_{\mathcal{W}}(a) := (\sigma_K \circ \mathbf{W}_K \circ \sigma_{K-1} \circ \mathbf{W}_{K-1} \circ \dots \circ \sigma_1 \circ \mathbf{W}_1)(a), \end{aligned} \tag{3}$$

where $\mathbf{W}_i: \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_{i+1}}$ and $\sigma_i: \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$ for $i \in \{1, \dots, K\}$ are affine linear operators and point-wise nonlinear mappings, respectively. Further \mathcal{W} denotes the dependence of the function $\Phi_{\mathcal{W}}$ on the operators \mathbf{W}_i . We consider the real vector spaces $\mathbb{R}^{n_1} = \mathbb{R}^L$ and $\mathbb{R}^{n_K} = \mathbb{R}^M$ as input and output spaces. Networks of the form (3) are called feed-forward networks as they have a sequential, forward directed

structure. We note that a great variety of more complex network architectures exist that, for example, also allow cycles or loops. In all of what follows, the network architecture is not essential, and everything can equally be applied to more sophisticated network designs. In a feed-forward neural network, fixing the depth K , the dimension of the intermediate spaces \mathbb{R}^{n_i} , and the functions σ_i gives a class of operators only depending on the parameters of the affine linear functions \mathbf{W}_i . These parameters are the entries of the matrices and are called weights of the artificial neural network. A particular choice of the linear operators is discrete convolution operators. One of the main advantages of convolutions is that the corresponding matrices only contain a small number of nonzero weights, which is computationally much more efficient than using full matrices (fully connected layers). Networks consisting of such discrete convolutions are called convolutional neural networks and are of particular interest for imaging tasks since they are able to detect local correlations. Further if K is not very small, a neural network is called deep, although there is no strict definition of when a network is considered to be deep. A typical choice for the nonlinear mappings σ_i is the rectified linear unit (ReLU)

$$\sigma(x) = \text{ReLU}(x) := \max\{0, x\},$$

or sigmoid functions.

Given a set of training data $(a_i, b_i)_{i=1}^N$, the goal now is to find good linear operators, such that the neural network fits the training data and is able to generalize the learned expertise. If we denote the vector of weights by $\mathcal{W} := (\mathbf{W}_1, \dots, \mathbf{W}_N)$, the corresponding minimization problem can be formulated by

$$\text{Find } \Phi_{\mathcal{W}} \text{ minimizing } \mathcal{L}(\mathcal{W}) := \frac{1}{N} \sum_{i=1}^N \mathbf{D}(\Phi_{\mathcal{W}}(a_i), b_i), \quad (4)$$

where $\mathbf{D}: \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, \infty)$ is some distance measure on the output space and \mathcal{L} represents the cost function. Assuming that \mathcal{L} admits calculation of a (sub-)gradient, minimization of \mathcal{L} is typically done by gradient descent methods. In these procedures, the parameters are iteratively updated by

$$\mathcal{W} \rightarrow \mathcal{W} - \eta \nabla \mathcal{L}(\mathcal{W}),$$

where η is a parameter determining the step size, also called learning rate in machine learning. In practice a much cheaper alternative is deployed which only takes into account the gradient of the cost function \mathcal{L} corresponding to a subset of the training data. Typically these subsets of training instances are randomly selected, resulting in stochastic gradient descent methods. The partial derivatives of the gradient are computed by the backpropagation algorithm (Hecht-Nielsen 1992; Higham and Higham 2019).

Optimization of (4) is challenging, since the cost function is non-convex. Various techniques to an improvement of this optimization process as well as the

generalization properties of an artificial neural network have been proposed. These techniques include:

- Evaluating the model with a data set not contained in the training set during training to estimate the generalization capability of the network; this set is typically called **validation set**.
- Including other operations (layers) in the network architecture; some examples are **pooling layers**, which reduce the dimension by taking the maximum or average over a small region of an intermediate output. Further possibilities to improve generalization properties and optimization are **dropout layers** and **batch normalization layers** and also **residual connections** and other skip connections that add or concatenate outputs obtained earlier in the network to inputs in later stages. Detailed explanation of these building blocks can be found in Goodfellow et al. (2016) and Lundervold and Lundervold (2019).
- More sophisticated variants of gradient descent algorithms including **momentum** or **Nesterov updates**; a summary and explanation of popular optimization algorithms are given in Ruder (2016).
- Including a penalty term \mathcal{P} for the weights in the cost function and minimizing

$$\mathcal{L}(W) = \frac{1}{N} \sum_{i=1}^N \mathbf{D}(\Phi_W(a_i), b_i) + \mathcal{P}(W).$$

The choice of the particular network and optimizer is very important for obtaining the best possible results and depends on the specific task to be solved. Likewise, the choice of the loss function \mathbf{D} plays an important role to obtain a valuable model. Depending on the specific task, a huge amount of different loss functions have been proposed, ℓ^2 , ℓ^1 , structural similarity (SSIM) and Wasserstein distance being the most popular, when working with images. In the following, however, we concentrate on illustrating the conceivable application of neural networks rather than on the concrete network design and optimization strategy.

Case Examples in X-Ray CT

To illustrate deep learning methods for tomographic image reconstruction, in the following, we consider the parallel beam geometry for X-ray computed tomography. In this imaging method, the particular mathematical model \mathbf{R} is the Radon transform, which evaluates the integrals over all lines across the radiative absorption coefficient of the tissue. In this case, the sought-after function f is the spatially depending absorption coefficient, and the measured data follows the physical model.

$$g(\theta, s) = \mathbf{R}f(\theta, s) = \int_{L(\theta, s)} f(x) \, d\sigma(x). \quad (5)$$

Here \mathbb{S}^{d-1} denotes the $(d - 1)$ -dimensional unit sphere, which declares the direction of the line, $s \in \mathbb{R}$ determines the distance of the line to the origin, and σ denotes the surface measure on $L(\theta, s)$. The data consists of all line integrals along lines $L(\theta, s) := s\theta^\perp + \mathbb{R}\theta$ where $(\theta, s) \in \mathbb{S}^{d-1} \times \mathbb{R}$ (Fig. 3). This operator \mathbf{R} is called Radon transform, and in theory exact inversion of the transform is possible. An extensive overview of the mathematical formulation of X-ray tomography and solution methods can be found, among others, in Natterer (2001), Deans (2007), and Scherzer et al. (2009).

In the following, we will shortly describe common reconstruction methods for X-ray computed tomography.

Analytic Reconstruction

For the Radon transform (5) for $d = 2$, such an exact reconstruction formula (Natterer 2001) is given by

$$f(x) = \frac{1}{4\pi^2} \int_{\mathbb{S}^1} \left(\int_{\mathbb{R}} \frac{\partial_s \mathbf{R}f(\theta, s)}{\theta \cdot x - s} ds \right) d\theta, \tag{6}$$

where ∂_s denotes the partial derivative in the s component and \cdot the standard inner product in \mathbb{R}^2 . Using the Hilbert transform

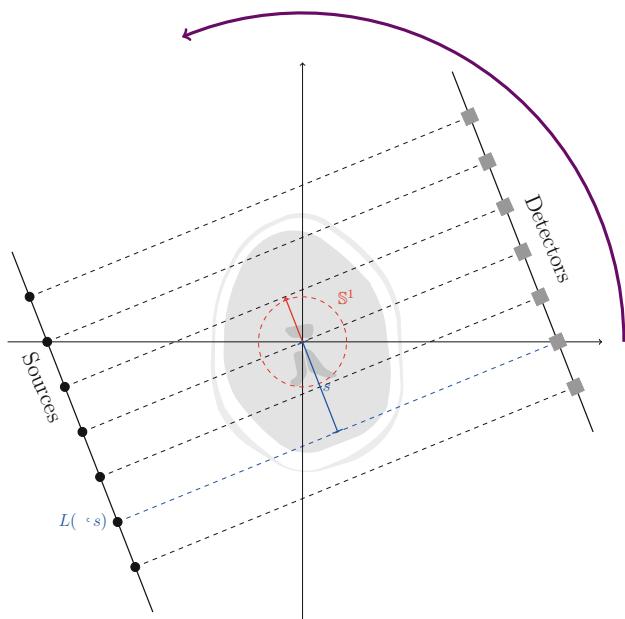


Fig. 3 Illustration of parallel beam CT. The sources and detectors rotate around the object. The vector $\theta \in \mathbb{S}^1$ determines the angle of the parallel lines and the scalar $s \in \mathbb{R}$ the distance of the line to the origin

$$H_s g(\theta, s) = \frac{1}{\pi} \int_{\mathbb{R}} \frac{g(\theta, t)}{s - t} dt$$

the inversion formula above can be written as

$$f(x) = \frac{1}{4\pi} \mathbf{R}^* (H_s \partial_s \mathbf{R}(f))(x), \tag{7}$$

where \mathbf{R}^* denotes the adjoint of the Radon transform defined by

$$\mathbf{R}^* g(x) = \int_{\mathbb{S}^1} g(\theta, \theta \cdot x) d\theta.$$

Here the improper integral arising in the Hilbert transform is understood in the sense of Cauchy. The operation $H_s \partial_s$ is called filtering, whereas the adjoint of the Radon transform is also called backprojection operator. Such inversion formulas of filtered backprojection type are available for different variants of the Radon transform as well, occurring in various tomographic imaging problems.

Analytic reconstruction in practice then consists of implementing a discretized version of the inversion formula. The inversion formula (7) can, for example, be implemented by:

- Approximation of the filtering operation $H_s \partial_s$ by discrete convolution in Fourier domain by a non-singular filter (e.g., Ram-Lak filter, Shepp-Logan filter)
- Interpolation to compute the values of the filtered data at the points $(\theta, \theta \cdot x)$.
- Numerical integration methods to compute the integral over \mathbb{S}^1 (backprojection).

Limited Angle Computed Tomography

For some applications, it is favorable to only measure line integrals along a limited range of angles, to reduce scanning time or being able to reduce the scanning area to a smaller region. In some applications, it is even impossible to measure all line integrals around the object under investigation due to physical constraints. Therefore, the data is limited to certain areas which make high-quality reconstructions with simple inversion formulas an unsolvable task. Recently, however, deep learning algorithms have made a huge advance that has made it possible to get a good reconstruction despite the limited data. In the case of limited angle computed tomography, we consider subsets of the form $\Lambda := \Gamma \times \mathbb{R} \subset \mathbb{S}^1 \times \mathbb{R}$, where Γ is a connected subset of \mathbb{S}^1 denoting the set of directions in which measurements are available. The set Γ corresponds to the range of measurement angles not covering the full range of 180° required for exact reconstruction. Thus, for example, the set covering only 90° is given by $\Gamma := \{(\sin(\alpha), \cos(\alpha)) \mid \alpha \in [0, \pi/2]\} \subset \mathbb{S}^1$. The restriction of the data can be formulated by $\chi_\Lambda g$, where

$$\chi_{\Lambda}(\theta, s) = \begin{cases} 1 & \theta \in \Gamma \\ 0 & \theta \notin \Gamma. \end{cases}$$

A reconstruction can then be found by finding a solution f minimizing the penalty term \mathcal{P} and matching the available data

$$f^* = \arg \min_{f \in \mathcal{X}} \|\chi_{\Lambda} g - \chi_{\Lambda} \mathbf{R}(f)\|_{\mathcal{Y}} + \mathcal{P}(f). \quad (8)$$

Iterative reconstruction methods employing a penalty term of the image gradient (\mathcal{P} total variation functional) yield satisfying results, but are computationally expensive. Therefore, deep learning methods that can be trained prior to reconstruction are a good option, as they can be employed very quickly to make predictions once their training process is complete.

The two most prominent deep learning methods for improving limited angle tomography can be assigned to two classes: methods that work in data domain and approaches that already use some initial reconstruction.

Learning in Data Domain

Deep learning methods working in data domain do not only aim for minimizing the data fidelity on the restricted data $\|\chi_{\Lambda} g - \chi_{\Lambda} \mathbf{R}(f)\|_{\mathcal{Y}}$ but also for finding an extension of the data to the set $\Lambda^c := (\mathbb{S}^1 \setminus \Gamma) \times \mathbb{R}$.

Given a set of N pairs of data $(\chi_{\Lambda} g_i, g_i)_{i=1}^N \subset \mathcal{Y} \times \mathcal{Y}$, the goal is to find some data extension operator $\Phi: \mathcal{Y} \rightarrow \mathcal{Y}$ in a certain operator class \mathcal{C} that maps $\chi_{\Lambda} g_i$ to g_i for every training sample. For natural images, this task would consist of image completion. This can be formulated by finding the operator

$$\Phi^* := \arg \min_{\Phi \in \mathcal{C}} \frac{1}{N} \sum_{i=1}^N \|\Phi(\chi_{\Lambda} g_i) - g_i\|_{\mathcal{Y}}, \quad (9)$$

where the norm $\|\cdot\|_{\mathcal{Y}}$ can be replaced by any distance measure $\mathcal{D}_{\mathcal{Y}}$ on \mathcal{Y} . Subsequently, given this extension operator Φ^* , one can obtain a reconstruction either by solving

$$f^* = \arg \min_{f \in \mathcal{X}} \|\Phi^*(\chi_{\Lambda} g) - \mathbf{R}f\|_{\mathcal{Y}} + \mathcal{P}(f),$$

or using a direct reconstruction method for the full operator \mathbf{R} . Many algorithms have been proposed to extend the data to the full range of 180° , as, for example, sinogram restoration based on Helgason-Ludwig consistency conditions (Huang et al. 2017) and other consistency conditions. A popular approach is to approximate the extension operator by a fully convolutional network or a generative adversarial network. In Anirudh et al. (2018), the authors propose a 1D convolutional network

to generate a latent code from the partial sinogram. Subsequently this latent code is fed to a 2D convolutional generator network, which is optimized together with a discriminator network, rating the generated image. Applying the full-view Radon transform to the generated image yields the projection data for all angles, and the missing values in the original sinogram are replaced by the new data. Typically one wants the extension operator Φ^* to be consistent with the given data, meaning that it does not change the available measured values of the data in Γ .

Learning in Image Domain

A second approach consists in using the limited data for an initial reconstruction (8) which is then refined by a learned operator. Artifacts occurring in limited angle computed tomography have been studied and characterized (Quinto 1988; Frikel and Quinto 2013) for a long time. Since these artefacts are deterministic and have a directional property, deep convolutional networks, which have proven very successful in detecting signal features and patterns, also seem to be suitable for removing limited angle artifacts.

Given a set of N functions in the manifold of desired solutions $\mathcal{M} \subset \mathcal{X}$, one can obtain pairs $(f_i, g_i)_{i=1}^N \subset \mathcal{M} \times \mathbf{R}(\mathcal{M})$ by computing the Radon transform and generate a set of training data $(f_i, \chi_\Lambda g_i)$ by restricting the Radon data to the set Λ . The goal in this approach is to find some operator that removes artefacts in the reconstructions f_i^* obtained by some iterative reconstruction algorithm for finding

$$f_i^* = \arg \min_{f \in \mathcal{X}} \|\chi_\Lambda g_i - \chi_\Lambda \mathbf{R}(f)\|_Y + \mathcal{P}(f).$$

or by some direct reconstruction algorithm (e.g., (6)). The refinement operator can then be calculated by

$$\Phi^* := \arg \min_{\Phi \in \mathcal{C}} \frac{1}{N} \sum_{i=1}^N \mathcal{D}_{\mathcal{X}}(\Phi(f_i^*), f_i), \quad (10)$$

for some distance measure $\mathcal{D}_{\mathcal{X}}$ on \mathcal{X} . Here \mathcal{C} again is a class of operators which can be defined by neural networks after discretization. One shortcoming of these post-processing networks is that they depend heavily on the set of training data and are vulnerable to adversarial examples or changes in the noise characteristics (Huang et al. 2018b). Including knowledge of the operator \mathbf{R} within the deep learning can potentially remedy these problems and are discussed in the following.

Using Knowledge of the Operator

For the Radon transform, the missing information in the projection data can be characterized in Fourier domain (Frikel and Quinto 2013). For the reconstructed image from incomplete data, the frequency components in a double wedge are

missing. This characterization in frequency domain has been exploited by Gu and Ye (2017) where they apply a directional wavelet transform (conourlets) partitioning the frequency domain and in a further step train a convolutional neural network that estimates the artifacts and finally adds the missing frequencies. A similar approach was proposed by the authors in Bubba et al. (2019). They use a shearlet frame for \mathcal{X} which can be split in visible and invisible coefficients. The shearlet frame is adapted to the operator \mathbf{R} and the set Λ , such that the visible coefficients carry reliable information about the unknown f , whereas the invisible coefficients do not contain relevant information. In a first step, the visible coefficients are obtained by an initial nonlinear reconstruction

$$f^+ \in \arg \min_{f \geq 0} \|\mathcal{S}(f)\|_{1,w} + \frac{1}{2} \|\chi_{\Lambda} g - \chi_{\Lambda} \mathbf{R}(f)\|_{\mathcal{Y}}.$$

Here \mathcal{S} denotes the analysis operator for the shearlet frame and $\|\cdot\|_{1,w}$ a weighted ℓ^1 norm. In a second step, a neural network Φ^* is trained to estimate the invisible coefficients from the visible ones.

$$r = \Phi^*(\mathcal{S}(f^+)). \quad (11)$$

The final reconstruction is obtained by applying the synthesis operator \mathcal{S}^* of the frame to the reliable visible coefficients combined with the learned invisible coefficients (11)

$$f^* = \mathcal{S}^* \left(\mathcal{S}(f^+) + r \right).$$

Other data-consistent deep learning approaches exploiting the knowledge of the operator were proposed in Schwab et al. (2019a,b) and Boink et al. (2020).

Learned Backprojection

Although some works for fully learned reconstructions for tomographic imaging $\Phi: \mathcal{Y} \rightarrow \mathcal{X}$ exist (Zhu et al. 2018; Boink and Brune 2019), they are strongly limited by the size of the images and the data, and for a known forward operator, a fully learned scheme seems inadequate. Nevertheless, it is possible to improve direct inversion methods by deep neural networks. In Würfl et al. (2018), the authors propose a reconstruction framework based on a filtered backprojection algorithm for limited angle tomography. Their framework consists in a weighting layer \mathbf{W} , which performs a pixel-wise independent weighting of the projection data, a 1D convolutional layer Φ with a single convolution mimicking the filtering operation in (7), and a backprojection step. The reconstruction is obtained by

$$f^* = \mathbf{R}^* \left(\Phi(\mathbf{W}(g)) \right).$$

A similar approach of using the backprojection algorithm as basis for the network for photoacoustic tomographic imaging was studied in Schwab et al. (2018, 2019c), where compensation weights were learned in order to improve reconstruction for limited data problems.

Reduction of Metal Artefacts

In the presence of metal in the investigated tissue located in the region $\Omega_m \subset \mathbb{R}^2$, its radiative attenuation coefficient can be modelled by

$$f = \chi_{\Omega_m^c} f + \chi_{\Omega_m} f.$$

Due to the linearity of the Radon transform, this leads to a composition of the data

$$\mathbf{R}f = \mathbf{R}(\chi_{\Omega_m^c} f) + f_m \mathbf{R}(\chi_{\Omega_m}),$$

where f_m denotes the radiative absorption coefficient of the metal. Most methods for artifact reduction in the presence of metal now aim at finding the set

$$\mathbf{M} := \left\{ (\theta, s) \in \mathbb{S}^1 \times \mathbb{R} \mid \mathbf{R}(\chi_{\Omega_m})(\theta, s) = 0 \right\},$$

consisting of the data which is responsible for the artifacts in the reconstruction. The set \mathbf{M} contains the reliable information of the measured data coming from the non-metal region; therefore, knowledge of this set would give the opportunity to remove the corrupted data in this region and apply a data extension operator. One possible approach to identify this set consists in three steps:

- (1) Reconstruction of an image from the raw measurements
- (2) Segmentation of metal in the reconstructed image
- (3) Application of the forward operator to the segmented image to obtain \mathbf{M} .

If the set \mathbf{M} is found, similar to (9), a training set can be generated by computing the Radon transform of the training examples f_i s and the corresponding corrupted data by setting $g_i(\theta, s) = 0$ for $(\theta, s) \notin \mathbf{M}_i$. Here the sets \mathbf{M}_i denote the region of corrupted data for the i th training instance. The extension operator $\Phi^*: \mathcal{Y} \rightarrow \mathcal{Y}$ should then satisfy

$$\Phi^* := \arg \min_{\Phi \in \mathcal{C}} \frac{1}{N} \sum_{i=1}^N \mathcal{D}_{\mathcal{Y}} (\Phi(\chi_{\mathbf{M}_i} g_i), g_i),$$

for the training data $(\chi_{\mathbf{M}_i} g_i, g_i)_{i=1}^N \subset \mathcal{Y} \times \mathcal{Y}$ and some distance measure $\mathcal{D}_{\mathcal{Y}}$ on \mathcal{Y} . Convolutional neural networks are best suited to learn such an extension

operator between the discretized spaces $\mathbf{Y} \rightarrow \mathbf{Y}$; in particular, multi-scale residual networks like the U-net (Park et al. 2017) or generative adversarial networks (Ghani and Karl 2019) are a popular choice for the sinogram correction task. In contrast to the analogue approach for limited angle tomography, here the mask \mathbf{M}_i depends on the particular instance in the training data. Therefore, it is necessary to extract the metal mask before the application of the data extension network. The design of a network that takes the linear interpolated masked data as well as the mask itself as an input was shown to be beneficial (Zhang and Yu 2018). After extending the data by the network, in a second step, any reconstruction method can be used to obtain an image with reduced metal artifacts. The authors in Bayaraa et al. (2020) tackle both, the problem of missing data due to detector offset and artifact reduction from high-density objects in dental cone-beam computerized tomography. They first apply a sinogram correction algorithm to extend the data to the missing region and then apply a complementary deep convolutional network to further improve the reconstruction quality.

A second class of deep learning methods utilizes a reconstruction from raw data and targets the removal of generated artifacts in a post-processing step (cf. Fig. 2). In this case, the training strategy is similar to (10), with some works proposing a patch-wise artifact removal (Gjesteby et al. 2018; Huang et al. 2018a). In Zhang and Yu (2018), the authors employ a convolutional network to obtain a prior image from an initial reconstruction, which is further utilized to generate a full sinogram. The full sinogram is then used for the purpose of replacing the metal trace in the original data and creating a corrected sinogram for the final reconstruction.

Low-Dose Computed Tomography

Since X-ray radiation creates a potential risk for the patient, it is desired to lower the radiation dose. There are two main strategies to achieve a reduction of the X-ray radiation for computed tomography, namely, limiting the X-ray flux by reducing the operating current and minimizing the number of measurements.

Lowering the radiation dose results in a noisy data and consequently a noisy reconstructed image with a low signal-to-noise ratio. This can potentially make medical diagnosis more difficult, and therefore a great amount of algorithms were proposed for improving image reconstruction for low-dose computed tomography. Especially with the availability of large datasets, such as the Low-Dose Parallel Beam (LoDoPaB)-CT data set (Leuschner et al. 2021), there has been a large body of work aimed at improving the reconstruction of low-dose data (Kang et al. 2017). Generally these methods can be categorized into (Chen et al. 2017):

- Filtering in data domain
- Iterative reconstruction
- Image processing

Algorithms of the first and third category have the advantage that they can be efficiently combined with classical reconstruction methods, whereas iterative reconstruction algorithms tend to suffer from numerical complexity. Deep learning methods have proven to be particularly suitable for tackling the reconstruction problem, as they are able to achieve image quality, either favorable or comparable to commercial iterative reconstructions, while at the same time being computationally more efficient (Shan et al. 2019).

Deep learning offers the possibility to account for filtering in the sinogram domain and image processing after some initial reconstruction. Given full dose measurements $g \in \mathcal{Y}$, the low-dose measurements can be defined by $\sigma(g)$, where $\sigma: \mathcal{Y} \rightarrow \mathcal{Y}$ denotes the transformation mapping from full-dose to low-dose data. A neural network Φ^* can then be trained to approximate the inverse of σ (Ghani and Karl 2018). Denoting the training data by $(\sigma(g_i), g_i)_{i=1}^N \subset \mathcal{Y} \times \mathcal{Y}$ and a suitable operator class $\mathcal{C} \subset \{\Phi: \mathcal{Y} \rightarrow \mathcal{Y}\}$, we want to find

$$\Phi^* = \arg \min_{\Phi \in \mathcal{C}} \frac{1}{N} \sum_{i=1}^N \mathcal{D}_{\mathcal{Y}}(\Phi(\sigma(g_i)), g_i).$$

The reconstruction can then be carried out by a classical method after the application of Φ^* to the sinogram.

In contrast to this approach, if we denote the reconstruction from low-dose measurements with a classical method by f^σ , then the learning task in image domain can be formulated by

$$\Phi^* = \arg \min_{\Phi \in \mathcal{C}} \frac{1}{N} \sum_{i=1}^N \mathcal{D}_{\mathcal{X}}(\Phi(f_i^\sigma), f_i),$$

where $\mathcal{C} \subset \{\Phi: \mathcal{X} \rightarrow \mathcal{X}\}$ is a class of possible denoising operators. Convolutional encoder-decoder networks have been found to be particularly well suited for this task (Chen et al. 2017), especially networks denoising not the whole reconstructed image but only image patches.

For low-dose CT reconstructions with undersampled data (small numbers of measurements), approaches in the same line have been proposed. In this application area, the learning task in the data domain entails upsampling the sinogram, whereas the learned image processing should remove the streak artifacts that occur in reconstructions from undersampled data.

Further Methods

The field of data-driven image reconstruction is a very dynamic one and is constantly evolving. We now describe another very important class of methods, namely, the learned iterative methods (Adler and Öktem 2017, 2018; Hauptmann et al. 2020). They also give the opportunity to incorporate knowledge about the forward model into the reconstruction process and yield impressive results

(Wu et al. 2017; Chen et al. 2020; Guazzo 2020). The structure of these learned iterative schemes is as follows. For an existing iterative procedure, a number of iterations is chosen, and in all iterations up to this number, the update process is augmented by a neural network. For given data g and an initial guess f_0 , the final reconstruction \hat{f} in its simplest form are given by

$$f^* = \left(\Phi_{\omega_N}^N \circ \mathbf{G}^N(g) \circ \dots \circ \Phi_{\omega_1}^1 \circ \mathbf{G}^1(g) \right) (f_0),$$

where $\Phi_{\omega_i}^i$ denote the augmentation networks and $\mathbf{G}^i(g)$ iterative updates of the reconstruction. These updates depend on the current iteration but also on the data g , which results in a final reconstruction f^* that is consistent to the given data. This class of algorithms can also be utilized for every inverse imaging problem, where the forward operator can be modeled, including the three example problems discussed above.

Other approaches that will continue to play a very important role in the future are unsupervised methods that do not rely on a paired dataset. Recently a variety of methods have been published in this field of research as well (Kwon and Ye 2021; Lee et al. 2020; Kuanar et al. 2019), just to name a few. The great advantage of these methods is that no paired training data need to be collected, which is very difficult or even impossible in many experimental applications.

Completeness in such a rapidly developing field of research is impossible; nevertheless, a more complete and detailed survey of deep learning methods for inverse imaging is given in the nice review article (Arridge et al. 2019).

Conclusion

Deep learning methods show excellent results for tomographic image reconstruction and represent a promising framework for obtaining good image quality for different measurement cases that create incomplete, corrupted, or noisy data. Various deep learning-based methods have meanwhile been designed in order to optimize tomographic image reconstruction. Among them are learned iterative schemes, network cascades, learned regularizers, and two-step approaches; we presented two-stage strategies of deploying data-driven methods to improve image reconstruction in frequently occurring imperfect data situations in X-ray CT. Most of these approaches can be similarly adapted to other tomographic imaging modalities as well. Nevertheless, it is important to consciously harness the power of deep learning to ensure robustness and guarantee meaningful images for diagnosis. In my opinion, knowledge of the physics (the modelling operator \mathbf{R}) and consistency constraints such as data consistency can help overcome these issues and should be incorporated in the design of deep learning approaches in tomographic imaging. Furthermore, careful and extensive validation and evaluation of these methods including experts' opinions from radiologists and medical doctors are necessary to exploit the indisputable power of deep learning for medical imaging.

References

- Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Probl.* **10**(6), 1217 (1994)
- Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**(12), 124007 (2017)
- Adler, J., Öktem, O.: Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018)
- Anirudh, R., Kim, H., Thiagarajan, J.J., Mohan, K.A., Champley, K., Bremer, T.: Lose the views: limited angle CT reconstruction via implicit sinogram completion. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6343–6352 (2018)
- Arridge, S., Maass, P., Öktem, O., Schönlieb, C.-B.: Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019)
- Bayaraa, T., Hyun, C.M., Jang, T.J., Lee, S.M., Seo, J.K.: A two-stage approach for beam hardening artifact reduction in low-dose dental CBCT. *IEEE Access* **8**, 225981–225994 (2020)
- Beard, P.: Biomedical photoacoustic imaging. *Interface Focus* **1**(4), 602–631 (2011)
- Boink, Y.E., Brune, C.: Learned SVD: solving inverse problems via hybrid autoencoding. *arXiv preprint arXiv:1912.10840* (2019)
- Boink, Y.E., Manohar, S., Brune, C.: A partially-learned algorithm for joint photo-acoustic reconstruction and segmentation. *IEEE Trans. Med. Imaging* **39**(1), 129–139 (2019)
- Boink, Y.E., Haltmeier, M., Holman, S., Schwab, J.: Data-consistent neural networks for solving nonlinear inverse problems. *arXiv preprint arXiv:2003.11253* (2020)
- Bubba, T.A., Kutyniok, G., Lassas, M., Maerz, M., Samek, W., Siltanen, S., Srinivasan, V.: Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.* **35**(6), 064002 (2019)
- Chen, H., Zhang, Y., Kalra, M.K., Lin, F., Chen, Y., Liao, P., Zhou, J., Wang, G.: Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE Trans. Med. Imaging* **36**(12), 2524–2535 (2017)
- Chen, G., Hong, X., Ding, Q., Zhang, Y., Chen, H., Fu, S., Zhao, Y., Zhang, X., Ji, H., Wang, G. et al.: Airmet: fused analytical and iterative reconstruction with deep neural network regularization for sparse-data CT. *Med. Phys.* **47**(7), 2916–2930 (2020)
- Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math. J. Issued Courant Inst. Math. Sci.* **57**(11), 1413–1457 (2004)
- Deans, S.R.: *The Radon Transform and Some of Its Applications*. Courier Corporation, Dover Publications, INC., Mineola, New York (2007)
- Elad, M.: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science & Business Media, New York (2010)
- Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*, vol. 375. Springer Science & Business Media, Dordrecht (1996)
- Friel, J., Quinto, E.T.: Characterization and reduction of artifacts in limited angle tomography. *Inverse Probl.* **29**(12), 125007 (2013)
- Ghani, M.U., Karl, W.C.: CNN based sinogram denoising for low-dose CT. In: *Mathematics in Imaging*, pp. MM2D–5. Optical Society of America, Optical Society of America, Orlando, Florida (2018)
- Ghani, M.U., Karl, W.C.: Fast enhanced CT metal artifact reduction using data domain deep learning. *IEEE Trans. Comput. Imaging*, *IEEE Trans. Comput. Imaging*, vol. 6, 181–193 (2019)
- Gjestebj, L., Shan, H., Yang, Q., Xi, Y., Claus, B., Jin, Y., De Man, B., Wang, G.: Deep neural network for CT metal artifact reduction with a perceptual loss function. In: *Proceedings of the Fifth International Conference on Image Formation in X-Ray Computed Tomography*, vol. 1 (2018)
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge, MA (2016)
- Gu, J., Ye, J.C.: Multi-scale wavelet domain residual learning for limited-angle CT reconstruction. *arXiv preprint arXiv:1703.01382* (2017)

- Guazzo, A.: Deep learning for PET imaging: from denoising to learned primal-dual reconstruction (2020)
- Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media (2009)
- Hauptmann, A., Adler, J., Arridge, S.R., Öktem, O.: Multi-scale learned iterative reconstruction. *IEEE Trans. Comput. Imaging*, vol. 6, 843–856 (2020)
- Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: *Neural Networks for Perception*, pp. 65–93. Elsevier (1992)
- Higham, C.F., Higham, D.J.: Deep learning: an introduction for applied mathematicians. *SIAM Rev.* **61**(4), 860–891 (2019)
- Hounsfield, G.N.: Computerized transverse axial scanning (tomography): part 1. description of system. *Br. J. Radiol.* **46**(552), 1016–1022 (1973)
- Huang, Y., Huang, X., Taubmann, O., Xia, Y., Haase, V., Hornegger, J., Lauritsch, G., Maier, A.: Restoration of missing data in limited angle tomography based on Helgason–Ludwig consistency conditions. *Biomed. Phys. Eng. Express* **3**(3), 035015 (2017)
- Huang, X., Wang, J., Tang, F., Zhong, T., Zhang, Y.: Metal artifact reduction on cervical CT images by deep residual learning. *Biomed. Eng. Online* **17**(1), 175 (2018a)
- Huang, Y., Würfl, T., Breininger, K., Liu, L., Lauritsch, G., Maier, A.: Some investigations on robustness of deep learning in limited angle tomography. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 145–153. Springer (2018b)
- Kang, E., Min, J., Ye, J.C.: A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction. *Med. Phys.* **44**(10), e360–e375 (2017)
- Kuanar, S., Athitsos, V., Mahapatra, D., Rao, K.R., Akhtar, Z., Dasgupta, D.: Low dose abdominal CT image reconstruction: an unsupervised learning based approach. In: *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1351–1355. IEEE (2019)
- Kwon, T., Ye, J.C.: Cycle-free cyclegan using invertible generator for unsupervised low-dose CT denoising. *arXiv preprint arXiv:2104.08538* (2021)
- Landweber, L.: An iteration formula for Fredholm integral equations of the first kind. *Am. J. Math.* **73**(3), 615–624 (1951)
- LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
- Lee, J., Gu, J., Ye, J.C.: Unsupervised CT metal artifact learning using attention-guided beta-cyclegan. *arXiv preprint arXiv:2007.03480* (2020)
- Leuschner, J., Schmidt, M., Bagger, D.O., Maass, P.: LoDoPab-CT, a benchmark dataset for low-dose computed tomography reconstruction. *Sci. Data* **8**(1), 1–12 (2021)
- Li, H., Schwab, J., Antholzer, S., Haltmeier, M.: Nett: solving inverse problems with deep neural networks. *Inverse Probl.* **36**(6), 065005 (2020)
- Lundervold, A.S., Lundervold, A.: An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik* **29**(2), 102–127 (2019)
- Lunz, S., Öktem, O., Schönlieb, C.-B.: Adversarial regularizers in inverse problems. *arXiv preprint arXiv:1805.11572* (2018)
- Mukherjee, S., Dittmer, S., Shumaylov, Z., Lunz, S., Öktem, O., Schönlieb, C.-B.: Learned convex regularizers for inverse problems. *arXiv preprint arXiv:2008.02839* (2020)
- Natterer, F.: *The Mathematics of Computerized Tomography*. SIAM, Philadelphia (2001)
- Obmann, D., Nguyen, L., Schwab, J., Haltmeier, M.: Sparse anett for solving inverse problems with deep learning. In: *2020 IEEE 17th International Symposium on Biomedical Imaging Workshops (ISBI Workshops)*, pp. 1–4. IEEE (2020)
- Park, H.S., Chung, Y.E., Lee, S.M., Kim, H.P., Seo, J.K.: Sinogram-consistency learning in CT for metal artifact reduction. *arXiv preprint arXiv:1708.00607*, 1 (2017)
- Purcell, E.M., Torrey, H.C., Pound, R.V.: Resonance absorption by nuclear magnetic moments in a solid. *Phys. Rev.* **69**(1–2), 37 (1946)
- Quinto, E.T.: Tomographic reconstructions from incomplete data-numerical inversion of the exterior radon transform. *Inverse Probl.* **4**(3), 867 (1988)

- Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational Methods in Imaging. Springer, New York (2009)
- Schwab, J., Antholzer, S., Nuster, R., Haltmeier, M.: Real-time photoacoustic projection imaging using deep learning. arXiv preprint arXiv:1801.06693 (2018)
- Schwab, J., Antholzer, S., Haltmeier, M.: Big in Japan: regularizing networks for solving inverse problems. *J. Math. Imaging Vis.*, vol. 62, 445–455 (2019a)
- Schwab, J., Antholzer, S., Haltmeier, M.: Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Probl.* **35**(2), 025008 (2019b)
- Schwab, J., Antholzer, S., Haltmeier, M.: Learned backprojection for sparse and limited view photoacoustic tomography. In: *Photons Plus Ultrasound: Imaging and Sensing 2019*, vol. 10878, p. 1087837. International Society for Optics and Photonics, SPIE BiOS, San Francisco, California (2019c)
- Shan, H., Padole, A., Homayounieh, F., Kruger, U., Khera, R.D., Nitiwarangkul, C., Kalra, M.K., Wang, G.: Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction. *Nat. Mach. Intell.* **1**(6), 269–276 (2019)
- Wang, G.: A perspective on deep imaging. *IEEE Access* **4**, 8914–8924 (2016)
- Werbos, P.: Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph. D. dissertation, Harvard University (1974)
- Wu, D., Kim, K., El Fakhri, G., Li, Q.: Iterative low-dose CT reconstruction with priors trained by artificial neural network. *IEEE Trans. Med. Imaging* **36**(12), 2479–2486 (2017)
- Wu, D., Kim, K., Kalra, M.K., De Man, B., Li, Q.: Learned primal-dual reconstruction for dual energy computed tomography with reduced dose. In: *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, vol. 11072, p. 1107206. International Society for Optics and Photonics (2019)
- Würfl, T., Hoffmann, M., Christlein, V., Breininger, K., Huang, Y., Unberath, M., Maier, A.K.: Deep learning computed tomography: learning projection-domain weights from image domain in limited angle problems. *IEEE Trans. Med. Imaging* **37**(6), 1454–1463 (2018)
- Zhang, Y., Yu, H.: Convolutional neural network based metal artifact reduction in x-ray computed tomography. *IEEE Trans. Med. Imaging* **37**(6), 1370–1381 (2018)
- Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R., Rosen, M.S.: Image reconstruction by domain-transform manifold learning. *Nature* **555**(7697), 487–492 (2018)



MRI Bias Field Estimation and Tissue Segmentation Using Multiplicative Intrinsic Component Optimization and Its Extensions

34

Samad Wali, Chunming Li, and Lingyan Zhang

Contents

Introduction	1204
Multiplicative Intrinsic Component Optimization	1207
Decomposition of MR Images into Multiplicative Intrinsic Components	1207
Mathematical Description of Multiplicative Intrinsic Components	1208
Energy Formulation for MICO	1209
Optimization of Energy Function and Algorithm	1211
Numerical Stability Using Matrix Analysis	1213
Execution of MICO	1215
Some Extensions	1217
Introduction of Spatial Regularization in MICO	1217
The Proposed TV-Based MICO Model and Its Solver	1217
Spatiotemporal Regularization for 4D Segmentation	1222
Modified MICO Formulation with Weighting Coefficients for Different Tissues	1224
Results and Discussions	1224
Conclusion	1232
References	1232

S. Wali

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

Department of Mathematics, Namal Univeristy, Mianwali, Pakistan

e-mail: samad.walikh@outlook.com

C. Li (✉) · L. Zhang

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

e-mail: chunming.li@uestc.edu.cn; 1799031768@qq.com

Abstract

In medical image analysis, energy minimization-based optimization approaches are invaluable. This chapter presents a joint optimization method called multiplicative intrinsic component optimization (MICO) for magnetic resonance (MR) images in bias field estimation and segmentation. Due to the intensity inhomogeneity in MR images, there are overlaps between the ranges of the intensities of different tissues, which often causes misclassification of tissues. To overcome this problem, our proposed method MICO can estimate bias field without avoiding intensity inhomogeneity and can benefit to achieve superior tissue segmentation results. We extended MICO formulation by connecting total variation (TV) as a convex regularization. In addition, for the TV-based MICO model, we implemented the alternating direction method of multipliers (ADMM), which can solve the model efficiently and guarantee its convergence. Quantitative evaluations and comparisons with other popular software have shown that MICO and TVMICO outperform them in terms of robustness and accuracy.

Keywords

MRI · Brain segmentation · Intensity inhomogeneity · Bias field estimation · Bias field correction · Energy minimization · Multiplicative intrinsic component optimization · 4D segmentation · Total variation · ADMM

Introduction

Image segmentation is a fundamental task in image processing in which an image is divided into numerous disjoint parts so that pixels in the same region have certain consistent properties such as intensity, color, and texture (Stockman and Shapiro 2001). Due to an inherent artifact known as intensity inhomogeneity, segmentation in magnetic resonance imaging (MRI) is a challenging task. It appears as slow intensity variations in the same tissue across the image domain (Li et al. 2008; Vovk et al. 2007). In MRI, intensity inhomogeneity can be caused by a variety of factors, including B0 and B1 field inhomogeneities and patient-centered interactions. Because of intensity inhomogeneity, there are overlaps between the ranges of intensities of various tissues, which frequently leads to tissue misclassification. Intensity inhomogeneities can also mislead other image analysis methods, such as image registration. As a result, before doing a quantitative analysis of MRI data, it is typically necessary to eliminate intensity inhomogeneity using a process known as bias field correction. Bias field correction is typically achieved by estimating the bias field that accounts for the intensity inhomogeneity in the MR image and then dividing the image by the estimated bias field to obtain a bias field corrected image.

Traditional segmentation techniques, such as the K-means clustering algorithm, frequently fail in the presence of image intensity inhomogeneities (Zheng et al. 2018). To use these techniques, bias field correction must be performed as a separate

preprocessing step to eliminate the intensity inhomogeneity (Juntu et al. 2005; Tustison et al. 2010). Because some modern image segmentation algorithms feature an inherent mechanism for dealing with intensity inhomogeneities, they may be used immediately for segmentation without the necessity for bias field correction in a subsequent preprocessing phase. In an iterative procedure, these algorithms often interleave bias field estimation and image segmentation (Li et al. 2014; Guillemaud and Brady 1997).

Wells et al. established a strategy for interleaved bias field estimation and segmentation based on an expectation-maximization (EM) algorithm (Wells et al. 1996). Guillemaud and Brady improved on this strategy in Guillemaud and Brady (1997). However, appropriate initialization is required for either the bias field or the classification estimate in these EM-based approaches (Styner et al. 2000). To accomplish initialization in MRI, these techniques often require manual choices of representative locations for each tissue class. Such initializations are often imprecise and irreproducible (Leemput et al. 1999). Furthermore, the outcome of bias field correction and segmentation is sensitive to the initial condition selections (Vovk et al. 2007).

Pham and Prince introduced an energy minimization strategy for segmentation and bias field estimation in (1999), which employed a fuzzy c-means (FCM) algorithm for image segmentation. Their technique, known as adaptive fuzzy c-means (AFCM), is an extension of FCM that includes a bias field as a component in the cluster centers. A smoothing factor was incorporated in their energy function to assure the smoothness of the bias field. The coefficient of the smoothing component, on the other hand, is sometimes challenging to adjust (Vovk et al. 2007), limiting the algorithm's effectiveness. Pham expanded AFCM to an improved formulation named FANTASM in a subsequent article Pham (2001) by including a spatial regularization procedure on the tissue membership functions. Although spatial regularization reduces the influence of noise, FANTASM suffers from the same issue as AFCM in terms of the smoothing term for the bias field.

The correction of bias fields is a crucial challenge in medical image processing. Over the last two decades, many bias field correction techniques have been presented. Prospective approaches Condon et al. (1987), Simmons et al. (1991), Wicks et al. (1993), Tincher et al. (1993), Axel et al. (1987), McVeigh et al. (1986), Narayana et al. (1988) and retrospective methods (Wells et al. 1996; Johnston et al. 1996; Dawant et al. 1993; Sled et al. 1998; Pham and Prince 1999; Leemput et al. 1999; Styner et al. 2000; Ahmed et al. 2002; Salvado et al. 2005; Li et al. 2008) are the two primary categories of existing bias correction methods. Prospective methods use special hardware or particular sequences to avoid intensity inhomogeneity throughout the sampling process. These approaches can correct some of the intensity inhomogeneities induced by the MR scanner, but they cannot address patient-dependent inhomogeneities, making them of limited utility in practical applications (Likar et al. 2001). Retrospective approaches, in contrast to prospective methods, focus only on the information contained within the collected image and can be used to reduce intensity inhomogeneities induced by patient-dependent effects. Vovk et al. (2007) provides a current survey of bias correcting approaches.

Homomorphic filtering (Johnston et al. 1996) is one of the earliest retrospective approaches for bias field elimination. This approach posits that intensity inhomogeneity is a low-frequency signal that can be smothered by using high-pass filtering. However, because the imaged objects typically contain low frequencies as well, filtering approaches frequently fail to achieve sufficient bias field corrections (Vovk et al. 2007). Dawant et al. (1993) presented a method for estimating the inhomogeneity field using splines fitting to the intensities of chosen points. Their approach is based on manually picking reference points inside white tissue. For bias field correction, in Sled et al. (1998), authors suggested an iterative approach named N3 that is based on intensity histograms. It seeks to generate the smooth bias field that sharpens the image's intensity histogram optimum. In Tustison et al. (2010), the N3 algorithm's implementation was enhanced by employing a quicker and more robust B-spline approximation to construct the bias field.

Variational models using total variation (TV) have been widely employed in a wide range of image applications, including bias field estimation and segmentation (He et al. 2012; Tu et al. 2016; Li et al. 2010). Because of its edge preservation property, convexity, and L1 norm sparsity behavior, total variation is quite beneficial (Li et al. 2016). It was initially employed as a regularization for image denoising (Rudin et al. 1992), and it has since been studied and is still useful for a variety of image-processing applications (Chen 2013). The non-smoothness of the total variation semi-norm, on the other hand, poses a barrier to its minimization. To address this issue, the most popular approach is to replace total variation in image restoration models with smoothed versions of the total variation (Liu et al. 2015). To tackle the non-smoothness issue in total variation, alternating direction method of multiplier (ADMM) (Gabay and Mercier 1976; Glowinski and Marroco 1975) is used, which is similar to split Bregman (Goldstein and Osher 2009) and proved to be particularly beneficial for L1 and TV-type optimization problems. We develop an efficient method by introducing two sets of auxiliary variables with closed-form solutions to all subproblems.

In this chapter, firstly, we propose a novel technique for bias field estimation and tissue segmentation in an energy minimization setting. In an energy minimization technique, bias field estimation and tissue membership functions are performed simultaneously. The proposed method optimizes two multiplicative intrinsic components of an MR image: the bias field, which compensates for intensity inhomogeneity, and the true image, which represents a physical property of the tissues. The spatial features of these two components are completely incorporated in their physical representations with the help of the proposed energy minimization approach. Secondly, we have extended the proposed MICO to total variational-based MICO, which we called TVMICO. We use an alternating direction method of multiplier (ADMM) to solve the TVMICO model. By introducing two new constraints, we have closed-form solutions to each sub-variational problem. Because of the convexity of the energy function in each of its variables, our technique, which we term multiplicative intrinsic component optimization (MICO) and TVMICO, both are robust. The proposed MICO formulation can be naturally extended to 3D/4D segmentation with spatial and temporal regularization.

Multiplicative Intrinsic Component Optimization

The formulation of MICO for bias field estimation and tissue segmentation based on the decomposition of an MRI into two multiplicative components is presented in this section. The proposed energy minimization technique leads to the MICO algorithm for combined bias field estimation and tissue segmentation. We follow Li et al. (2014) for most mathematical formulation and notations.

Decomposition of MR Images into Multiplicative Intrinsic Components

Consider $I(x)$ to be the intensity of an observed MR image at voxel x . In most cases, an MR image can be modeled as follows:

$$I(x) = b(x)J(x) + n(x), \quad (1)$$

where $J(x)$ is the clean image, $b(x)$ is the bias field that accounts for the observed image's intensity inhomogeneity, and $n(x)$ is zero-mean additive noise. The widely accepted assumptions in the literature for both J and b are given in Wells et al. (1996), Leemput et al. (1999), and Pham and Prince (1999). The bias field b is supposed to vary smoothly. The true image J describes a physical characteristic of the tissues being imaged, which should ideally take a specific value for voxels of the same tissue type. As a result, for all point x in the i -th tissue, we assume that $J(x)$ is approximately a constant c_i .

In this chapter, we consider Eq. (1) decomposes the MR image I into two multiplicative components b and J , as well as additive zero-mean noise n . From this aspect, we specify systematically biased field estimation and tissue segmentation as a variational-based problem, which is seeking accurate decomposition of given MRI I into two multiplicative components b and J . It is important to mention here that the bias field b and the true image J are intrinsic components of the observed MR image I . In this chapter, we consider an observed image I as a function $I : \Omega \rightarrow \mathcal{R}$ on a continuous domain Ω .

In computer vision, a given observed image I can be decomposed into reflectance image R and the illumination image S that can be shown in multiplicative form as $I = RS$. These multiplicative components of an observed image are similar to Eq. (1). The terminologies intrinsic images were introduced by Barrow and Tenenbaum in (1978) to express these two multiplicative components. In computer vision, estimating intrinsic images from an observed scene image has been a significant challenge. Several methods for estimating the intrinsic images from a scene image based on different assumptions on the two intrinsic images have been presented (Tappen et al. 2005; Weiss 2001; Kimmel et al. 2003).

The bias field b and the real image J are considered as multiplicative intrinsic components of an observed MR image in this study. From an observed MR image, we present a unique approach for estimating these two components. We should point

out that the method proposed in this chapter differs from those used in computer vision to estimate reflectance and illumination images. In fact, due to a lack of knowledge about the unknown intrinsic images R and S , estimation of intrinsic images is an ill-posed problem.

If no prior knowledge of the multiplicative components b and J of the observed MR image I is used, estimation of these components is an underdetermined or ill-posed problem. To solve the problem, we have to gain some knowledge about the bias field b and true image J . The piecewise constant property of the true image J and the smoothly varying property of the bias field b are used in this study to present a strategy that uses the basic properties of the true image and bias field. In the development of our proposed technique, the decomposition of the MR image I into two multiplicative intrinsic components b and J with their respective spatial properties is completely exploited.

Mathematical Description of Multiplicative Intrinsic Components

We could use a suitable mathematical representation and description of the bias field b and true image J to appropriately utilize their features. Assume we have a collection of functions g_1, \dots, g_M that ensures the bias field's smoothly varying property. The bias field in our method is a linear combination of a series of smooth basis functions. It has been studied that for a given sufficiently large number of M basis, a function can be approximated by a linear combination of several basis functions to an arbitrary degree of accuracy (Powell 1981). We use 20 polynomials of the first three degrees as the basis functions in MICO applications to 1.5T and 3T MRI images. The optimal coefficients w_1, \dots, w_M in the linear combination $b(x) = \sum_{k=1}^M w_k g_k$ are needed to determine and used to estimate the bias field. The coefficients w_1, \dots, w_M are represented as a column vector $\mathbf{w} = (w_1, \dots, w_M)^T$, where $(\cdot)^T$ is the transpose operator. A column vector-valued function $G(x) = (g_1(x), \dots, g_M(x))^T$ represents the basis functions $g_1(x), \dots, g_M(x)$. Therefore, the bias field $b(x)$ can be expressed in the vector form shown below.

$$b(x) = \mathbf{w}^T G(x). \quad (2)$$

In our proposed variational-based minimization approach for bias field estimation, Eq. (2) will be utilized as a vector representation. It enables us to calculate the optimal bias field obtained from the energy minimization problem using efficient vector and matrix calculations, as will be explained in section “[Optimization of Energy Function and Algorithm](#)”.

More formally, the true image J has piecewise approximately constant property, and it can be expressed as follows. We suppose that there are N different types of tissues in the image domain Ω . For x in the i -th tissue, the true image $J(x)$ is approximately a constant c_i . The location where the i -th tissue is located is denoted as Ω_i . The membership function u_i may be used to represent each Ω_i region (tissue).

The membership function u_i is a binary membership function in the ideal case when each voxel contains just one kind of tissue, with $u_i(x) = 1$ for $x \in \Omega_i$ and $u_i(x) = 0$ for $x \notin \Omega_i$. Because of the partial volume effect, one voxel may include more than one type of tissue, especially at the interface between adjacent tissues. In this scenario, the N tissues are represented by fuzzy membership functions $u_i(x)$ with values ranging from 0 to 1 and satisfying $\sum_{i=1}^N u_i = 1$. The fuzzy membership function $u_i(x)$ value can be construed as the proportion of the i -th tissue within the voxel x . A column vector-valued function $\mathbf{u} = (u_1, \dots, u_N)^T$, where T is the transpose operator, can be used to express such membership functions u_1, \dots, u_N . The space of all such vector-valued functions is denoted as \mathcal{U} .

$$\mathcal{U} \triangleq \{ \mathbf{u} = (u_1, \dots, u_N)^T : 0 \leq u_i(x) \leq 1, i = 1, \dots, N, \text{ and} \\ \sum_{i=1}^N u_i(x) = 1, \text{ for all } x \in \Omega \} \quad (3)$$

The true image J can be approximated by the following combination of membership functions u_i and constants c_i .

$$J(x) = \sum_{i=1}^N c_i u_i(x). \quad (4)$$

The function in Eq. (4) is a piecewise constant function when the membership functions u_i are binary functions, with $J(x) = c_i$ for $x \in_i = \{x : u_i(x) = 1\}$. If u_1, \dots, u_N are the binary membership functions, the segmentation is called the hard segmentation, while the corresponding regions $\Omega_1, \dots, \Omega_N$ show an image domain Ω partition, with the conditions as $\cup_{i=1}^N \Omega_i = \Omega$ and $\Omega_i \cap \Omega_j = \emptyset$. On the other hand, the functions u_1, \dots, u_N are fuzzy membership functions with values between 0 and 1 representing a soft segmentation result.

We propose an energy minimization approach for simultaneous bias field estimation and tissue segmentation based on the image model Eq. (1). The membership function $\mathbf{u} = (u_1, \dots, u_N)$ gives the outcome of tissue segmentation. The estimated bias field b is used to compute the bias field corrected image, which is expressed as the reciprocal, i.e., $\frac{I}{b}$.

Energy Formulation for MICO

Based on the image model Eq. (1) and the intrinsic features of the bias field and the true image as mentioned in section “[Decomposition of MR Images into Multiplicative Intrinsic Components](#)”, we present an energy minimization formulation for bias field estimation and tissue segmentation. In light of the image model (1), we address the problem of determining the multiplicative intrinsic components b and J of an observed MR image I to minimize the following energy.

$$F(b, J) = \int_{\Omega} |I(x) - b(x)J(x)|^2 dx. \tag{5}$$

Minimization of energy problem Eq. (5) is obviously an ill-posed problem if the variables b and J are not constrained. Indeed, in the absence of constraints, every nonzero function b and $J = I/B$ optimizes the energy $F(b, J)$. To solve the problem, we must limit the search spaces of b and J by utilizing some information about the unknowns b and J . The characteristics of the bias field b and the true image J described in section “[Decomposition of MR Images into Multiplicative Intrinsic Components](#)” are the information that may be used to limit the search spaces of b and J to specific search subspaces that reflect these properties. Using binary membership functions u_1, \dots, u_N and the knowledge that the true image J is piecewise approximately constant, we can confine the true image J 's search space to the subspace of piecewise constant functions as in Eq. (4) $J(x) = \sum_{i=1}^N c_i u_i(x)$. The search space of the bias field b , on the other hand, is constrained to the subspace of all functions of the type $b(x) = \mathbf{w}^T G(x)$, as shown in Eq. (2). The energy $F(b, J)$ may be written in terms of three variables, $\mathbf{u} = (u_1, \dots, u_N)^T$, $\mathbf{c} = (c_1, \dots, c_N)^T$, and $\mathbf{w} = (w_1, \dots, w_M)^T$, i.e.:

$$F(b, J) = F(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \int_{\Omega} \left| I(x) - \mathbf{w}^T G(x) \sum_{i=1}^N c_i u_i(x) \right|^2 dx, \tag{6}$$

Thus, optimizing b and J involves minimizing the energy F with respect to \mathbf{u} , \mathbf{c} , and \mathbf{w} . Because u_i is the binary membership function of the region Ω_i , we obtain the following:

$$u_i(x) = \begin{cases} 1, & x \in \Omega_i; \\ 0, & x \notin \Omega_i. \end{cases}$$

Therefore, we have as follows:

$$\sum_{i=1}^N c_i u_i(x) = c_i \text{ for } x \in \Omega_i$$

As a result, the energy F may be stated as follows:

$$\begin{aligned} F(\mathbf{u}, \mathbf{c}, \mathbf{w}) &= \int_{\Omega} \left| I(x) - \mathbf{w}^T G(x) \sum_{i=1}^N c_i u_i(x) \right|^2 dx \\ &= \sum_{i=1}^N \int_{\Omega_i} \left| I(x) - \mathbf{w}^T G(x) c_i \right|^2 dx \end{aligned}$$

$$= \sum_{i=1}^N \int_{\Omega} \left| I(x) - \mathbf{w}^T G(x) c_i \right|^2 u_i(x) dx \quad (7)$$

We obtain by rearranging the order of summation and integration in Eq. (7) as follows:

$$F(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \int_{\Omega} \sum_{i=1}^N \left| I(x) - \mathbf{w}^T G(x) c_i \right|^2 u_i(x) dx. \quad (8)$$

The formulation of the energy F in Eq. (8) allows us to construct an efficient energy minimization technique, which is discussed in section “[Optimization of Energy Function and Algorithm](#)”. We derive the optimal membership function $\hat{\mathbf{u}} = (u_1, \dots, u_N)^T$ as the segmentation result by minimizing the energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$, as well as the optimal vector $\hat{\mathbf{w}}$, from which the estimated bias field is defined by $b(x) = \hat{\mathbf{w}}^T G(x)$.

The ideal membership functions u_1, \dots, u_N that minimize the energy given in Eq. (8) are binary functions with values of 0 or 1, leading to a hard segmentation conclusion, as will be demonstrated in section “[Optimization of Energy Function and Algorithm](#)”. Many applications prefer fuzzy (or soft) segmentation results, which are provided by fuzzy membership functions with values ranging from 0 to 1, as in the fuzzy C-means (FCM) clustering approach (Bezdek et al. 1984). To accomplish fuzzy segmentation, we change the energy function F in Eq. (8) by adding a fuzzifier $q \geq 1$ to generate the following energy:

$$F_q(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \int_{\Omega} \sum_{i=1}^N \left| I(x) - \mathbf{w}^T G(x) c_i \right|^2 u_i^q(x) dx. \quad (9)$$

The optimal membership functions that minimize the energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ for the scenario $q > 1$ are fuzzy membership functions with values between 0 and 1. By minimizing the energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ in Eq. (8) or $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ in Eq. (9), our technique accomplishes image segmentation and bias field estimation, subject to the constraints $\mathbf{u} \in \mathcal{U}$. The fact that the energy $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ is convex in each variable, \mathbf{u} , \mathbf{c} , or \mathbf{w} , is a desired characteristic (Li et al. 2009). This characteristic guarantees that the energy $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ has a unique minimum point for each of its variables.

Optimization of Energy Function and Algorithm

We used alternating minimization technique in which one can achieve the minimum and independent solution of $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ with respect to each of its variables given

the other two fixed. The alternating minimization of $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ with respect to each of its variables is described below.

Optimization of \mathbf{c}

The energy $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ is optimized with respect to the variable \mathbf{c} for fixed \mathbf{w} and $\mathbf{u} = (u_1, \dots, u_N)^T$. It is simple to present that $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ is minimized by $\mathbf{c} = \hat{\mathbf{c}} = (c_1, \dots, c_N)^T$ with the following:

$$\hat{c}_i = \frac{\int_{\Omega} I(x)b(x)u_i^q(x)dx}{\int_{\Omega} b^2(x)u_i^q(x)dx}, \quad i = 1, \dots, N. \quad (10)$$

Optimization of \mathbf{w} and Bias Field Estimation \hat{b}

We minimize the energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ with respect to the variable \mathbf{w} for fixed \mathbf{u} and \mathbf{c} . This may be accomplished by solving the equation $\frac{\partial F}{\partial \mathbf{w}} = 0$. It is simple to demonstrate that:

$$\frac{\partial F}{\partial \mathbf{w}} = -2\mathbf{v} + 2A\mathbf{w}$$

where \mathbf{v} is a column vector with M dimensions and here A is an $M \times M$ matrix provided by the following:

$$\mathbf{v} = \int_{\Omega} G(x)I(x) \left(\sum_{i=1}^N c_i u_i^q(x) \right) dx, \quad (11)$$

$$A = \int_{\Omega} G(x)G^T(x) \left(\sum_{i=1}^N c_i^2 u_i^q(x) \right) dx. \quad (12)$$

The equation $\frac{\partial F}{\partial \mathbf{w}} = 0$ can be represented as a linear equation:

$$A\mathbf{w} = \mathbf{v} \quad (13)$$

We compute the estimated bias field as $\hat{b}(x) = \hat{\mathbf{w}}^T G(x)$ given the solution to this equation, $\hat{\mathbf{w}} = A^{-1}\mathbf{v}$. The non-singularity of matrix A is demonstrated in section “[Numerical Stability Using Matrix Analysis](#)”.

As a result, the linear equation $\frac{\partial F}{\partial \mathbf{w}} = -2\mathbf{v} + 2A\mathbf{w} = 0$ has a unique solution $\hat{\mathbf{w}} = A^{-1}\mathbf{v}$. The vector $\hat{\mathbf{w}}$ can be represented explicitly by using Eq. (12) as follows:

$$\hat{\mathbf{w}} = \left(\int_{\Omega} G(x)G^T(x) \left(\sum_{i=1}^N c_i^2 u_i^q(x) \right) dx \right)^{-1} \int_{\Omega} G(x)I(x) \left(\sum_{i=1}^N c_i u_i^q(x) \right) dx. \quad (14)$$

The estimated bias field is obtained using the optimum vector $\hat{\mathbf{w}}$ provided by Eq. (14).

$$\hat{b}(x) = \hat{\mathbf{w}}^T G(x) \quad (15)$$

We will verify the non-singularity of the matrix A , as well as the numerical stability of the foregoing calculation for solving the linear system (13) in section “[Numerical Stability Using Matrix Analysis](#)”. These are two critical concerns in the implementation of our proposed technique.

Optimization of \mathbf{u}

We begin with the scenario where $q > 1$ and minimize the energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ for fixed \mathbf{c} and \mathbf{w} , subject to the constraint that $\mathbf{u} \in \mathcal{U}$. It can be demonstrated that $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ is minimized at $\mathbf{u} = \hat{\mathbf{u}} = (\hat{u}_1, \dots, \hat{u}_N)^T$, obtained by the following:

$$\hat{u}_i(x) = \frac{(\delta_i(x))^{\frac{1}{1-q}}}{\sum_{j=1}^N (\delta_j(x))^{\frac{1}{1-q}}}, \quad i = 1, \dots, N, \quad (16)$$

where:

$$\delta_i(x) = |I(x) - \mathbf{w}^T G(x)c_i|^2. \quad (17)$$

For $q = 1$, it can be presented that the minimizer $\hat{\mathbf{u}} = (\hat{u}_1, \dots, \hat{u}_N)^T$ is provided by the following:

$$\hat{u}_i(x) = \begin{cases} 1, & i = i_{\min}(x); \\ 0, & i \neq i_{\min}(x), \end{cases} \quad (18)$$

where:

$$i_{\min}(x) = \arg \min_i \{\delta_i(I(x))\}.$$

Numerical Stability Using Matrix Analysis

The bias field estimate computation comprises calculating the vector \mathbf{v} in (11), the matrix A in (12), and the inverse matrix A^{-1} in (14). The matrix A is an $M \times M$ matrix, where M is the number of basis functions. We use $M = 20$ basis functions in this chapter; hence the dimension of matrix A is a 20×20 . The non-singularity of the matrix A assures that the inverse matrix A^{-1} exists and that the Eq. (13) has a unique solution. We will also demonstrate that the numerical calculation of the inverse matrix A^{-1} is stable.

The non-singularity of matrix A stated in Eq. (12) is demonstrated in the following way. We begin by defining $h_m(x) \triangleq g_m(x)\sqrt{\sum_{i=1}^N c_i^2 u_i^q(x)}$. Thus, the (m, k) entry of the matrix A can be represented as the inner product of h_m and h_k provided by the following:

$$\langle h_m, h_k \rangle = \int_{\Omega} h_m(x)h_k(x)dx.$$

As a result, the matrix A is the *Gramian matrix* of h_1, \dots, h_M . The Gramian matrix of h_1, \dots, h_M is non-singular according to linear algebra (Horn and Johnson 1985) if and only if they are linearly independent. It is clear that the above-defined functions h_1, \dots, h_M are linearly independent, implying that A is non-singular.

The importance of numerical stability in solving the Eq. (13) cannot be overstated. The *condition number* of the matrix A characterizes the numerical stability of solving the Eq. (13); for more details see Golub and Loan (1996). A positive-definite matrix A 's condition number is given by the following:

$$\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A),$$

where $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the minimal and maximal eigenvalues of matrix A , respectively. For very large value of the condition number $\kappa(A)$, minor variations in the matrix A or the vector \mathbf{v} , which are most likely caused by image noise and accumulating intermediate rounding errors, can cause very large variation of the solution $\hat{b}\hat{w}$ to the Eq. (13). As a result, it is vital to guarantee that the condition number $\kappa(A)$ is not huge, as shown below, to ensure the robustness of the bias field computation.

The matrix analysis that follows is predicated on the orthogonality of the basis functions, that is:

$$\int_{\Omega} g_m(x)g_k(x)dx = \delta_{mk}, \tag{19}$$

here $\delta_{mk} = 0$ for $m \neq k$ and $\delta_{mk} = 1$ for $m = k$.

It can be demonstrated that for the above-specified matrix A in Eq. (12) with the basis functions g_1, \dots, g_M satisfying the orthogonality criterion in Eq. (19):

$$0 < \min_i\{c_i^2\} \leq \lambda_{\min}(A) \leq \lambda_{\max}(A) \leq \max_i\{c_i^2\}$$

As a result, A 's condition number is determined by the following:

$$\kappa(A) \leq \frac{\max_i\{c_i^2\}}{\min_i\{c_i^2\}}. \tag{20}$$

For instance, if $\max_i\{c_i\} = 250$ and $\min_i\{c_i\} = 50$, by the inequality (20), we have $\kappa(A) \leq \frac{250^2}{50^2} = 25$. We observed that the condition numbers of the matrix A are at

this level in the implementations of our approach to actual MRI data, which is small enough to assure the numerical stability of the inversion operation.

Execution of MICO

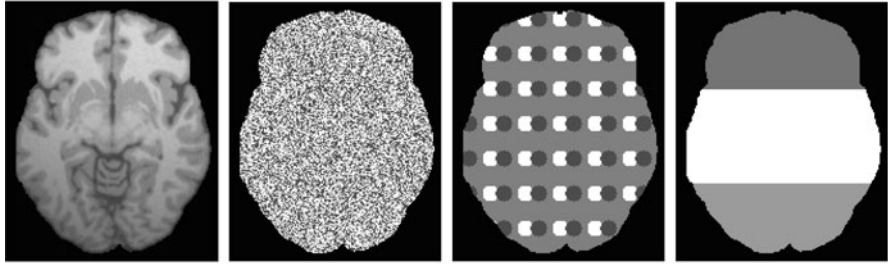
We summarize the technique for minimizing the energy $F_q(\mathbf{u}, \mathbf{c}, \mathbf{w})$ for $q \geq 1$ as the following iteration process from section “[Optimization of Energy Function and Algorithm](#)”:

- Step-1. Initialize \mathbf{u} and \mathbf{c} .
- Step-2. Update b as \hat{b} in Eq. (15).
- Step-3. Update \mathbf{c} as $\hat{\mathbf{c}}$ in Eq. (10).
- Step-4. Update \mathbf{u} as $\hat{\mathbf{u}}$ in Eq. (16) for the case $q > 1$ or (18) for the case $q = 1$.
- Step-5. Check the convergence condition: if convergence has been obtained or the iteration number exceeds a predefined maximum number, terminate the iteration; otherwise, go to Step-2.

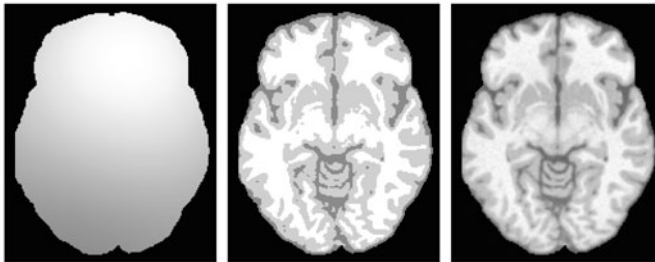
During the iteration procedure described above, each of the three variables is updated with the other two variables computed in the previous iteration. In Step-1 of the preceding iteration process, we only need to initialize two of the three variables, such as \mathbf{u} and \mathbf{c} . In Step-5, the convergence criteria is $|\mathbf{c}^{(n)} - \mathbf{c}^{(n-1)}| < \varepsilon$, where $\mathbf{c}^{(n)}$ is the vector \mathbf{c} updated in Step-3 at the n -th iteration, and ε is set to 0.001.

We used a synthetic image in Fig. 1a to show the robustness of our proposed technique to initialization, using three alternative initializations of the membership functions u_1, \dots, u_N and the constants c_1, \dots, c_N . The initial membership function $\mathbf{u} = (u_1, \dots, u_N)$ and the vector $\mathbf{c} = (c_1, \dots, c_N)$ can be visualized as an image defined by $J_{\mathbf{u}, \mathbf{c}}(x) = \sum_{i=1}^N c_i u_i(x)$. The images $J_{\mathbf{u}, \mathbf{c}}$ for the three different initializations of \mathbf{u} and \mathbf{c} are shown in Fig. 1b, c, and d that show a wide range of patterns. The first initialization illustrated in Fig. 1b is achieved by randomly generating the membership functions $u_1(x), \dots, u_N(x)$ and the constants c_1, \dots, c_N . The bias field converges to the same function for these three alternative initializations of \mathbf{u} and \mathbf{c} up to a scalar multiple. The three estimated bias fields are the same, up to a minor difference, when the bias fields are normalized (e.g., dividing the bias field b by its maximum value $\max_x \{b(x)\}$), as shown in Fig. 1e. Meanwhile, the membership function \mathbf{u} converges to the same vector-valued function, with just a minor variation, providing the identical segmentation result as shown in Fig. 1f. The corrected bias field image is provided in Fig. 1g.

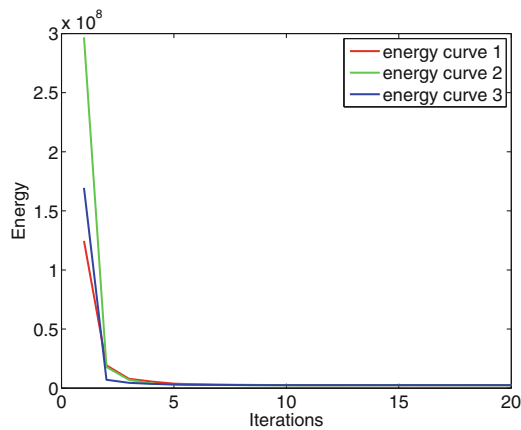
We display the energy minimization $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ of the variables \mathbf{u} , \mathbf{c} , and \mathbf{w} computed at each iteration up to the 20 iterations in Fig. 1h. The energy $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ rapidly drops to the same value from three distinct initial values corresponding to three separate initializations. Figure 1h also presents the fast convergence of the iteration in MICO, as we can see that the energy is rapidly decreased and converges to the minimal value in less than 10 iterations. As a result, in our MICO applications, we often just perform 10 iterations.



(a) Original image. (b) Initialization 1. (c) Initialization 2. (d) Initialization 3.



(e) Estimated bias field. (f) Segmentation result. (g) Bias field corrected image.



(h) Energy minimization curve.

Fig. 1 Robustness of our proposed method to different initializations. (a) Original image, (b)–(d) three possible initializations of the membership functions are visualized, (e) estimated bias field, (f) segmentation result, (g) bias field correction result, (h) curves illustrating the energy F used in the iteration process for three different initializations (b), (c), and (d)

Some Extensions

Introduction of Spatial Regularization in MICO

The original MICO formulation described above can be easily extended by including a regularization term on the membership functions. Regularization of the membership functions can be accomplished using the MICO formulation by combining the total variations (TV) of the membership functions in the following energy:

$$\mathcal{F}(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \lambda F(\mathbf{u}, \mathbf{c}, \mathbf{w}) + \sum_{i=1}^N TV(u_i), \quad (21)$$

where F is the energy defined in (8), $\lambda > 0$ is the weight of F , and TV is the total variations of u defined by the following:

$$TV(u) = \int_{\Omega} |\nabla u(x)| dx. \quad (22)$$

This energy should be minimized subject to the constraint that $0 \leq u_i(x) \leq 1$ and $\sum_{i=1}^N u_i(x) = 1$ for every point x . The variational formulation in (21) is referred to by *TVMICO* formulation. The definition of this energy (21) is simple; however, dealing with the aforementioned point-wise constraint is not straightforward in the context of energy minimization.

Many scholars have developed numerous numerical approaches (Goldstein and Osher 2009) to address variational problems in the context of image segmentation using a TV regularization term $TV(u)$ for a membership function u subject to the constraint $0 \leq u(x) \leq 1$ in recent years. These approaches can only segment images into two complementary regions, denoted by the membership functions u and $1-u$. In general, three or more membership functions u_1, \dots, u_N are employed to represent $N > 2$ regions for segmentation. Li et al. developed a numerical strategy to address the energy minimization problem with TV regularization on the membership functions in Li et al. (2010); they used the operator splitting method proposed by Lions and Mercier in (1979). The numerical technique provided in Li et al. (2010) can be used to minimize the energy \mathcal{F} with respect to the membership functions u_1, \dots, u_N in Eq. (21). The energy minimizations with respect to the variables \mathbf{c} and \mathbf{w} , which are independent of the TV regularization term of the membership functions, remain the same as described in section “[Optimization of Energy Function and Algorithm](#)”.

The Proposed TV-Based MICO Model and Its Solver

Formulation of Proposed Model

Equation (21) can be modified with the help of the definition of total variation as follows:

$$\min_{\mathbf{u}, \mathbf{c}, \mathbf{w}} \lambda \int_{\Omega} \left| I(x) - \sum_{i=1}^N c_i \mathbf{w}^T G(x) \right|^2 u_i^q(x) dx + \sum_{i=1}^N \int_{\Omega} \|\nabla u_i^q(x)\| dx, \quad (23)$$

where λ is a positive parameter which can balance the length of the boundary $\partial\Omega_i$ because Tv the second term in Eq. 23 equals to the length of the boundary Ω at i th position. We will discuss both cases for $q = 1$ and $q > 1$. When $q = 1$, u_i can only take values 0 and 1, and then the vector-valued function for bounded variation space can be defined as follows:

$$\mathcal{U}_0 \triangleq \left\{ \mathbf{u} = (u_1, \dots, u_N)^T : u_i \in BV(\omega), u_i(x) \in \{0, 1\}, i = 1, \dots, N, \right. \\ \left. \text{and } \sum_{i=1}^N u_i(x) = 1, \text{ for all } x \in \Omega \right\} \quad (24)$$

At each point x , there is only one function with a value of 1, while all the other functions have a value of 0. As a result, set \mathcal{U}_0 is not continuous, which causes challenges and instability in numerical implementations. However, we may relax binary indicator function defined in Eq. 24 to fuzzy membership functions u_i that meet the nonnegativity and sum-to-one constraint, i.e., (u_1, \dots, u_N) belongs to the set described as \mathcal{U} in Eq. (3). It is self-evident that $u_i(x) \in [0, 1]$ and is a simplex at any x . As a result, $u_i(x)$ may be thought of as the chance that pixel x belongs to the i th class.

The proposed model Eq. 23 is a convex with respect to \mathbf{u} , \mathbf{c} , and \mathbf{w} independently, but not in together. The TV could be with L_2 (He et al. 2012) and L_1 (Li et al. 2016) fidelity terms. We can also use some nonlinear and nonconvex regularizations such as total generalized variation (Wali et al. 2019a) and Euler’s elastica (Liu et al. 2019; Wali et al. 2019b) for further extensions; however, these models need more constrains to relax and require efficient algorithm such as ADMM. In this section, we only focus on L_1 fidelity term, and we called our proposed method as total variation-based multiplicative intrinsic component optimization (TVMICO).

ADMM and Its Numerical Analysis

In this subsection, ADMM is used to solve the proposed fuzzy-based MICO model (23). We introduce two additional variables $p = (p_1, \dots, p_N)$ and $v = (v_1, \dots, v_N)$ with constraints as $\nabla u_i = p_i$ and $u_i = v_i$. With these constraints the minimization problem Eq. (23) can be written as follows:

$$\min_{\mathbf{p}, \mathbf{v}, \mathbf{u}, \mathbf{c}, \mathbf{w}} \sum_{i=1}^N \left\{ \lambda \int_{\Omega} \left| I(x) - c_i \mathbf{w}^T G(x) \right|^2 v_i(x) dx + \int_{\Omega} \|p_i(x)\| dx \right\} + l_{\mathcal{U}}(\mathbf{v}), \\ \text{subject to } \nabla u_i = p_i, u_i = v_i, \forall i = 1, \dots, N, \quad (25)$$

where $l_{\mathcal{U}}$ is the indicator function, i.e.:

$$l_{\mathcal{Q}}(\mathbf{v}) = \begin{cases} 0, & \mathbf{v} \in \mathcal{Q}; \\ \infty, & \text{otherwise.} \end{cases}$$

The unconstrained augmented Lagrangian functional for Eq. (25) can be formulated as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{p}, \mathbf{v}, \mathbf{u}, \mathbf{c}, \mathbf{w}; \mu_{\mathbf{p}}, \mu_{\mathbf{v}}) &= \sum_{i=1}^N \left\{ \lambda \int_{\Omega} \left| I(x) - c_i \mathbf{w}^T G(x) \right|^2 v_i(x) dx + \int_{\Omega} \left\| p_i(x) \right\| dx \right\} \\ &+ l_{\mathcal{Q}}(\mathbf{v}), + \sum_{i=1}^N \left\{ \langle \mu_{p_i}, \nabla u_i - p_i \rangle + \frac{\gamma}{2} \int_{\Omega} \left\| \nabla u_i(x) - p_i(x) \right\|^2 dx \right\} \\ &+ \sum_{i=1}^N \left\{ \langle \mu_{v_i}, u_i - v_i \rangle + \frac{\gamma}{2} \int_{\Omega} \left| u_i(x) - v_i(x) \right|^2 dx \right\}, \end{aligned} \quad (26)$$

where $\mu_{p_i} = \mu_{p_1}, \dots, \mu_{p_N}$ and $\mu_{v_i} = \mu_{v_1}, \dots, \mu_{v_N}$ are Lagrange multipliers and γ is a positive constant. Here $\langle \mu_{p_i}, \nabla u_i - p_i \rangle = \int_{\Omega} \mu_{p_i}^T(x) (\nabla u_i(x) - p_i(x)) dx$ and $\langle \mu_{v_i}, u_i - v_i \rangle = \int_{\Omega} \mu_{v_i}^T(x) (u_i(x) - v_i(x)) dx$. The ADMM can update Lagrangian multipliers after solving primal variables in Gauss-Seidel manner. The ADMM for solving Eq. (26) can be described in the following Algorithm 1.

Algorithm 1 Proposed alternating direction method of multipliers for (26)

1. **Initialization:** primal and dual variables $\mathbf{p}^0, \mathbf{v}^0, \mathbf{u}^0, \mathbf{c}^0, \mathbf{w}^0$ and Lagrange multipliers $\mu_{\mathbf{p}}^0, \mu_{\mathbf{v}}^0$.
2. **Compute primal and dual variables:** for $k = 1, 2, \dots$:

$$\mathbf{p}^{k+1} = \arg \min_{\mathbf{p}} \mathcal{L}(\mathbf{p}, \mathbf{v}^k, \mathbf{u}^k, \mathbf{c}^k, \mathbf{w}^k; \mu_{\mathbf{p}}^k, \mu_{\mathbf{v}}^k) \quad (27)$$

$$\mathbf{v}^{k+1} = \arg \min_{\mathbf{v}} \mathcal{L}(\mathbf{p}^{k+1}, \mathbf{v}, \mathbf{u}^k, \mathbf{c}^k, \mathbf{w}^k; \mu_{\mathbf{p}}^k, \mu_{\mathbf{v}}^k) \quad (28)$$

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{u}} \mathcal{L}(\mathbf{p}^{k+1}, \mathbf{v}^{k+1}, \mathbf{u}, \mathbf{c}^k, \mathbf{w}^k; \mu_{\mathbf{p}}^k, \mu_{\mathbf{v}}^k) \quad (29)$$

$$\mathbf{c}^{k+1} = \arg \min_{\mathbf{c}} \mathcal{L}(\mathbf{p}^{k+1}, \mathbf{v}^{k+1}, \mathbf{u}^{k+1}, \mathbf{c}, \mathbf{w}^k; \mu_{\mathbf{p}}^k, \mu_{\mathbf{v}}^k) \quad (30)$$

$$\mathbf{w}^{k+1} = \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{p}^{k+1}, \mathbf{v}^{k+1}, \mathbf{u}^{k+1}, \mathbf{c}^{k+1}, \mathbf{w}; \mu_{\mathbf{p}}^k, \mu_{\mathbf{v}}^k) \quad (31)$$

3. **Update the Lagrange multipliers:**

$$\mu_{p_i}^{k+1} = \mu_{p_i}^k + \gamma (\nabla u_i^{k+1} - p_i^{k+1})$$

$$\mu_{v_i}^{k+1} = \mu_{v_i}^k + \gamma (u_i^{k+1} - v_i^{k+1})$$

4. **Endfor** until some stopping criterion meets and get **output**.
-

In the following, we will present the solutions of subproblems individually.

p-Subproblem

We can write terms from (26) associated with primal variable \mathbf{p} and fixed all other variables as follows:

$$\mathbf{p}^{k+1} = \arg \min_{\mathbf{p}} \sum_{i=1}^N \left\{ \int_{\Omega} \|p_i(x)\| dx - \int_{\Omega} (\mu_{p_i}^k(x))^T p_i(x) dx + \frac{\gamma}{2} \int_{\Omega} \|\nabla u_i^k(x) - p_i(x)\|^2 dx \right\}. \tag{32}$$

Equation (32) is equivalent to the following:

$$\mathbf{p}^{k+1} = \arg \min_{\mathbf{p}} \sum_{i=1}^N \left\{ \int_{\Omega} \|p_i(x)\| dx + \frac{\gamma}{2} \int_{\Omega} \|p_i(x) - X^k\|^2 dx \right\}, \tag{33}$$

where $X^k = \nabla u_i^k(x) + \frac{\mu_{p_i}^k(x)}{\gamma}$. Equation (33) has a close form solution, and it can be solved by shrinkage operator; we can compute \mathbf{p}^{k+1} as follows:

$$\mathbf{p}^{k+1} = S\left(X^k, \frac{1}{\gamma}\right). \tag{34}$$

S denotes the shrinkage operator, which is defined as follows:

$$S(X, \gamma) = \frac{X}{\|X\|} * \max(\|X\| - \gamma, 0).$$

v-Subproblem

The subproblem for \mathbf{v} is as follows:

$$\mathbf{v}^{k+1} = \arg \min_{\mathbf{v}} \sum_{i=1}^N \left\{ \lambda \int_{\Omega} \left| I(x) - c_i^k(\mathbf{w}^k)^T G(x) \right|^2 v_i(x) dx - \int_{\Omega} (\mu_{v_i}^k(x))^T v_i(x) dx + \frac{\gamma}{2} \int_{\Omega} \left| u_i^k(x) - v_i(x) \right|^2 dx + l_{\mathcal{Q}}(\mathbf{v}) \right\}. \tag{35}$$

Equation (35) is equivalent to the following:

$$\mathbf{v}^{k+1} = \arg \min_{\mathbf{v}} \sum_{i=1}^N \left\{ \frac{\gamma}{2} \int_{\Omega} \left| v_i(x) - Y^k \right|^2 \right\} + l_{\mathcal{Q}}(\mathbf{v}), \tag{36}$$

where $Y^k = u_i^k(x) + \frac{\mu_{v_i}^k(x)}{\gamma} - \frac{\lambda |I(x) - c_i^k(\mathbf{w}^k)^T G(x)|^2}{\gamma}$. Because \mathcal{Q} is a convex simplex at any x in domain Ω , the solution is given by the following:

$$\mathbf{v}^{k+1} = \Pi_{\mathcal{U}}\left(\left[Y^k\right]_{i=1}^N\right), \quad (37)$$

where Π denotes the projection onto the simplex \mathcal{U} ; for more details, please see Chen and Ye (2011).

u-Subproblem

The subproblem for \mathbf{u} is as follows:

$$\begin{aligned} \mathbf{u}^{k+1} = \arg \min_{\mathbf{u}} \sum_{i=1}^N \left\{ \int_{\Omega} (\nabla u_i^k(x))^T \mu_{p_i}(x) + \frac{\gamma}{2} \int_{\Omega} \left\| \nabla u_i^k(x) - p_i^{k+1}(x) \right\|^2 dx \right. \\ \left. + \int_{\Omega} (u_i^k(x))^T \mu_{v_i}^k(x) + \frac{\gamma}{2} \int_{\Omega} \left| u_i(x) - v_i^{k+1}(x) \right|^2 dx \right\}. \end{aligned} \quad (38)$$

Its identical representation is as follows:

$$\mathbf{u}^{k+1} = \arg \min_{\mathbf{u}} \sum_{i=1}^N \frac{\gamma}{2} \left\{ \int_{\Omega} \left\| \nabla u_i(x) - Z_1^k \right\|^2 + \left| u_i(x) - Z_2^k \right|^2 \right\}, \quad (39)$$

where $Z_1^k = p_i^{k+1}(x) - \frac{\mu_{p_i}^k(x)}{\gamma}$ and $Z_2^k = v_i^{k+1}(x) - \frac{\mu_{v_i}^k(x)}{\gamma}$. By using first optimality condition for each \mathbf{u}^{k+1} , we have the following:

$$\nabla^T \left(\nabla u_i(x) - Z_1^k \right) + \left(u_i(x) - Z_2^k \right) = 0.$$

The closed-form solution of \mathbf{u}^{k+1} can be produced from the following equation:

$$(\nabla^T \nabla + I) u_i^{k+1}(x) = \nabla^T p_i^{k+1}(x) + v_i^{k+1}(x) - \frac{\nabla^T \mu_{p_i}^k(x)}{\gamma} - \frac{\nabla^T \mu_{v_i}^k(x)}{\gamma}.$$

We follow Wang et al. (2008), where diagonalized technique is used to get the fast solution for \mathbf{u}^{k+1} .

Solutions for Subproblems \mathbf{c} , \mathbf{w} , and Bias Field Estimation \mathbf{b}

The \mathbf{c} -subproblem can be formulated as follows:

$$\mathbf{c}^{k+1} = \arg \min_{\mathbf{c}} \sum_{i=1}^N \left\{ \lambda \int_{\Omega} \left| I(x) - c_i \mathbf{w}^T G(x) \right|^2 v_i(x) dx \right\}. \quad (40)$$

To find \mathbf{c}^{k+1} , we compute the similar solution used in basic MICO described in section “[Multiplicative Intrinsic Component Optimization](#)” with regularization parameter λ .

$$\mathbf{c}^{k+1} = \frac{\int_{\Omega} \lambda I(x) \mathbf{b}(x) u_i^{k+1}(x) dx}{\int_{\Omega} \lambda \mathbf{b}^2(x) u_i^{k+1}(x) dx}, \quad i = 1, \dots, N. \quad (41)$$

Here $\mathbf{b}(x) = (\mathbf{w}^k)^T G(x)$ is the bias field.

$$\begin{aligned} \mathbf{w}^{k+1} &= \left(\int_{\Omega} \lambda G(x) G^T(x) \left(\sum_{i=1}^N (c_i^{k+1})^2 u_i^{k+1}(x) \right) dx \right)^{-1} \\ &\times \int_{\Omega} \lambda G(x) I(x) \left(\sum_{i=1}^N c_i^{k+1} u_i^{k+1}(x) \right) dx. \end{aligned} \quad (42)$$

The estimated bias field is calculated by \mathbf{b}^{k+1} using the optimum vector \mathbf{w}^{k+1} given by Eq. (42).

$$\mathbf{b}^{k+1} = (\mathbf{w}^{k+1})^T G(x) \quad (43)$$

Spatiotemporal Regularization for 4D Segmentation

The TVMICO formulation in (21) can be further extended to 4D MICO with spatiotemporal regularization of the tissue membership functions for segmentation of 4D data, which is a series of 3D scans of the same subject at different time points. While the basic MICO formulation described in section “[Multiplicative Intrinsic Component Optimization](#)” allows for multiple 4D extensions with different spatiotemporal regularization mechanisms, we only provide a simple and natural 4D extension of the basic MICO formulation as an example in the following.

We first outline a model of serial MR images collected from the same subject at different periods before presenting the 4D MICO formulation. By employing rigid registration with six degrees of freedom, we assumed that all images in a longitudinal series are registered to the first image in the series. As a result, all of the registered images in the series are in a same space, denoted by Ω , which can be represented by a 4D image $I(x, t)$ with spatial variable $x \in \Omega$ and temporal variable t in a time period $[0, L]$. Here $I(\cdot, t)$ can be modeled as a series of images.

$$I(x, t) = b(x, t)J(x, t) + n(x, t) \quad (44)$$

where $J(\cdot, t)$ is the true image, $b(\cdot, t)$ is the bias field, and $n(\cdot, t)$ is additive noise.

We assume there are N types of tissues in the image domain Ω . The true image $J(x, t)$ can be approximated by $J(x, t) = \sum_{i=1}^N c_i(t)u_i(x, t)$, where N is the number of tissues in Ω , $u_i(\cdot, t)$ is the membership function of the i -th tissue, and the constant $c_i(t)$ is the value of the true image $J(x, t)$ in the i -th tissue. For convenience, we represent the constants $c_1(t), \dots, c_N(t)$ with a column vector

$\mathbf{c}(t) = (c_1(t), \dots, c_N(t))^T$. The membership functions $u_1(x, t), \dots, u_N(x, t)$ are also represented by a vector-valued function $\mathbf{u}(x, t) = (u_1(x, t), \dots, u_N(x, t))^T$.

The bias field $b(\cdot, t)$ at each time point t is estimated by a linear combination of a set of smooth basis functions $g_1(x), \dots, g_M(x)$. Using the vector representation in section “[Mathematical Description of Multiplicative Intrinsic Components](#)”, the bias field $b(\cdot, t)$ at the time point t can be expressed as follows:

$$b(x, t) = \mathbf{w}(t)^T G(x), \quad (45)$$

with $\mathbf{w}(t) = (w_1(t), \dots, w_M(t))^T$, where $w_1(t), \dots, w_M(t)$ are the time-dependent coefficients of the basis function $g_j(x)$, $j = 1, \dots, M$.

The spatiotemporal regularization of the membership functions $u_i(x, t)$ can be naturally taken into account in the following variational formulation with a data term (image-based term) and a spatiotemporal regularization term as follows:

$$\mathcal{G}(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \lambda \int_{[0, L]} F(\mathbf{u}(\cdot, t), \mathbf{c}(t), \mathbf{w}(t)) dt + \sum_{i=1}^N TV(u_i) \quad (46)$$

where $\lambda > 0$ is a constant, $F(\mathbf{u}(\cdot, t), \mathbf{c}(t), \mathbf{w}(t))$ is the data term defined in (8) for the image $I(\cdot, t)$ at the time point t , namely:

$$F(\mathbf{u}(\cdot, t), \mathbf{c}(t), \mathbf{w}(t)) = \int_{\Omega} \sum_{i=1}^N |I(x, t) - \mathbf{w}(t)^T G(x) c_i(t)|^2 u_i^q(x, t) dx,$$

and $TV(u_i)$ is the spatiotemporal regularization term on the membership function \mathbf{u} , which can be expressed as follows:

$$TV(u_i) = \int |\nabla u_i(x, t)| dx dt, \quad (47)$$

where the gradient operator ∇ is with respect to the spatial and temporal variables x and t . We call the above variational formulation a 4D MICO formulation.

The minimization of the energy \mathcal{G} is subject to the constraints on the membership function. Therefore, we solve the following constrained energy minimization problem:

$$\text{Minimize } \mathcal{G}(\mathbf{u}, \mathbf{c}, \mathbf{w}) \quad (48)$$

$$\text{subject to } 0 \leq u_i(x) \leq 1, i = 1, \dots, N, \text{ and } \sum_{i=1}^N u_i(x) = 1$$

The minimization of the energy \mathcal{G} with respect to $\mathbf{c}(t)$ and $\mathbf{w}(t)$ is independent of the spatiotemporal regularization term in (46). The optimal vectors $\mathbf{c}(t)$ and

$\mathbf{w}(t)$ can be computed for each time point t independently from the image $I(\cdot, t)$ as in the energy minimization for the basic MICO formulation described in section “[Optimization of Energy Function and Algorithm](#)”. The numerical technique in Li et al. (2010) for variational formulations with TV regularization can be used to minimize \mathcal{G} with respect to the 4D membership function \mathbf{u} subject to the constraint in Eq. (48). In our future research work focusing on 4D segmentation based on the fundamental MICO formulation, we will provide a detailed explanation of the numerical approach for addressing the constrained energy minimization problem in Eq. (48) and its modified variants.

Modified MICO Formulation with Weighting Coefficients for Different Tissues

By inserting weighting coefficients $\lambda_1, \dots, \lambda_N$ for the N tissues in the specification of the energy function $F(\mathbf{u}, \mathbf{c}, \mathbf{w})$ in Eq. (8), the basic MICO formulation in section “[Multiplicative Intrinsic Component Optimization](#)” may be adjusted. The modified energy is defined as follows:

$$F(\mathbf{u}, \mathbf{c}, \mathbf{w}) = \int_{\Omega} \sum_{i=1}^N \lambda_i |I(x) - \mathbf{w}^T G(x) \mathbf{c}_i|^2 u_i^q(x) dx, \quad (49)$$

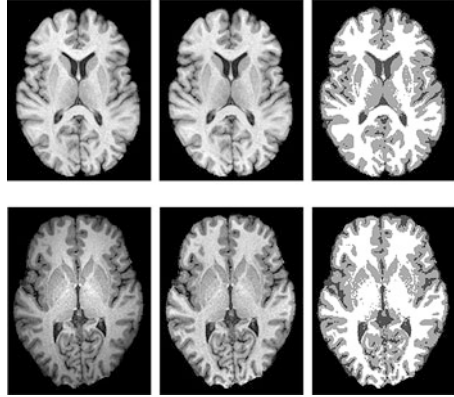
here λ_i is the coefficient for the i -th tissue. The parameters $\lambda_1, \dots, \lambda_N$ provide users the option of improving the outcomes of the standard MICO formulation in 2. For instance, if the i -th tissue is over-segmented using the standard MICO formulation in section “[Multiplicative Intrinsic Component Optimization](#)”, the above-modified formulation in Eq. (49) with a large $\lambda_i > 1$ can be used instead.

Results and Discussions

Our approach has been thoroughly validated on both synthetic and real MRI data, including 1.5T and 3T MRI data. In this part, we first provide experimental results for various synthetic and actual MR images, including those with significant intensity inhomogeneities. We also give quantitative evaluation findings and comparisons with other well-known methodologies.

In our MICO applications for 1.5T and 3T MR images, we employ 20 polynomials of the first three orders as the basis functions g_1, \dots, g_M with $M = 20$. For images obtained from 1.5T and 3T MRI scanners, our technique with these 20 basis functions works effectively. The intensity inhomogeneities in high-field (e.g., 7T) MRI scanners exhibit more complex profiles than 1.5T and 3T MR pictures. More basis functions are required in this circumstance so that a wider variety of bias fields may be well represented by linear combinations. Given an appropriately large number of basis functions, any function can be well approximated by a linear

Fig. 2 Our method’s bias correction and tissue segmentation outcomes on 1.5T (upper row) and 3T (bottom row) MR scanner data. The original image, bias field corrected image, and segmentation result are displayed in the left, center, and right columns, respectively



combination of a set of basis functions up to arbitrary precision (Powell 1981). The numerical stability of the computation of the inverse matrix A^{-1} in Eq. (14), with A being a $M \times M$ matrix, is a significant numerical challenge, especially when M is large. Thanks to the matrix analysis in section “Numerical Stability Using Matrix Analysis”. We have demonstrated that the condition number of the matrix A is bounded by a constant as in Eq. (20), which is independent of the number of basis functions. This provides the numerical stability of the bias field computation, independent of the number of basis functions employed.

In our experiments, MICO has been used to 1.5T and 3T MRI data with promising results. In Fig. 2, we exhibit the bias field correction and segmentation outcomes of our technique for 1.5T and 3T MR images, accordingly. In the left, center, and right columns, the original images, bias field corrected images, and segmentation results are displayed, respectively. We tested MICO on the two images in the left column of Fig. 3 to show that our approach can deal with severe intensity inhomogeneities. The second, third, and fourth columns, respectively, show the estimated bias field, segmentation results, and bias field corrected images acquired by our approach. Despite the images’ severe intensity inhomogeneities, our technique produces desirable bias field correction and tissue segmentation results, as demonstrated in Fig. 3.

The segmentation accuracy of our approach and the well-known software FSL, SPM, and FANTASM are quantitatively evaluated and compared in the following experiment. These three programs are available for free download at <http://www.fmrib.ox.ac.uk/fsl/> (for FSL), <http://www.fil.ion.ucl.ac.uk/spm/software/> (for SPM), and <http://mipav.cit.nih.gov/> (for FANTASM). The data for our quantitative analysis was obtained from BrainWeb (<http://www.bic.mni.mcgill.ca/brainweb/>). BrainWeb also provides ground truth, which can be used to quantify segmentation accuracy.

It is worth noting that the intensity inhomogeneities created by BrainWeb are linear, which makes them reasonably straightforward to handle. To test segmentation algorithms in a more challenging scenario, we created simulated MR images with nonlinear intensity inhomogeneities as shown below. The range of values of the

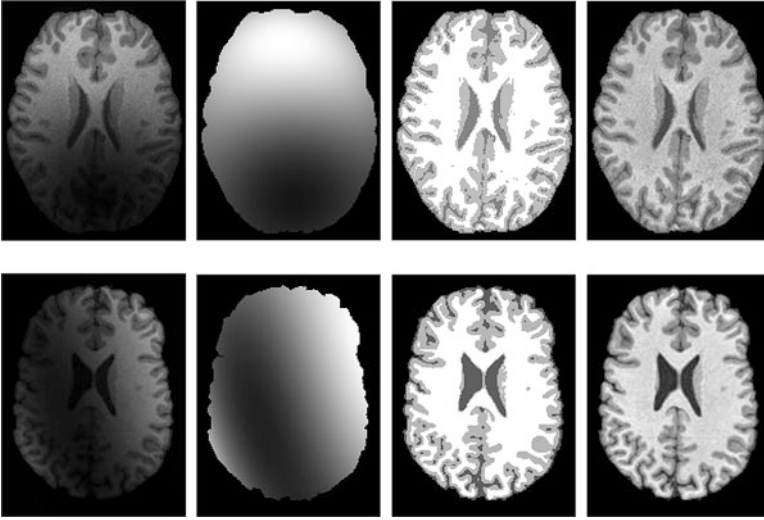


Fig. 3 The left column displays the results for images with extreme intensity inhomogeneity. Columns 2, 3, and 4 show the estimated bias fields, segmentation results, and bias field corrected images, respectively

bias field in the interval $[1 - \alpha, 1 + \alpha]$ with $\alpha > 0$ indicates the degree of intensity inhomogeneity. We created five image sets with $\alpha = 0.1, 0.2, 0.3, 0.4,$ and 0.5 . We constructed six alternative bias fields with values in $[1 - \alpha, 1 + \alpha]$ for each α and multiplied them with the original image obtained from BrainWeb to obtain six images with varying intensity inhomogeneities. The images were then subjected to six different degrees of noise. Thus, the five sets of images have 30 images with varying degrees of intensity inhomogeneities and noise levels. We first show the segmentation results of the 4 tested methods for 2 of the 30 images in Fig. 4; we first show the segmentation results of the 4 tested methods for 2 of the 30 images, 1 with the lowest degree of intensity inhomogeneity (generated with $\alpha = 0.1$) and the other 1 the highest degree of intensity inhomogeneity (generated with $\alpha = 0.5$). By visual comparison, the segmentation results of the four approaches for an image with a low degree of intensity inhomogeneity seem similar, as shown in the upper row of Fig. 4. Our technique has a distinct benefit for images with a high degree of intensity inhomogeneity, as seen in the lower row of Fig. 4.

By evaluating the segmentation results using the Jaccard similarity (JS) index (Shattuck et al. 2001), a more objective and exact comparison of the segmentation accuracy of the four segmentation techniques can be done.

$$J(\mathcal{S}_1, \mathcal{S}_2) = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1 \cup \mathcal{S}_2|}, \quad (50)$$

here $|\cdot|$ indicates a region's area, \mathcal{S}_1 is the algorithm's segmented region, and \mathcal{S}_2 is the corresponding region generated from a reference segmentation result or the

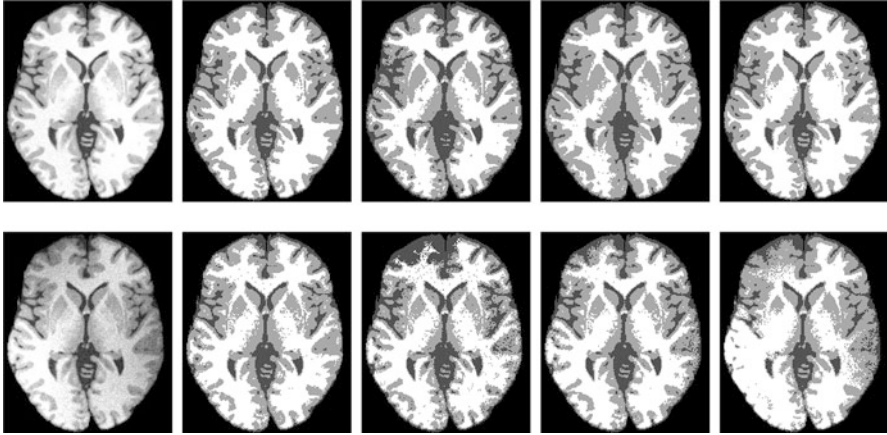


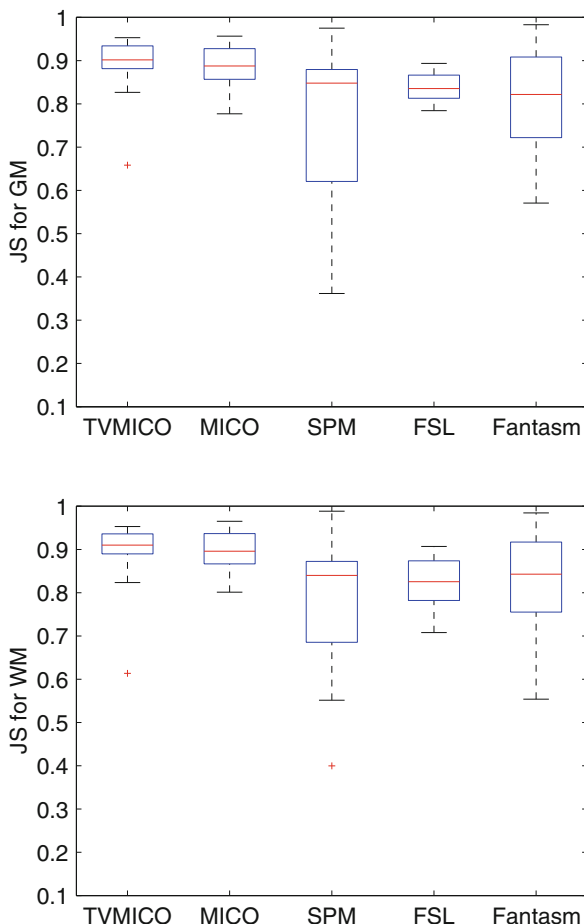
Fig. 4 Comparison of our method with SPM, FSL, and FANTASM on synthetic images with different degrees of intensity inhomogeneities. The input images are displayed in the first column from the left, containing one with a low degree of intensity inhomogeneity (in the top row) and one with a high degree of intensity inhomogeneity (in the lower row). The segmentation results of our technique, SPM, FSL, and FANTASM are displayed in the second, third, fourth, and fifth columns, respectively

ground truth. We have the ground truth of the segmentation of the WM, GM, and CSF for synthetic data from the BrainWeb, which can be directly utilized as S_2 in Eq. (50) to compute the JS index. The greater the JS value, the more similar the algorithm segmentation is to the reference segmentation.

The comparison of JS values of the 4 approaches on the 30 synthetic images with varying degrees of intensity inhomogeneities and different amounts of noise is shown in Fig. 5. The box plot of the JS values for the GM and WM generated by our approach (MICO and TVMICO), SPM, FSL, and FANTASM is shown in Fig. 5. In terms of segmentation accuracy and robustness, the box plot of the JS values in Fig. 5 clearly shows that MICO and TVMICO perform better than SPM, FSL, and FANTASM.

We see that the box in the box plot for the basic MICO is comparatively shorter, and there are no outliers in the JS values throughout all 30 test images. This demonstrates the basic MICO's intended robustness. The TVMICO is slightly more accurate than the regular MICO; however there are outliers in the TVMICO's JS values. The performance of TVMICO is determined by the parameter λ in Eq. (21), which must be modified in some circumstances. We set $\lambda = 0.01$ for all 30 test images in this experiment and observed that the results are generally favorable, except for one scenario, which results in outliers in the box plot in Fig. 5. In comparison, the basic MICO is more robust and has more steady performance than TVMICO, while the latter is somewhat more accurate in most circumstances. In reality, the difference in segmentation accuracy between MICO and TVMICO is not substantial for images with reasonable noise levels. When robustness is a priority and the image noise level is low, we recommend using

Fig. 5 Quantitative evaluation of TVMICO (with $\lambda = 0.01$), MICO, SPM, FSL, and FANTASM segmentation outcomes for 30 images using Jaccard similarity index with ground truth



the basic MICO. Otherwise, TVMICO has better segmentation and bias-corrected results; see Figs. 6, 7 and 8. When the noise level is high, the results obtained by our proposed TVMICO outperform the basic MICO. Figures 6, 7 and 8 show the progress in segmentation and bias field correction in zoomed images. We added various intensity inhomogeneities and noise to the images generated from the atrophy simulator to assess the performance of our technique in the presence of intensity inhomogeneities and noise. In this experimental result, we set $\lambda = 0.008$ in the TVMICO formulation in Eq. (23). We observed that the performance of the TVMICO formulation is affected by the parameter λ as well as certain extra parameters in the numerical method for energy minimization with respect to the membership functions. More information on the implementation and validations of the 4D MICO formulation in Eq. (46) and its modified variants will be published in a subsequent publication as an extension of this study. In the case of fully automatic

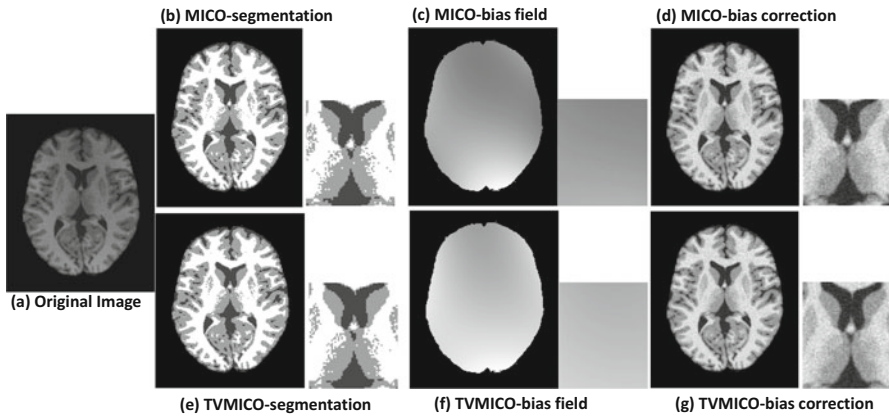


Fig. 6 On BrainWeb data, we obtained results for tissue segmentation and bias correction using our proposed MICO and TVMICO. Figure (a) shows the original image, (b) and (e) show the segmentation results, (c) and (f) show bias fields, and (d) and (g) provide bias field corrected images

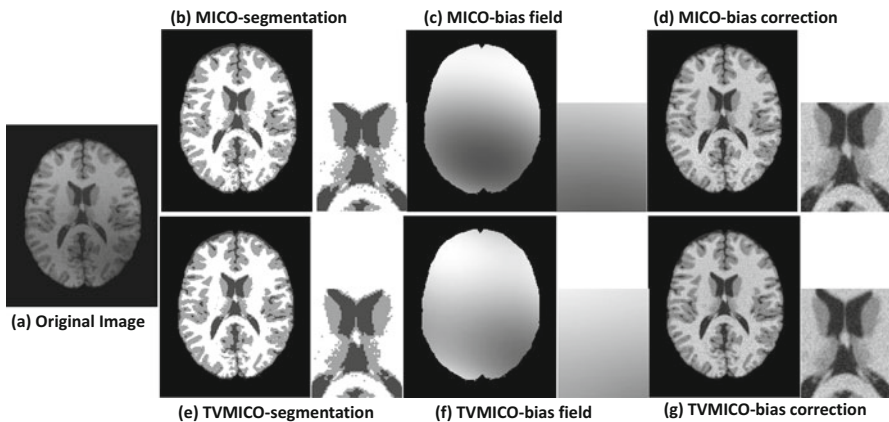


Fig. 7 On BrainWeb data, we obtained results for tissue segmentation and bias correction using our proposed MICO and TVMICO. Figure (a) shows the original image, (b) and (e) show the segmentation results, (c) and (f) show bias fields, and (d) and (g) provide bias field corrected images

segmentation of huge data sets, robustness and stability of performance are critical. The basic MICO is preferred to TVMICO because of its robustness and stability.

MICO's estimated bias field \hat{b} can be used to compute the bias field corrected image I/\hat{b} . We examined the performance of MICO's bias field correction and compared it to two well-known bias field correction methods, namely, the N3 approach described in Sled et al. (1998) and the entropy minimization method proposed in Likar et al. (2001).

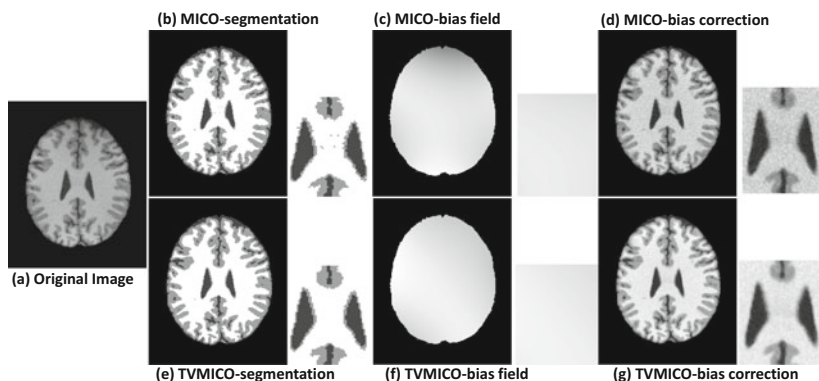


Fig. 8 On BrainWeb data, we obtained results for tissue segmentation and bias correction using our proposed MICO and TVMICO. Figure (a) shows the original image, (b) and (e) show the segmentation results, (c) and (f) show bias fields, and (d) and (g) provide bias field corrected images

The performance of bias field correction can be measured by calculating the coefficient of variations (CV) and coefficient of joint variation from the intensity inhomogeneities of the bias field corrected images (CJV).

The CV is defined for each tissue T (WM or GM) by the following:

$$CV(T) = \frac{\sigma(T)}{\mu(T)},$$

where $\sigma(T)$ and $\mu(T)$ denote the standard deviation and mean of the intensities in the tissue T , respectively. The CJV is defined as follows:

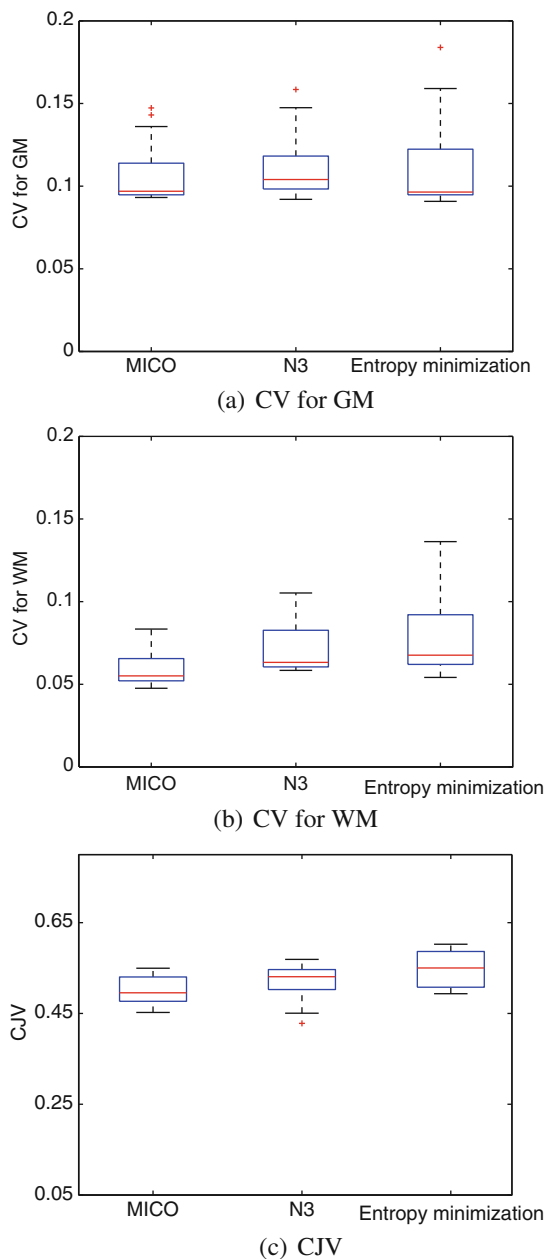
$$CJV = \frac{\sigma(WM) + \sigma(GM)}{|\mu(WM) - \mu(GM)|}.$$

The CV and CJV of the bias field corrected images are used to evaluate the performance of bias field correction, with lower CV and CJV values indicating better bias field correction outcomes.

We used our approach, as well as the N3 and entropy minimization algorithms included in the MIPAV software, to analyze 15 pictures from 3 Tesla MRI scanners. MIPAV software is freely accessible at <http://mipav.cit.nih.gov/>. The CV and CJV values of the 3 tested techniques for the 15 images are displayed in Fig. 9, demonstrating that our method outperforms the N3 and entropy minimization methods.

It is worth noting that the GM and WM are the ground truth in the conventional definition of CV and CJV in the literature on bias field correction (Vovk et al. 2007). We used an approximation of the ground truth of GM/WM by the intersection of the segmented GM/WM obtained by applying the K-means algorithm to the bias-

Fig. 9 In terms of CV and CJV, we compared the performance of our method MICO, the N3 algorithm, and the entropy reduction method on 15 images from 3T MR scanners (a) CV for GM. (b) CV for WM. (c) CJV



corrected images by the three compared bias field correction methods: our method and the well-known N3 method (Sled et al. 1998) and the entropy minimization method (Likar et al. 2001).

As previously stated, we only employed 20 polynomials as basis functions in estimating the bias field. However, MICO's bias field rectification capabilities can be improved by adding additional and various types of basis functions, such as B-spline functions, to expand the spectrum of bias fields represented as linear combinations of the basis functions. It would allow MICO to be used to very high-field MRI (e.g., 7-Tesla) and other medical images with extreme intensity inhomogeneities.

Conclusion

First, in this chapter, we introduced an energy minimization-based technique named multiplicative intrinsic component optimization (MICO) for bias field estimation and segmentation of MR images. Second, we expanded the MICO formulation by using total variation (TV) as a convex regularization. Furthermore, we implemented the alternating direction method of multipliers (ADMM) for the TV-based MICO model, which can solve the model efficiently and guarantee its convergence. We computed the bias field using matrix and vector calculus, and we utilized matrix analysis to establish the numerical stability of the computation for bias field optimization. The evaluation and comparison of our technique with other methods on synthetic and actual MR data indicate its robustness, accuracy, and efficiency. Our approach has been applied effectively to 1.5T and 3T MR images with promising outcomes. In comparison to other popular software, the results of the experiments reveal that our technique provides essential improvements in terms of segmentation accuracy and robustness. We also demonstrated that the MICO formulation can be naturally extended to 3D/4D segmentation with spatial/spatiotemporal regularization, producing encouraging results.

References

- Ahmed, M., Yamany, S., Mohamed, N., Farag, A., Moriarty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Trans. Med. Imaging* **21**(3), 193–199 (2002)
- Axel, L., Costantini, J., Listerud, J.: Intensity correction in surface-coil MR imaging. *Am. J. Radiol.* **148**(2), 418–420 (1987)
- Barrow, H., Tenenbaum, J.: Recovering intrinsic scene characteristics from images. In: Hanson, A., Riseman, E. (eds.) *Computer Vision Systems*, pp. 3–26. Academic, Orlando (1978)
- Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)
- Chen, K.: *Introduction to Variational Image-Processing Models and Applications* (2013)
- Chen, Y., Ye, X.: Projection onto a simplex, arXiv preprint arXiv:1101.6081 (2011)
- Condon, B.R., Patterson, J., Wyper, D.: Image nonuniformity in magnetic resonance imaging: its magnitude and methods for its correction. *Br. J. Radiol.* **60**(1), 83–87 (1987)
- Dawant, B., Zijdenbos, A., Margolin, R.: Correction of intensity variations in MR images for computer-aided tissues classification. *IEEE Trans. Med. Imaging* **12**(4), 770–781 (1993)
- Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**(1), 17–40 (1976)
- Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical*

- Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique **9**(R2) 41–76 (1975)
- Goldstein, T., Osher, S.: The split bregman method for L1 regularized problems, UCLA CAM Report 08-29 (2009)
- Goldstein, T., Osher, S.: The split bregman method for l1-regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
- Golub, G., Loan, C.V.: *Matrix Computations*, 3rd edn. The Johns Hopkins University Press, Baltimore/London (1996)
- Guillemaud, R., Brady, M.: Estimating the bias field of MR images. *IEEE Trans. Med. Imaging* **16**(3), 238–251 (1997)
- He, Y., Hussaini, M.Y., Ma, J., Shafei, B., Steidl, G.: A new fuzzy c-means method with total variation regularization for segmentation of images with noisy and incomplete data. *Pattern Recogn.* **45**(9), 3463–3471 (2012)
- Horn, R., Johnson, C.: *Matrix Analysis*. Cambridge University Press, Cambridge (1985)
- Johnston, B., Atkins, M.S., Mackiewicz, B., Anderson, M.: Segmentation of multiple sclerosis lesions in intensity corrected multispectral MRI. *IEEE Trans. Med. Imaging* **15**(2), 154–169 (1996)
- Juntu, J., Sijbers, J., Van Dyck, D., Gielen, J.: Bias field correction for MRI images. In: *Computer Recognition Systems*, pp. 543–551. Springer, Berlin/Heidelberg (2005)
- Kimmel, R., Elad, M., Shaked, D., Keshet, R., Sobel, I.: A variational framework for retinex. *Int. J. Comput. Vis.* **52**(1), 7–23 (2003)
- Leemput, V., Maes, K., Vandermeulen, D., Suetens, P.: Automated model-based bias field correction of MR images of the brain. *IEEE Trans. Med. Imag.* **18**(10), 885–896 (1999)
- Li, C., Kao, C., Gore, J.C., Ding, Z.: Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Imag. Proc.* **17**(10), 1940–1949 (2008)
- Li, C., Huang, R., Ding, Z., Gatenby, C., Metaxas, D., Gore, J.: A variational level set approach to segmentation and bias correction of medical images with intensity inhomogeneity. In: *Proceedings of Medical Image Computing and Computer Aided Intervention (MICCAI)*. LNCS, vol. 5242, Part II, pp. 1083–1091 (2008)
- Li, C., Xu, C., Anderson, A.W., Gore, J.C.: MRI tissue classification and bias field estimation based on coherent local intensity clustering: a unified energy minimization framework. In: *International Conference on Information Processing in Medical Imaging*, pp. 288–299. Springer (2009)
- Li, F., Ng, M.K., Li, C.: Variational fuzzy Mumford–Shah model for image segmentation. *SIAM J. Appl. Math.* **70**(7), 2750–2770 (2010)
- Li, C., Gore, J.C., Davatzikos, C.: Multiplicative intrinsic component optimization (mico) for MRI bias field estimation and tissue segmentation. *Mag. Resonan. Imaging* **32**(7), 913–923 (2014)
- Li, F., Osher, S., Qin, J., Yan, M.: A multiphase image segmentation based on fuzzy membership functions and l1-norm fidelity. *J. Sci. Comput.* **69**(1), 82–106 (2016)
- Likar, B., Viergever, M., Pernus, F.: Retrospective correction of MR intensity inhomogeneity by information minimization. *IEEE Trans. Med. Imaging* **20**(12), 1398–1410 (2001)
- Lions, P., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16**(6), 964–979 (1979)
- Liu, J., Huang, T.-Z., Selesnick, I.W., Lv, X.-G., Chen, P.-Y.: Image restoration using total variation with overlapping group sparsity. *Inf. Sci.* **295**, 232–246 (2015)
- Liu, Z., Wali, S., Duan, Y., Chang, H., Wu, C., Tai, X.-C.: Proximal ADMM for Euler’s elastica based image decomposition model. *Numer. Math. Theory Methods Appl.* **12**(2), 370–402 (2019)
- McVeigh, E.R., Bronskil, M.J., Henkelman, R.M.: Phase and sensitivity of receiver coils in magnetic resonance imaging. *Med. Phys.* **13**(6), 806–814 (1986)
- Narayana, P.A., Brey, W.W., Kulkarni, M.V., Sivenpiper, C.L.: Compensation for surface coil sensitivity variation in magnetic resonance imaging. *Magn. Reson. Imaging* **6**(3), 271–274 (1988)
- Pham, D.: Spatial models for fuzzy clustering. *Comput. Vis. Image Underst.* **84**(2), 285–297 (2001)

- Pham, D., Prince, J.: Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Trans. Med. Imaging* **18**(9), 737–752 (1999)
- Powell, M.J.D.: *Approximation Theory and Methods*. Cambridge University Press, Cambridge (1981)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1–4), 259–268 (1992)
- Salvado, O., Hillenbrand, C., Wilson, D.: Correction of intensity inhomogeneity in MR images of vascular disease. In: *EMBS'05*, pp. 4302–4305. IEEE, Shanghai (2005)
- Shattuck, D.W., Sandor-Leahy, S.R., Schaper, K.A., Rottenberg, D.A., Leahy, R.M.: Magnetic resonance image tissue classification using a partial volume model. *Neuroimage* **13**, 856–876 (2001)
- Simmons, A., Tofts, P.S., Barker, G.J., Arridge, S.R.: Sources of intensity nonuniformity in spin echo images at 1.5t. *Magn. Reson. Med.* **32**(1), 121–128 (1991)
- Sled, J., Zijdenbos, A., Evans, A.: A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Trans. Med. Imaging* **17**(1), 87–97 (1998)
- Stockman, G., Shapiro, L.G.: *Computer Vision*. Prentice Hall, Upper Saddle River (2001)
- Styner, M., Brechbuhler, C., Szekely, G., Gerig, G.: Parametric estimate of intensity inhomogeneities applied to MRI. *IEEE Trans. Med. Imaging* **19**(3), 153–165 (2000)
- Tappen, M., Freeman, W., Adelson, E.: Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(9), 1459–1472 (2005)
- Tincher, M., Meyer, C.R., Gupta, R., Williams, D.M.: Polynomial modeling and reduction of RF body coil spatial inhomogeneity in MRI. *IEEE Trans. Med. Imaging* **12**(2), 361–365 (1993)
- Tu, X., Gao, J., Zhu, C., Cheng, J.-Z., Ma, Z., Dai, X., Xie, M.: MR image segmentation and bias field estimation based on coherent local intensity clustering with total variation regularization. *Med. Biol. Eng. Comput.* **54**(12), 1807–1818 (2016)
- Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C.: N4itk: improved n3 bias correction. *IEEE Trans. Med. Imaging* **29**(6), 1310–1320 (2010)
- Tustison, N., Avants, B., Cook, P., Zheng, Y., Egan, A., Yushkevich, P., Gee, J.: N4itk: improved n3 bias correction. *IEEE Trans. Med. Imaging* **29**(6), 1310–1320 (2010)
- Vovk, U., Pernus, F., Likar, B.: A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Trans. Med. Imaging* **26**(3), 405–421 (2007)
- Vovk, U., Pernus, F., Likar, B.: A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Trans. Med. Imaging* **26**(3), 405–421 (2007)
- Wali, S., Zhang, H., Chang, H., Wu, C.: A new adaptive boosting total generalized variation (TGV) technique for image denoising and inpainting. *J. Vis. Commun. Image Represent.* **59**, 39–51 (2019a)
- Wali, S., Shakoor, A., Basit, A., Xie, L., Huang, C., Li, C.: An efficient method for Euler's elastica based image deconvolution. *IEEE Access* **7**, 61226–61239 (2019b)
- Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imaging Sci.* **1**(3), 248–272 (2008)
- Weiss, Y.: Deriving intrinsic images from image sequences. In: *Proceedings of 8th International Conference on Computer Vision (ICCV)*, vol. II, pp. 68–75 (2001)
- Wells, W., Grimson, E., Kikinis, R., Jolesz, F.: Adaptive segmentation of MRI data. *IEEE Trans. Med. Imaging* **15**(4), 429–442 (1996)
- Wicks, D.A.G., Barker, G.J., Tofts, P.S.: Correction of intensity nonuniformity in MR images of any orientation. *Magn. Reson. Imag.* **11**(2), 183–196 (1993)
- Zheng, X., Lei, Q., Yao, R., Gong, Y., Yin, Q.: Image segmentation based on adaptive k-means algorithm. *EURASIP J. Image Video Process.* **2018**(1), 1–10 (2018)



Data-Informed Regularization for Inverse and Imaging Problems

35

Jonathan Wittmer and Tan Bui-Thanh

Contents

Introduction	1236
A Data-Informed Regularization (DI) Approach	1238
Data-Informed Regularization Derivation	1238
A Statistical Data-Informed (DI) Inverse Framework	1245
Properties of the DI Regularization Approach	1250
Applications to Imaging Problems	1256
Image Deblurring	1256
Image Denoising	1265
X-Ray Tomography	1265
Conclusions	1269
References	1271

Submitted to the editors DATE.

This work was partially funded by the National Science Foundation awards NSF-1808576 and NSF-CAREER-1845799; by the Defense Threat Reduction Agency award DTRA-M1802962; by the Department of Energy award DE-SC0018147; by KAUST; by 2018 ConTex award; and by 2018 UT-Portugal CoLab award. The authors are grateful to the supports.

J. Wittmer (✉)

Department of Aerospace Engineering and Engineering Mechanics, UT Austin, Austin, TX, USA
e-mail: jonathan.wittmer@utexas.edu

T. Bui-Thanh (✉)

Department of Aerospace Engineering and Engineering Mechanics, The Oden Institute for Computational Engineering and Sciences, UT Austin, Austin, TX, USA
e-mail: tanbui@ices.utexas.edu

Abstract

This chapter presents a new regularization method for inverse and imaging problems, called data-informed (DI) regularization, that implicitly avoids regularizing the data-informed directions. Our approach is inspired by and has a rigorous root in disintegration theory. We shall, however, present an elementary and constructive path using the classical truncated SVD and Tikhonov regularization methods. Deterministic and statistical properties of the DI approach are rigorously discussed, and numerical results for image deblurring, image denoising, and X-ray tomography are presented to verify our findings.

Keywords

Inverse problems · Imaging · Tikhonov regularization · Truncated SVD · Data-informed regularization

Introduction

Regularization is often employed to facilitate the well-posedness of inverse (and imaging) problems. An inverse solution is thus a trade-off between the data misfit and the regularization. Due to noise and limited availability, available data typically informs limited directions in the parameter space where the inverse solution resides. A desired regularization, we argue, should minimally interfere with these data-informed directions. However, most regularization techniques regularize all parameter directions, including the data-informed ones, thus polluting the resulting inverse solution. Finding a “right” regularization remains an open problem in inverse and imaging communities.

Over the past decades, many different regularization approaches have been proposed including Tikhonov regularization (Tikhonov and Arsenin 1977), total variation regularization (Rudin et al. 1992; Beck and Teboulle 2009), and non-convex regularization strategies (Ramirez-Giraldo et al. 2011; Babacan et al. 2009; Nikolova 2005), to name a few. In the Bayesian statistical framework, these regularization strategies can be encoded as prior distributions for the inverse solutions. Perhaps the simplest and the most popular regularization strategy is the Tikhonov approach, which corresponds to a Gaussian prior in the Bayesian framework (Stuart 2010). One shortcoming of the Tikhonov prior is that it tends to be a smoothing prior (Mueller and Siltanen 2012), highly diffusing discontinuities. The total variation (TV) prior, which induces an anisotropic diffusion, seeks to minimally penalize discontinuities in the inverse solution (Rudin et al. 1992; Beck and Teboulle 2009; Mueller and Siltanen 2012). However, the TV prior is known to produce a staircasing effect due to non-differentiability of the TV functional

(Nikolova 2004). Because of the lack of differentiability, smooth approximations to the TV prior can be used, or more sophisticated optimization methods must be employed (Mueller and Siltanen 2012; Goldstein and Osher 2009). Similarly, inverse formulations using non-convex priors also require advanced optimization methods such as alternating direction method of multipliers (ADMM) (Chartr and Wohlberg 2013; Boley 2013; Boyd et al. 2010) or iteratively reweighted least-squares (IRLS) (Chartr and Yin 2008) to find the inverse solution.

This chapter presents a new regularization method for inverse and imaging problems, called data-informed (DI) regularization, that implicitly avoids regularizing the data-informed directions. *Our approach is inspired by and has a rigorous root in the disintegration theory.* We have, however, discovered a constructive path to understand our approach using the classical truncated SVD and Tikhonov regularization methods. The goal of this chapter is to share this constructive path to the DI approach and presents advantages/disadvantages of the DI approach on several existing applications in imaging. As will be shown theoretically and numerically, the DI approach avoids polluting the data-informed directions while regularizing the less data-informed ones.

Compared to existing approaches, our method has many distinct and advantageous features: (1) it automatically determines the directions equally informed by the data and any Tikhonov regularization while leaving the most informative directions untouched. In fact, we will show that, similar to the balanced truncation idea in control theory (see, e.g., Gugercin and Antoulas 2004; Antoulas 2005 and the references therein), this is done implicitly by seeking directions in parameter space that balance the information from regularization and data and removing the regularization on them. (2) We will show that our approach has an intuitive statistical interpretation, namely, it transforms both the data distribution (i.e., the likelihood) and prior distribution (induced by Tikhonov regularization) to the same Gaussian distribution whose covariance matrix is diagonal and the diagonal elements are exactly the singular values of a composition of the prior covariance matrix, the forward map, and the noise covariance matrix. (3) Though constructively derived and its insights obtained from the truncated singular value decomposition (SVD), the inverse solution resulting from our approach does not necessarily require the computation of an SVD, which may not be feasible for large-scale applications. We will present a nested matrix-free approach to obtain an approximate inverse solution. Our approach is thus more expensive than Tikhonov regularization when truncated SVD is not affordable. (4) By construction, features in DI solutions, dictated by the data-informed directions, are insensitive to the regularization parameter. For many inverse and imaging problems, these features dominate the solution, and thus the inverse solution resulting from our regularization technique is robust with respect to regularization parameter values. These findings will be demonstrated and supported by various numerical results from deblurring, denoising, and X-ray tomography problems.

A Data-Informed Regularization (DI) Approach

Data-Informed Regularization Derivation

In this section we review the key ideas behind regularization by truncation using the singular value decomposition (SVD). This provides the basic insights into the data-informed regularization technique. A statistical interpretation of the data-informed inverse framework will be discussed in section “A Statistical Data-Informed (DI) Inverse Framework”. To begin, let us consider a linear inverse problem to determine $\mathbf{x} \in \mathbb{R}^p$ given

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{d \times p}$, $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \lambda^2 \mathbf{I})$, $\mathbf{I} \in \mathbb{R}^{d \times d}$, and $\mathbf{y} \in \mathbb{R}^d$. In the following, the identity matrix \mathbf{I} may have different size at different places and the actual size should be clear from the context. The simplest approach to attempt to solve this inverse problem is perhaps the least-squares approach:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2, \tag{2}$$

where $\|\cdot\|$ denotes the standard Euclidean norm. Figure 1a plots the exact synthetic solution (black curve) against the least-squares solution (red curve) for a deconvolution problem with $d = p = 101$ and $\lambda = 0.05$. As can be seen, the least-squares solution blows up (or is unstable) due to the ill-conditioning of $\mathbf{A}^T \mathbf{A}$, which is not surprising since the inverse problem is (typically) ill-posed.

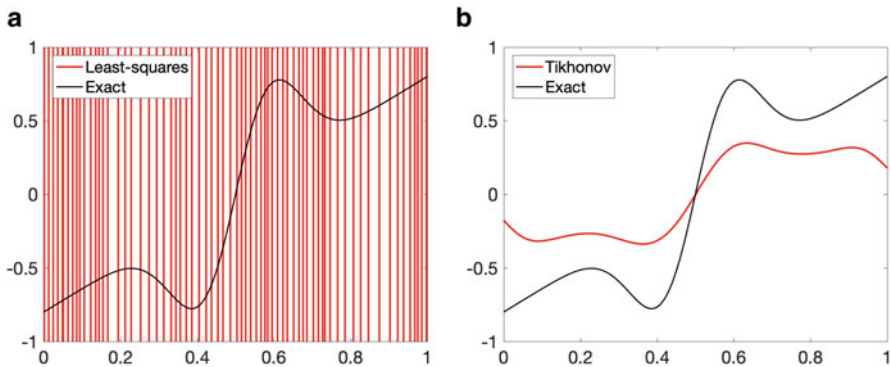


Fig. 1 Deconvolution using (a) the least-squares approach and (b) a Tikhonov regularization with regularization parameter $\alpha = 1$ and $\mathbf{x}_0 = \mathbf{0}$

To overcome the ill-posedness, a classical Tikhonov regularization approach casts the above inverse problem into

$$\min_x \frac{1}{2} \|Ax - y\|^2 + \frac{\alpha}{2} \|x - x_0\|^2,$$

where x_0 is given. A Tikhonov solution is presented in Fig. 1b for $\alpha = 1$ and $x_0=0$. Though this approach stabilizes the solution, it also smooths out the solution everywhere.

Regularization by truncation does not require an explicit introduction of a regularization term as in Tikhonov regularization. For example, the truncated SVD starts with the SVD decomposition of A and then truncates all the singular vectors U_j and V_j corresponding to sufficiently small singular values, i.e.,

$$\begin{aligned}
 A &= U \Sigma V^T \\
 &= \begin{pmatrix} U_1 & U_n & U_{n+1} & U_d \\ \left| \right. & \left| \right. & \left| \right. & \left| \right. \\ \left| \right. & \dots & \left| \right. & \left| \right. \\ \left| \right. & & \left| \right. & \left| \right. \\ \left| \right. & & \left| \right. & \left| \right. \end{pmatrix} \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix} \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{pmatrix} \begin{pmatrix} V_1^T \\ V_n^T \\ V_{n+1}^T \\ V_p^T \end{pmatrix} \\
 &= U^n \Sigma^n (V^n)^T,
 \end{aligned}$$

where $U^n := [U_1, \dots, U_n]$ (the first n columns of U corresponding to n nonzero singular values (This rank- n decomposition is often known as the reduced SVD.)), $\Sigma^n := \text{diag}[\sigma_1, \dots, \sigma_n]$ ($\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$), $n \leq \min\{d, p\}$, and $V^n := [V_1, \dots, V_n]$ (the first n columns of V corresponding to n nonzero singular values). U^n forms an orthonormal basis for the column space of A , and V^n forms an orthonormal basis for the row space of A . The solution of (2) with this rank- n truncation using the pseudo-inverse A^\dagger reads

$$x_{\text{SVD}}^n = A^\dagger y = V^n (\Sigma^n)^{-1} (U^n)^T y = \sum_{i=1}^n \frac{U_i^T y}{\sigma_i} V_i.$$

However, to avoid potentially dividing by very small singular values, a truncated SVD (TSVD) (Hansen 1990) with rank less than n is typically used. The rank- r TSVD solution using only the r largest singular values (with $r \leq n$) can be written as

$$\mathbf{x}_{\text{TSVD}}^r := \sum_{i=1}^r \frac{\mathbf{U}_i^T \mathbf{y}}{\sigma_i} \mathbf{V}_i, \tag{3}$$

Figure 2 applies the TSVD approach to the deconvolution problem and compares the results with the Tikhonov regularization. As can be seen, TSVD solutions are stable and do not seem to over-regularize the solution. However, as r increases, TSVD solutions tend to be more oscillatory (more unstable). How can this behavior of TSVD be explained?

The answer lies on the fact that the j th column of \mathbf{A} is the observational vector when the parameter \mathbf{x} is the j th canonical basis vector in \mathbb{R}^p . Thus the range space (column space) of \mathbf{A} can be understood as the *observable subspace* in \mathbb{R}^d . Within this observable subspace, we say that the subspace spanned by \mathbf{U}_j , i.e., $\text{span}\{\mathbf{U}_j\}$, is more observable than the subspace spanned by \mathbf{U}_i , i.e., $\text{span}\{\mathbf{U}_i\}$, when $j < i$. Equivalently, $\text{span}\{\mathbf{U}_i\}$ is less observable than $\text{span}\{\mathbf{U}_j\}$. With this (relative) definition, $\text{span}\{\mathbf{U}_1\}$ is most observable, while $\text{span}\{\mathbf{U}_n\}$ is least observable. Clearly $j < i$ implies $1/\sigma_j \geq 1/\sigma_i$. Consequently, in the TSVD solution (3), less

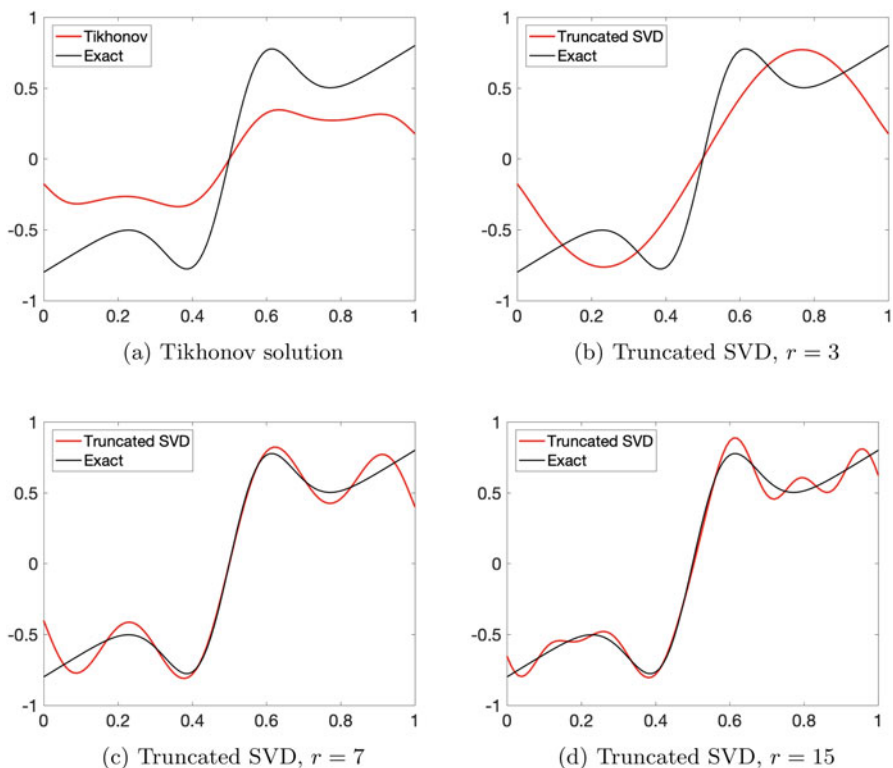


Fig. 2 Deconvolution using (a) a Tikhonov regularization with regularization parameter $\alpha = 1$ and $\mathbf{x}_0 = \mathbf{0}$; (b) truncated SVD with $r = 3$; (c) truncated SVD with $r = 10$; and (d) truncated SVD with $r = 15$

observable modes (directions) U_i tend to promote oscillation and/or instability. In particular, U_n is the least stable (or most oscillatory) direction. As r increases, more oscillatory directions are amplified and added to the TSVD solution. This is exactly what Fig. 2 shows.

Given r (to be chosen based on the noise level ε), we define the **data-more-informed** parameter subspace as the row subspace spanned by $\{V_1, \dots, V_r\}$ (corresponding to the **observable** subspace spanned by $\{U_1, \dots, U_r\}$). Similarly, we define the **data-less-informed** parameter subspace as the row subspace spanned by $\{V_{r+1}, \dots, V_n\}$ (corresponding to the less observable subspace spanned by $\{U_{r+1}, \dots, U_n\}$). For brevity, we use **data-informed** and **data-uninformed** instead of **data-more-informed** and **data-less-informed**, though the latter is more precise as the definitions are relative. Additionally, we use *modes and directions* interchangeably to refer to the corresponding singular vectors V_j themselves, rather than the subspaces spanned by them.

The TSVD solution (3) clearly resides in the **data-informed** parameter subspace. The question is *where to truncate so that the solution is data-informed?* The result of Fig. 2 and its discussion suggest that r should be neither too large nor too small. That is, we seek to find r such that the solution captures information informed by the data while being least oscillatory. Clearly r is problem-dependent. For example, inspired by the Morozov’s discrepancy principle (Morozov 1966), if the noise level ε is given (or can be estimated), r can be chosen such that $\sigma_j \geq \varepsilon$ for $j \leq r$.

A closer look at the TSVD solution (3) shows that the truncated SVD approach zeroes out the data-uninformed modes V_j for $j \geq r + 1$. We next show that this is equivalent to infinitely regularizing data-uninformed directions. To see this, let us now consider a regularization scheme where the data-uninformed modes are penalized infinitely, i.e., formally

$$\min \frac{1}{2} \|Ax - y\|^2 + \frac{1}{2} \|L(x - x_0)\|^2, \tag{4}$$

where

$$\begin{aligned} L^T L &:= \infty [I - V^r (V^r)^T] = \infty (V^r)^\perp ((V^r)^\perp)^T \\ &= [V^r, (V^r)^\perp] \begin{bmatrix} 0 & 0 \\ 0 & \infty I \end{bmatrix} [V^r, (V^r)^\perp]^T, \end{aligned}$$

and $[I - V^r (V^r)^T]$ is the orthogonal projection onto the data-uninformed subspace spanned by $\{V_j\}_{j=r+1}^d$. Here, multiplication by infinity is understood in the usual limit sense, e.g., $\infty I := \lim_{\alpha \rightarrow \infty} \alpha I$. Thus, regularization—an infinite amount in this case—is only added in data-uninformed directions. The solution of (4) is formally given by

$$\begin{aligned}
\mathbf{x}_{Inf} &= \left\{ \mathbf{A}^T \mathbf{A} + \infty \left(\mathbf{I} - \mathbf{V}^r (\mathbf{V}^r)^T \right) \right\}^{-1} \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right) \\
&= \left\{ \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right] \left(\left[\begin{array}{c|c} (\boldsymbol{\Sigma}^r)^2 & 0 \\ \hline 0 & \mathbf{D}^2 \end{array} \right] + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \infty \mathbf{I} \end{array} \right] \right) \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right]^T \right\}^{-1} \mathbf{A}^T \mathbf{y} \\
&= \mathbf{V}^r (\boldsymbol{\Sigma}^r)^{-2} (\mathbf{V}^r)^T \mathbf{A}^T \mathbf{y} = \mathbf{V}^r (\boldsymbol{\Sigma}^r)^{-1} (\mathbf{U}^r)^T \mathbf{y} =: \mathbf{x}_{TSVD}^r,
\end{aligned}$$

where $(\mathbf{V}^r)^\perp$ is the orthogonal complement of \mathbf{V}^r in \mathbb{R}^p , $\boldsymbol{\Sigma}^r := \text{diag}[\sigma_1, \dots, \sigma_r]$, and $\mathbf{D} := \text{diag}[\sigma_{r+1}, \dots, \sigma_p]$. The second equality clearly shows that the regularization scheme adds infinity to all singular values that correspond to data-uninformed modes. The last equality proves that infinite regularization on data-uninformed parameter subspace is the same as the TSVD approach.

The beauty of the TSVD approach is that it avoids putting any regularization on data-informed parameter directions, and hence avoids polluting inverse solutions in these directions, while annihilating data-uninformed directions. However, it is often the case that there is no clear-cut between the data-informed and data-uninformed ones (i.e., $\sigma_k = 0$ for $k \geq r + 1$) but gradual (sometimes exponential) decay of the singular values of \mathbf{A} . In that case, completely removing less data-informed directions may not be ideal, as they may still contain valuable parameter information encoded in the data. Instead, we may want to impose finite regularization in the data-uninformed directions, i.e.,

$$\min \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \frac{1}{2} \|\mathbf{L}(\mathbf{x} - \mathbf{x}_0)\|^2, \quad (5)$$

where

$$\begin{aligned}
L^T L &:= \alpha \left(\mathbf{I} - \mathbf{V}^r (\mathbf{V}^r)^T \right) = \alpha (\mathbf{V}^r)^\perp \left((\mathbf{V}^r)^\perp \right)^T \\
&= \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right] \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \alpha \mathbf{I} \end{array} \right] \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right]^T.
\end{aligned}$$

Let us call this approach the **data-informed (DI) regularization method**. The inverse solution in this case reads

$$\begin{aligned}
\mathbf{x}_{DI} &= \left\{ \mathbf{A}^T \mathbf{A} + \alpha \left(\mathbf{I} - \mathbf{V}^r (\mathbf{V}^r)^T \right) \right\}^{-1} \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right) \\
&= \left\{ \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right] \left(\left[\begin{array}{c|c} (\boldsymbol{\Sigma}^r)^2 & 0 \\ \hline 0 & \mathbf{D}^2 \end{array} \right] + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \alpha \mathbf{I} \end{array} \right] \right) \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right]^T \right\}^{-1}
\end{aligned}$$

$$\begin{aligned}
& \times \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right) \\
& = \left\{ \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right] \left(\left[\begin{array}{c|c} (\boldsymbol{\Sigma}^r)^2 & 0 \\ \hline 0 & \mathbf{D}^2 \end{array} \right] + \alpha \left[\begin{array}{c|c} \mathbf{I} & 0 \\ \hline 0 & \mathbf{I} \end{array} \right] - \left[\begin{array}{c|c} \alpha \mathbf{I} & 0 \\ \hline 0 & 0 \end{array} \right] \right) \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right]^T \right\}^{-1} \\
& \times \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right).
\end{aligned}$$

The last equality suggests that the DI approach can be considered as *first applying the same* (Note that α need not be the same for all directions.) *(finite) regularization for all parameter directions and then removing regularization in the data-informed directions.*

A few observations are in order: (1) When $r = 0$, DI becomes the standard Tikhonov regularization; (2) when $\alpha \rightarrow \infty$ DI approaches the truncated SVD; and (3) when $\alpha \ll \sigma_i$ for $i \leq r$ (i.e., regularization in the data-informed modes is negligible), the Tikhonov solution

$$\begin{aligned}
\mathbf{x}_{Tikhonov} & = \left\{ \mathbf{A}^T \mathbf{A} + \alpha \mathbf{I} \right\}^{-1} \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right) \\
& = \left\{ \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right] \left(\left[\begin{array}{c|c} (\boldsymbol{\Sigma}^r)^2 & 0 \\ \hline 0 & \mathbf{D}^2 \end{array} \right] \right. \right. \\
& \quad \left. \left. + \alpha \left[\begin{array}{c|c} \mathbf{I} & 0 \\ \hline 0 & \mathbf{I} \end{array} \right] \right) \left[\mathbf{V}^r, (\mathbf{V}^r)^\perp \right]^T \right\}^{-1} \left(\mathbf{A}^T \mathbf{y} + L^T L \mathbf{x}_0 \right)
\end{aligned}$$

is close to the DI solution \mathbf{x}_{DI} as the contribution of the regularization to data-informed modes is negligible. These observations are clearly demonstrated in Fig. 3 for a 1D deconvolution with $\lambda = 0.05$ with various combinations of regularization parameter α and the number of retained data-informed modes r . An important feature of the DI technique that can be seen from this result is that for each r the DI solution is robust with the regularization parameter, that is, the solution does not alter significantly, especially for moderate-to-large regularization, while Tikhonov solution is damped out as the regularization parameter increases. The last column of Fig. 3 shows that for $r = 20$ the DI solution retains high-frequency modes which are not regularized and is thus oscillatory.

In order to gain more insights into the behavior of DI regularization, we compute the relative error between the solutions using the DI approach and the truth for a wide range of regularization parameters and a few values of r . The results are shown in Fig. 4. As can be seen, when $r = 1$, DI is essentially Tikhonov, which is not surprising as all modes in the DI solution are regularized exactly the same as Tikhonov except for the first one (lowest frequency). For $r = \{5, 10\}$, the DI solution

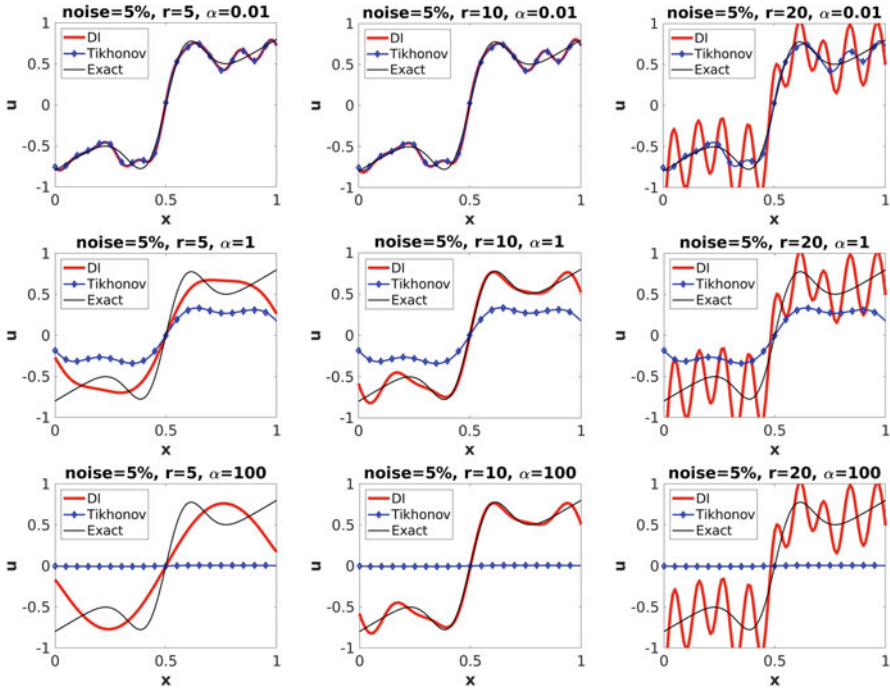


Fig. 3 Deconvolution with noise level $\lambda = 5\%$ using DI and Tikhonov regularization for various values of regularization parameter and r

behaves the same as Tikhonov for the under-regularization regime ($\alpha < 0.01$) as expected, and it outperforms Tikhonov for $\alpha > 0.01$ as the retained data-informed modes, which determine the quality of the deconvolution solution, are left untouched. For $r = 20$, the retained modes now also include high-frequency modes, and hence the DI approach is not as accurate as Tikhonov for $\alpha < 1$. For all cases with significant number of modes retained, i.e., $r > 5$, the DI solution quality is insensitive to a large range of the regularization parameter. Note that methods for choosing the regularization parameter α in practice include L-curve (Hansen and O’Leary 1993; Hansen 1992), the Morozov’s discrepancy principle (Morozov 1966), and generalized cross-validation (Golub et al. 1979). These methods are inherently computationally costly, and this can be mitigated using the DI approach as it is robust with regularization parameter.

We have used a rank- r SVD approximation to derive and gain insights into the DI approach. For large-scale problems, this low rank-decomposition could be prohibitively expensive. To lead to an alternative computational approach (see Algorithm 2) and more importantly to provide a probabilistic view point of the DI approach, let us take $r = n$ until the end of section “A Statistical Data-Informed (DI) Inverse Framework”. In this case, since $\mathbf{V}^n (\mathbf{V}^n)^T$ is the orthogonal projection into

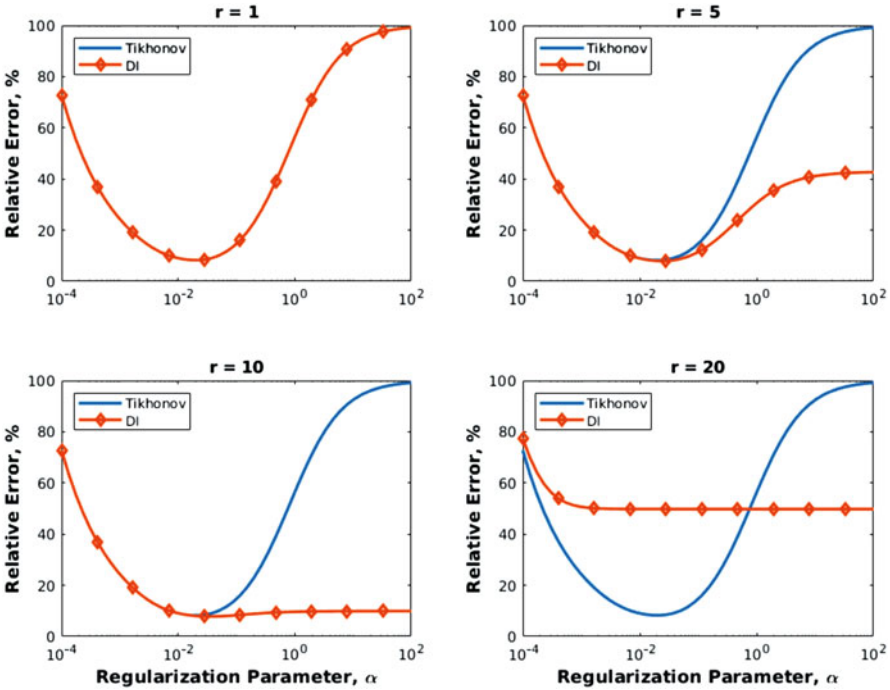


Fig. 4 Deconvolution with noise level $\lambda = 5\%$ using DI and Tikhonov regularizations for $\alpha = [10^{-4}, 10^2]$ and $r = \{1, 5, 10, 20\}$

the row space of A , i.e., $V^n (V^n)^T = A^T (AA^T)^\dagger A$, we can rewrite the inverse (optimization) problem (5) as

$$\min_x J := \frac{1}{2} \|Ax - y\|^2 + \frac{1}{2} \|L(x - x_0)\|^2, \tag{6}$$

where

$$L^T L := \alpha \left(I - A^T (AA^T)^\dagger A \right).$$

In this form, the DI regularization approach (6) not only avoids using V^n explicitly but also brings us to a statistical data-informed inverse framework in the next section.

A Statistical Data-Informed (DI) Inverse Framework

The cost function in (6) can be rewritten as

$$\exp(-J) = \frac{\exp\left(-\frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|^2\right) \times \exp\left(-\frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_0\|^2\right)}{\exp\left(-\frac{\alpha}{2} (\mathbf{Ax} - \mathbf{Ax}_0)^T (\mathbf{AA}^T)^\dagger (\mathbf{Ax} - \mathbf{Ax}_0)\right)}.$$

From a Bayesian inverse perspective (Kaipio and Somersalo 2005; Tarantola 2005; Franklin 1970; Lehtinen et al. 1989; Lasanen 2002; Stuart 2010; Piiroinen 2005), the numerator is the product of the likelihood

$$\pi_{\text{like}}(\mathbf{y}|\mathbf{x}) \propto \exp\left(-\frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|^2\right)$$

from the observational model (1) with the noise $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the Gaussian prior

$$\pi_{\text{prior}}(\mathbf{x}) \propto \exp\left(-\frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}_0\|^2\right) \tag{7}$$

with mean \mathbf{x}_0 and \mathbf{I}/α covariance matrix. In other words, the numerator is a Bayesian posterior with the aforementioned likelihood and Gaussian prior. *The key difference compared to the Bayesian approach is the denominator.*

We now show that the denominator is nothing more than the push-forward of the prior (7) via the forward map \mathbf{A} . Indeed, let $\tilde{\mathbf{y}} := \mathbf{Ax}$ be a random variable induced by the forward map \mathbf{A} . With $\mathbf{x} \sim \mathcal{N}(\mathbf{x}_0, \mathbf{I}/\alpha)$, $\tilde{\mathbf{y}}$ is also a Gaussian with mean $\tilde{\mathbf{y}}_0$ and covariance matrix \mathbb{C} where

$$\begin{aligned} \tilde{\mathbf{y}}_0 &:= \mathbb{E}_{\mathbf{x}}[\mathbf{Ax}] = \mathbf{Ax}_0 \\ \mathbb{C} &:= \mathbb{E}_{\mathbf{x}}[(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}_0)(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}_0)^T] = \mathbb{E}_{\mathbf{x}}[\mathbf{A}(\mathbf{x} - \mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)^T \mathbf{A}^T] = \frac{1}{\alpha} \mathbf{AA}^T. \end{aligned}$$

Note that it is necessary to use the pseudo-inverse for the inverse of the covariance \mathbb{C} , i.e., $\mathbb{C}^{-1} := \alpha (\mathbf{AA}^T)^\dagger$, since \mathbf{A} may not have full row rank and thus the push-forward distribution can be a degenerate Gaussian.

Remark 1. The push-forward of the prior through the parameter-to-observable map \mathbf{A} depends on \mathbf{x} . *It is through this push-forward term that the data-informed (DI) approach learns the data-informed parameter directions.* Indeed, this new approach, through the push-forward term, changes the original prior

$$\exp\left(-\frac{\alpha}{2} (\mathbf{x} - \mathbf{x}_0)^T \mathbf{I} (\mathbf{x} - \mathbf{x}_0)\right)$$

to the new one

$$\exp\left(-\frac{\alpha}{2}(\mathbf{x} - \mathbf{x}_0)^T \left[\mathbf{I} - \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^\dagger \mathbf{A} \right] (\mathbf{x} - \mathbf{x}_0)\right)$$

in such a way that the new prior leaves the data-informed directions, i.e., the row space of \mathbf{A} , untouched, and hence only regularizes data-uninformed directions. The data-informed approach accomplishes this by the push-forward of the prior via the parameter-to-observable map \mathbf{A} .

We can now define the DI posterior as

$$\pi_{\text{DI}}(\mathbf{x}|\mathbf{y}) = \frac{\pi_{\text{like}}(\mathbf{y}|\mathbf{x}) \times \pi_{\text{prior}}(\mathbf{x})}{\mathbf{A}\#\pi_{\text{prior}}(\mathbf{x})}, \quad (8)$$

where $\mathbf{A}\#\pi_{\text{prior}}(\mathbf{x})$ denotes the push-forward of $\pi_{\text{prior}}(\mathbf{x})$ via the parameter-to-observable map \mathbf{A} .

We have constructively derived the DI approach by modifying the truncated SVD method and Gaussian prior with scaled-identity covariance matrix. In practice, the prior can be more informative about the correlations among components of \mathbf{x} and in that case the covariance matrix is no longer an identity matrix. Let us denote by $\pi_{\text{prior}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}_0, \mathbf{\Gamma}/\alpha)$ the Gaussian prior with covariance matrix $\mathbf{\Gamma}/\alpha$. Let us also consider a more general data distribution where, for a given parameter \mathbf{x} , the data is distributed by the Gaussian $\mathcal{N}(\mathbf{A}\mathbf{x}, \mathbf{\Lambda})$. In order to use most of the above results, let us whiten both the parameter and observations. In particular, $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{y}$ is the whitened observations (inducing $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}$ as the new parameter-to-observable forward map), and $\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{x}$ is the prior-whitened parameter. (Here, the square roots for $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are understood in the broader sense including: (1) if $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are diagonal matrices, the square roots are simply diagonal matrices with square roots of the diagonal elements; (2) if $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ are not diagonal matrices, these square roots are understood in the spectral decomposition sense. For example: let $\mathbf{\Gamma} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ be the spectral decomposition of $\mathbf{\Gamma}$, then $\mathbf{\Gamma}^{1/2} := \mathbf{V}\mathbf{\Sigma}^{1/2}\mathbf{V}^T$. Note that this is meaningful as we assume the corresponding Gaussian distribution is non-degenerate and hence $\mathbf{\Sigma}$ is the diagonal matrix with positive diagonal elements; and (3) if Cholesky-type decomposition is available, i.e., $\mathbf{\Gamma} = \mathbf{L}\mathbf{L}^T$ (\mathbf{L} is not necessarily a Cholesky factorization), then $\mathbf{\Gamma}^{1/2} = \mathbf{L}$, and we simply add the “transpose” operator at appropriate places for one of the square roots.) The push-forward of the prior via $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}$ now reads (Note that using the modified forward map $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}$, though making the presentation clearer and constructive, is not necessary as using the original map \mathbf{A} yields the same result.)

$$\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\#\pi_{\text{prior}}(\mathbf{x}) = \mathcal{N}\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\mathbf{x}_0, \frac{1}{\alpha}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{\Lambda}^{-\frac{1}{2}}\right), \quad (9)$$

The DI posterior (8) with whitened parameter, whitened observations, and induced parameter-to-observable map now becomes

$$\begin{aligned} \pi_{\text{DI}}(\mathbf{x}|\mathbf{y}) &= \frac{\pi_{\text{like}}(\mathbf{y}|\mathbf{x}) \times \pi_{\text{prior}}(\mathbf{x})}{\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \# \pi_{\text{prior}}(\mathbf{x})} \\ &\propto \frac{\exp\left(-\frac{1}{2} \|\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{A}^{-\frac{1}{2}} \mathbf{y}\|^2\right) \times \exp\left(-\frac{\alpha}{2} \|\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x}_0\|^2\right)}{\exp\left(-\frac{\alpha}{2} \|\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x}_0\|^2 \left(\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}}\right)^\dagger\right)}, \end{aligned} \quad (10)$$

which, after writing the push-forward measure in terms of the whitened parameter, reads

$$\pi_{\text{DI}}(\mathbf{x}|\mathbf{y}) \propto \frac{\exp\left(-\frac{1}{2} \|\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{A}^{-\frac{1}{2}} \mathbf{y}\|^2\right) \times \exp\left(-\frac{\alpha}{2} \|\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x}_0\|^2\right)}{\exp\left(-\frac{\alpha}{2} \|\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x}_0\|^2 \left\|_{\mathbf{\Gamma}^{\frac{1}{2}} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}} \left(\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}}\right)^\dagger \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}}}\right\|}\right)},$$

or equivalently

$$-\log(\pi_{\text{DI}}(\mathbf{x}|\mathbf{y})) \propto \frac{1}{2} \|\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{A}^{-\frac{1}{2}} \mathbf{y}\|^2 + \frac{1}{2} \left\| \mathbf{L} \left(\mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{x}_0 \right) \right\|^2, \quad (11)$$

where

$$\begin{aligned} \mathbf{L}^T \mathbf{L} &= \alpha \left(\mathbf{I} - \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}} \left(\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}} \right)^\dagger \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} \right) \\ &= \alpha \left(\mathbf{I} - \mathbf{V}^n \mathbf{V}^{nT} \right) = \alpha \left(\mathbf{V}^n \right)^\perp \left(\left(\mathbf{V}^n \right)^\perp \right)^T \\ &= \left[\mathbf{V}^n, \left(\mathbf{V}^n \right)^\perp \right] \begin{bmatrix} 0 & 0 \\ 0 & \alpha \mathbf{I} \end{bmatrix} \left[\mathbf{V}^n, \left(\mathbf{V}^n \right)^\perp \right]^T, \end{aligned} \quad (12)$$

where \mathbf{V}^n contains the first n right singular vectors of the following SVD

$$\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} := \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T. \quad (13)$$

As can be seen, the push-forward measure seeks to find the first n columns of \mathbf{V} associated with the n nonzero singular values. The DI method then avoids regularizing these “data-informed directions” \mathbf{V}^n . In other words, in the whitened parameter, the induced regularization by the prior is identity, and the DI approach

removes regularization in the parameter subspace spanned by V^n . From (13) it is clear that V^n now depends on both the prior covariance Γ and the observational covariance Λ in addition to A . *So how do we understand the parameter subspace spanned by V^n and hence the DI approach?* To that end, let us define $\Sigma^{\frac{1}{2}}$ to be the same as Σ except on the main diagonal where $\Sigma^{\frac{1}{2}}(i, i) = \sqrt{\Sigma(i, i)} = \sqrt{\sigma_i}$ (note that $\Sigma^{\frac{1}{2}}$ is nothing more than the square root of Σ when Σ is a square matrix). Let Ψ be the first n rows of $\Sigma^{\frac{1}{2}}$ and Φ be the first n columns of $\Sigma^{\frac{1}{2}}$. Clearly, by definition $\Psi(i, i) = \Phi(i, i) = \sqrt{\sigma_i}$ for $i \leq n$.

Let us define the following maps

$$z := T\mathbf{x}, \quad \text{where} \quad T := \Psi V^T \Gamma^{-\frac{1}{2}}, \quad (14)$$

$$\mathbf{w} := S\mathbf{y}, \quad \text{where} \quad S := \Phi^T U^T \Lambda^{-\frac{1}{2}}. \quad (15)$$

where \mathbf{z} are the first n coordinates of \mathbf{x} in V , after whitening via $\Gamma^{-\frac{1}{2}}$ and then being scaled by Ψ . Similarly, \mathbf{w} are the first n coordinates of \mathbf{y} in U , after whitening via $\Lambda^{-\frac{1}{2}}$ and then being scaled by Φ . The map T pushes forward the prior in \mathbf{x} to the prior in \mathbf{z} as

$$\pi_{\text{prior}}(\mathbf{z}) \sim \exp\left(-\frac{1}{2} \sum_{i=1}^n \sigma_i^{-1} (z_i - \bar{z}_i)^2\right), \quad (16)$$

where $\bar{\mathbf{z}} = T\mathbf{x}_0$. Similarly, given \mathbf{x} (and hence \mathbf{z}), the induced likelihood in terms of \mathbf{w} is given by

$$\pi_{\text{like}}(\mathbf{w}|\mathbf{z}) \sim \exp\left(-\frac{1}{2} \sum_{i=1}^n \sigma_i^{-1} (\mathbf{w}_i - \sigma_i z_i)^2\right). \quad (17)$$

As can be seen from (16) and (17), the maps T and S transform the original parameter \mathbf{x} and original data \mathbf{y} to new parameter \mathbf{z} and new data \mathbf{w} . Two observations are in order: (1) though in general the original parameter and data dimensions are different, the new parameter and data have the same dimension; and (2) the new data \mathbf{w} and new parameter \mathbf{z} , up to the difference in the mean, have the same distribution. In particular, both \mathbf{z} and \mathbf{w} are \mathbb{R}^n -vectors of independent Gaussian distributions with diagonal covariance matrix $\Theta \in \mathbb{R}^{n \times n}$ with $\Theta_{ii} = \sigma_i$. Both z_i and w_i , up to the difference in the mean, are the same Gaussian distribution with variance σ_i . Since $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$, the independent random variable z_i (and hence w_i) is ranked from the one with most variance to the one with least variance.

Let us call the i th column of U , namely U_i , the i th important direction in the data space and the i th column of V , namely V_i , the i th important direction in

the parameter space. Let us also rank the degree of importance of U_i and V_i by the magnitude of σ_i . It follows that the transformations T and S map the original parameter \mathbf{x} and data \mathbf{y} into new parameter \mathbf{z} and data \mathbf{w} in which the corresponding parameter z_i and data w_i are equally important. This is similar to the concept of balanced transformation in control theory (see, e.g., Gugercin and Antoulas 2004; Antoulas 2005 and the references therein). The new parameter \mathbf{z} is thus equally data-informed and prior-informed. In particular z_i is equally less data-informed and prior-informed relatively to z_j for $j < i$.

The DI method thus regularizes only the (equally) data-uninformed and prior-uninformed parameters/directions.

Properties of the DI Regularization Approach

Deterministic Properties

It is easy to see the optimality condition of the optimization problem $\max_{\mathbf{x}} \log(\pi_{\text{DI}}(\mathbf{x}|\mathbf{y}))$ is given by

$$\mathbf{H}\mathbf{x}_{\text{DI}} = \mathbf{b}, \quad (18)$$

where

$$\mathbf{H} := \left\{ \mathbf{A}^T \mathbf{\Lambda}^{-1} \mathbf{A} + \alpha \left[\mathbf{\Gamma}^{-1} - \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \left(\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \right)^\dagger \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \right] \right\},$$

$$\mathbf{b} := \mathbf{A}^T \mathbf{\Lambda}^{-1} \mathbf{y} + \alpha \left[\mathbf{\Gamma}^{-1} - \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \left(\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \right)^\dagger \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \right] \mathbf{x}_0$$

In order to solve the optimality condition (18) in practice, we can use the rank- r approximation

$$\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} = \mathbf{U}^n \mathbf{\Sigma}^n (\mathbf{V}^n)^T \approx \mathbf{U}^r \mathbf{\Sigma}^r (\mathbf{V}^r)^T \quad (19)$$

for the push-forward matrix $\mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \left(\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \right)^\dagger \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A}$, where again n is the largest index for which $\sigma_n > 0$. Thus rank- r approximations (only for the regularization/prior) for \mathbf{H} and \mathbf{y} are given by

$$\mathbf{H}^r := \mathbf{A}^T \mathbf{\Lambda}^{-1} \mathbf{A} + \alpha \left(\mathbf{\Gamma}^{-1} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{V}^r (\mathbf{V}^r)^T \mathbf{\Gamma}^{-\frac{1}{2}} \right),$$

$$\mathbf{b}^r := \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{V}^n \mathbf{\Sigma}^n (\mathbf{U}^n)^T \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{y} + \alpha \left[\mathbf{\Gamma}^{-1} - \mathbf{\Gamma}^{-\frac{1}{2}} \mathbf{V}^r (\mathbf{V}^r)^T \mathbf{\Gamma}^{-\frac{1}{2}} \right] \mathbf{x}_0.$$

Note that we don't perform low-rank approximation for $\mathbf{A}^T \mathbf{\Lambda}^{-1} \mathbf{y}$ in \mathbf{y} as it requires only a matrix-vector product. We also leave the first term in \mathbf{H}^r as is, since we invert

\mathbf{H}^r using the conjugate gradient (CG) method which requires only matrix-vector products. In the numerical results section, we present a nested optimization method (see Algorithm 2) that avoids the low-rank approximation altogether. The analysis of such method is, however, more technical and thus left for future work. *The rank- r approximation to the solution of the optimality condition (18)* is defined as

$$\mathbf{H}^r \mathbf{x}_{\text{DI}}^r = \mathbf{b}^r, \quad (20)$$

for which the corresponding DI inverse formulation is given in (24) (by replacing r_ε with r), which reduces to (5) when $\Lambda = \mathbf{I}$ and $\Gamma = \mathbf{I}$. We can rewrite \mathbf{H}^r in terms of n singular vectors corresponding to the n nonzero singular values as

$$\mathbf{H}^r = \alpha \Gamma^{-\frac{1}{2}} \left[\mathbf{I} + \mathbf{V}^n \mathbf{D}^n (\mathbf{V}^n)^T \right] \Gamma^{-\frac{1}{2}},$$

where \mathbf{D}^n is an $n \times n$ diagonal matrix with $\mathbf{D}^n(i, i) = (\sigma_i^2 - \alpha) / \alpha$ for $i \leq r$ and $\mathbf{D}^n(i, i) = \sigma_i^2 / \alpha$ for $r < i \leq n$.

Lemma 1. *The DI solution with r data-informed modes reads*

$$\mathbf{x}_{\text{DI}}^r := \Gamma^{\frac{1}{2}} \mathbf{V}^n \Theta^n (\mathbf{U}^n)^T \Lambda^{-\frac{1}{2}} \mathbf{y} + \left[\mathbf{I} - \Gamma^{\frac{1}{2}} \mathbf{V}^n \bar{\Gamma}^n (\mathbf{V}^n)^T \Gamma^{-\frac{1}{2}} \right] \mathbf{x}_0, \quad (21)$$

where Θ^n is an $n \times n$ diagonal matrix with $\Theta^n(i, i) = \sigma_i^{-1}$ for $i \leq r$ and $\Theta^n(i, i) = \sigma_i / (\sigma_i^2 + \alpha)$ for $r < i \leq n$. Here, $\bar{\Gamma}^n$ is an $n \times n$ diagonal matrix with $\bar{\Gamma}^n(i, i) = 1$ for $i \leq r$ and $\bar{\Gamma}^n(i, i) = \sigma_i^2 / (\sigma_i^2 + \alpha)$ for $r < i \leq n$. Furthermore,

$$\mathbf{A} \mathbf{x}_{\text{DI}}^n = \Lambda^{\frac{1}{2}} \mathbf{U}^n (\mathbf{U}^n)^T \Lambda^{-\frac{1}{2}} \mathbf{y}. \quad (22)$$

Proof. Using a Woodbury formula, we have

$$(\mathbf{H}^r)^{-1} = \frac{1}{\alpha} \Gamma^{\frac{1}{2}} \left[\mathbf{I} - \mathbf{V}^n \bar{\mathbf{d}}_{\text{DI}}^{n,r} (\mathbf{V}^n)^T \right] \Gamma^{\frac{1}{2}}, \quad (23)$$

where $\bar{\mathbf{d}}_{\text{DI}}^{n,r}$ is an $n \times n$ diagonal matrix with $\bar{\mathbf{d}}_{\text{DI}}^{n,r}(i, i) = (\sigma_i^2 - \alpha) / \sigma_i^2$ for $i \leq r$ and $\bar{\mathbf{d}}_{\text{DI}}^{n,r}(i, i) = \sigma_i^2 / (\sigma_i^2 + \alpha)$ for $r < i \leq n$. The computation of the product $(\mathbf{H}^r)^{-1} \mathbf{y}^r$ to arrive at the assertion is straightforward algebraic manipulation and hence omitted.

The result (22) shows that the image of the DI solution \mathbf{x}_{DI} through the parameter-to-observable map is exactly the data if $\mathbf{U}^n (\mathbf{U}^n)^T = \mathbf{I}$ or $\Lambda^{-\frac{1}{2}} \mathbf{y}$ resides in the column space of \mathbf{U}^n . This happens, for example, when \mathbf{A} has full row rank

and the number of data is not more than the dimension of the parameter, i.e., $d \leq p$. In this case, retaining all modes corresponding to nonzero singular values in the DI solution makes the data misfit vanish, that is, the DI solution in this case would match the noise, which is undesirable. As discussed in section “[Data-Informed Regularization Derivation](#)”, r should be smaller than n for the solution to be meaningful. Let us define

$$r_\varepsilon := \max \{i : 1 \leq i \leq n \text{ and } \sigma_i \geq \varepsilon\},$$

for some $\varepsilon > 0$ (which, as discussed before, can be chosen using the Morozov’s discrepancy principle), and the “reconstruction operator” (Colton and Kress 1998; Kirsch 2011)

$$\mathcal{R}_\varepsilon := (\mathbf{H}^{r_\varepsilon})^{-1} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}}.$$

Theorem 1. For any $\varepsilon > 0$ and $\alpha > 0$, consider the inverse problem

$$\min_{\mathbf{x}} \mathcal{J} = \frac{1}{2} \left\| \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{A}^{-\frac{1}{2}} \mathbf{y} \right\|^2 + \frac{1}{2} \left\| L \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{x} - \mathbf{x}_0) \right\|^2, \tag{24}$$

using the DI approach with rank- r_ε approximation, where

$$L^T L = \alpha \left(\mathbf{I} - \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}} \left(\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{A}^{-\frac{1}{2}} \right)^\dagger \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} \right).$$

The following hold:

- (i) The inverse problem with rank- r_ε DI approach, i.e., the optimization problem (24), is well-posed in the Hadamard sense.
- (ii) Suppose that the nullspace of \mathbf{A} is trivial, i.e., $\mathcal{N}(\mathbf{A}) = \{0\}$, then the DI technique is a regularization strategy (Colton and Kress 1998; Kirsch 2011) in the following sense

$$\lim_{\varepsilon \rightarrow 0} \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} = \mathbf{x}.$$

- (iii) If $\alpha = \mathcal{O}(\varepsilon)$ and $\mathcal{N}(\mathbf{A}) = \{0\}$, then the rank- r_ε DI technique is an admissible regularization method.

Proof. From Lemma 1 we see that the DI solution $\mathbf{x}_{\text{DI}}^{r_\varepsilon}$ is unique and furthermore

$$\left\| \mathbf{x}_{\text{DI}}^{r_\varepsilon} \right\| \leq \beta(\varepsilon, \alpha) \left\| \mathbf{\Gamma}^{\frac{1}{2}} \right\| \left\| \mathbf{A}^{-\frac{1}{2}} \right\| \left\| \mathbf{y} \right\| + \left(1 + \sqrt{\kappa(\mathbf{\Gamma})} \right) \left\| \mathbf{x}_0 \right\|,$$

where $\kappa(\mathbf{\Gamma})$ denotes the condition number of $\mathbf{\Gamma}$, $\beta(\varepsilon, \alpha)$ is a constant defined as

$$\beta(\varepsilon, \alpha) := \frac{1}{\min_{r_\varepsilon < i \leq n} \{r_\varepsilon, \sigma_i + \alpha/\sigma_i\}},$$

which shows the DI solution is stable, which in turn proves *i*). To see assertion *ii*), we use the definition of \mathcal{R}_ε and the SVD of $\mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{I}^{\frac{1}{2}}$ to arrive at

$$\begin{aligned} \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{A} &= \frac{1}{\alpha} \mathbf{I}^{\frac{1}{2}} \left[\mathbf{I} - \mathbf{V}^n \bar{\mathbf{d}}^{n, r_\varepsilon} (\mathbf{V}^n)^T \right] \mathbf{V}^n (\boldsymbol{\Sigma}^n)^2 (\mathbf{V}^n)^T \mathbf{I}^{-\frac{1}{2}} = \\ &= \mathbf{I}^{\frac{1}{2}} \mathbf{V}^n \left[\begin{array}{c|c} \mathbf{I} & 0 \\ \hline 0 & \text{diag} \left(\frac{\sigma_i^2}{\sigma_i^2 + \alpha} \right)_{r_\varepsilon < i \leq n} \end{array} \right] (\mathbf{V}^n)^T \mathbf{I}^{-\frac{1}{2}}, \end{aligned}$$

which implies

$$\lim_{\varepsilon \rightarrow 0} \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} = \mathbf{I}^{\frac{1}{2}} \mathbf{V}^n \mathbf{I} (\mathbf{V}^n)^T \mathbf{I}^{-\frac{1}{2}} \mathbf{x} = \mathbf{x},$$

where we have used the fact that $r_\varepsilon \rightarrow n$ as $\varepsilon \rightarrow 0$ and that $\mathbf{V}^n (\mathbf{V}^n)^T = \mathbf{I}$ since $\mathcal{N}(\mathbf{A}) = \{0\}$.

For assertion *iii*), it is sufficient to show that

$$\sup_y \left\{ \left\| \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{y} - \mathbf{x} \right\| : \left\| \mathbf{A}^{-\frac{1}{2}} (\mathbf{A} \mathbf{x} - \mathbf{y}) \right\| \leq \varepsilon \right\} \rightarrow 0 \text{ as } \varepsilon \rightarrow 0,$$

for any \mathbf{x} . We have

$$\begin{aligned} \left\| \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{y} - \mathbf{x} \right\| &\leq \left\| \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} - \mathbf{x} \right\| + \left\| \mathcal{R}_\varepsilon \mathbf{A}^{-\frac{1}{2}} (\mathbf{A} \mathbf{x} - \mathbf{y}) \right\| \\ &\leq \left\| \mathbf{I}^{\frac{1}{2}} \mathbf{V}^n \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \text{diag} \left(\frac{-\alpha}{\sigma_i^2 + \alpha} \right)_{r_\varepsilon < i \leq n} \end{array} \right] (\mathbf{V}^n)^T \mathbf{I}^{-\frac{1}{2}} \right\| \|\mathbf{x}\| + \|\mathcal{R}_\varepsilon\| \varepsilon \\ &\leq \frac{\alpha}{\sigma_n^2 + \alpha} \sqrt{\kappa(\boldsymbol{\Gamma})} \|\mathbf{x}\| + \varepsilon \left\| \mathbf{I}^{\frac{1}{2}} \left[\begin{array}{c|c} \text{diag} \left(\frac{1}{\sigma_i} \right)_{i \leq r_\varepsilon} & 0 \\ \hline 0 & \text{diag} \left(\frac{\sigma_i}{\sigma_i^2 + \alpha} \right)_{r_\varepsilon < i \leq n} \end{array} \right] \right\| \\ &\leq \frac{\alpha}{\sigma_n^2 + \alpha} \sqrt{\kappa(\boldsymbol{\Gamma})} \|\mathbf{x}\| + \varepsilon \sigma_n^{-1} \left\| \mathbf{I}^{\frac{1}{2}} \right\|, \end{aligned}$$

where we have used the result from *(ii)*, definition of \mathcal{R}_ε , and the orthonormality of \mathbf{V} and \mathbf{U} . Using the assumption $\alpha = \mathcal{O}(\varepsilon)$ concludes the proof.

Remark 2. Note that most of the above arguments are still valid for infinite dimensional setting, i.e., $p = \infty$, assuming that $\mathbf{\Gamma}$ is a trace class. Indeed, $\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}}$ is then a compact operator, and we can invoke the infinite dimensional singular value decomposition (Colton and Kress 1983) for $\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}}$. Note that all the matrices are now interpreted as operators, transpose operator (superscript T) as adjoint operator, and $\mathbf{\Gamma}^{-\frac{1}{2}}$ as pseudo-inverse if $\mathcal{N}(\mathbf{\Gamma}) \neq \{0\}$. We leave out the details for the sake of brevity.

Statistical Properties

Now we discuss some probabilistic aspects of the DI prior and the DI posterior. Since the regularization parameter α plays no role in the following discussion, we absorb it into $\mathbf{\Gamma}$. We define the DI prior as

$$\pi_{\text{DI-prior}}(\mathbf{x}) \sim \exp \left\{ -\frac{1}{2} \left\| L \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{x} - \mathbf{x}_0) \right\|^2 \right\}. \tag{25}$$

From (12), the DI prior (pseudo-) inverse covariance is given by

$$\begin{aligned} (\mathbb{C}^n)^\dagger &:= \mathbf{\Gamma}^{-\frac{1}{2}} \left[\mathbf{I} - \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \left(\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \mathbf{\Lambda}^{-\frac{1}{2}} \right)^\dagger \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} \right] \mathbf{\Gamma}^{-\frac{1}{2}} \\ &= \mathbf{\Gamma}^{-\frac{1}{2}} \left[\mathbf{I} - \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{A}^T \left(\mathbf{A} \mathbf{\Gamma} \mathbf{A}^T \right)^\dagger \mathbf{A} \mathbf{\Gamma}^{\frac{1}{2}} \right] \mathbf{\Gamma}^{-\frac{1}{2}} = \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{V}^n)^\perp \left((\mathbf{V}^n)^\perp \right)^T \mathbf{\Gamma}^{-\frac{1}{2}}, \end{aligned}$$

where we have used the fact that $\mathbf{\Lambda}$ is invertible in the second equality. Thus, \mathbf{A} actually contributes to neither the DI prior nor its rank- r version

$$(\mathbb{C}^r)^\dagger := \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{V}^r)^\perp \left((\mathbf{V}^r)^\perp \right)^T \mathbf{\Gamma}^{-\frac{1}{2}}.$$

The rank- r DI covariance thus reads

$$\mathbb{C}^r := \mathbf{\Gamma}^{\frac{1}{2}} (\mathbf{V}^r)^\perp \left[(\mathbf{V}^r)^\perp \right]^T \mathbf{\Gamma}^{\frac{1}{2}} = \mathbf{\Gamma}^{\frac{1}{2}} \left(\mathbf{I} - \mathbf{V}^r (\mathbf{V}^r)^T \right) \mathbf{\Gamma}^{\frac{1}{2}} \tag{26}$$

which is clearly symmetric positive semidefinite in \mathbb{R}^p , though degenerate. (The nullspace of \mathbb{C}^r : $\mathcal{N}(\mathbb{C}^r) := \left\{ \mathbf{x} : \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{x} \in \mathcal{R}(\mathbf{V}^r) \right\}$, where $\mathcal{R}(\cdot)$ denotes the range space.) The DI prior (25) is not a well-defined density in \mathbb{R}^p , that is, it is not absolutely continuous with respect to the Lebesgue measure in \mathbb{R}^p . This is not surprising as we argue above that the DI prior is the prior on the less data-informed directions. Let us define

$$\mathbf{z}^\perp := \mathbf{T}^\perp \mathbf{x}, \quad \text{where} \quad \mathbf{T}^\perp := \left((\mathbf{V}^r)^\perp \right)^T \mathbf{\Gamma}^{-\frac{1}{2}}.$$

Theorem 2. *The following hold true:*

- (i) \mathbf{z} and \mathbf{z}^\perp are distributed by the push-forward density of the prior through \mathbf{T} and \mathbf{T}^\perp , respectively. In particular, $\mathbf{z} \sim \mathcal{N}(\mathbf{T}\mathbf{x}_0, \mathbf{I})$ and $\mathbf{z}^\perp \sim \mathcal{N}(\mathbf{T}^\perp\mathbf{x}_0, \mathbf{I})$.
- (ii) The DI prior density is the density of \mathbf{z}^\perp and hence is well-defined.
- (iii) The DI prior density is the conditional density of \mathbf{x} given \mathbf{z} .

Proof. Assertion (i) is straightforward. To see the second assertion, we note that the density of \mathbf{z}^\perp , ignoring the normalized constant, can be written as

$$\begin{aligned} \exp \left\{ -\frac{1}{2} \left\| \mathbf{z}^\perp - \mathbf{T}^\perp \mathbf{x}_0 \right\|^2 \right\} &= \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}_0) (\mathbf{T}^\perp)^T \mathbf{T}^\perp (\mathbf{x} - \mathbf{x}_0) \right\} \\ &= \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}_0) \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{V}^r)^\perp \left((\mathbf{V}^r)^\perp \right)^T \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{x} - \mathbf{x}_0) \right\}, \end{aligned}$$

which is exactly the DI prior (25). In other words, we have shown that the DI prior is a well-defined density on \mathbf{z}^\perp . To see assertion (iii), we observe that

$$\pi_{\text{prior}}(\mathbf{x}) = \pi_{\text{prior}}\left(\mathbf{V}^r \mathbf{z} + (\mathbf{V}^r)^\perp \mathbf{z}^\perp\right),$$

and owing to $\mathbf{z} = \mathbf{T}\mathbf{x}$, again ignoring the normalized constant, we have

$$\begin{aligned} \pi_{\text{prior}}(\mathbf{x}|\mathbf{z}) &= \frac{\pi_{\text{prior}}(\mathbf{x})}{\pi(\mathbf{z})} \\ &= \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{x}_0) \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{V}^r)^\perp \left((\mathbf{V}^r)^\perp \right)^T \mathbf{\Gamma}^{-\frac{1}{2}} (\mathbf{x} - \mathbf{x}_0) \right\}, \end{aligned}$$

which is exactly the DI prior since $\pi(\mathbf{z}) = \mathcal{N}(\mathbf{T}\mathbf{x}_0, \mathbf{I})$ is exactly the push-forward density of $\pi_{\text{prior}}(\mathbf{x})$ via the map \mathbf{T} .

Remark 3. Note that the above decomposition of \mathbf{x} into \mathbf{z} and \mathbf{z}^\perp , through the maps \mathbf{T} and \mathbf{T}^\perp , is still valid for infinite dimensional settings. However, \mathbf{z}^\perp would be distributed by an infinite dimensional Gaussian measure with identity covariance operator, which is not a valid Gaussian measure. A more general understanding of the DI prior is through disintegration. Indeed, under mild conditions on the map \mathbf{T} and its push-forward measure of the prior measure, the DI prior $\pi_{\text{prior}}(\mathbf{x}|\mathbf{z})$ is nothing more than a disintegration of the prior measure via the map \mathbf{T} , and this view is also valid for infinite dimensional settings.

To quantify the uncertainty in the DI inverse solution (21), we can use the covariance matrix of the DI posterior (10). For linear inverse problems with Gaussian prior and Gaussian noise—the problems considered in this chapter—the

Table 1 The difference between the DI and the Tikhonov covariance matrices

	$i \leq r$	$r < i \leq n$
DI posterior	$\bar{d}_{\text{DI}}^{n,r}(i, i) = \frac{\sigma_i^2 - \alpha}{\sigma_i^2}$	$\bar{d}_{\text{DI}}^{n,r}(i, i) = \frac{\sigma_i^2}{\sigma_i^2 + \alpha}$
Tikhonov posterior	$\bar{d}_{\text{Tik}}^{n,r}(i, i) = \frac{\sigma_i^2}{\sigma_i^2 + \alpha}$	$\bar{d}_{\text{Tik}}^{n,r}(i, i) = \frac{\sigma_i^2}{\sigma_i^2 + \alpha}$

covariance matrix is exactly the inverse of the Hessian. For rank- r DI approach, the DI posterior covariance matrix $C_{\text{DI}}^{\text{post}}$ is given in (23), i.e.,

$$C_{\text{DI}}^{\text{post}} = \frac{1}{\alpha} \mathbf{\Gamma} - \frac{1}{\alpha} \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{V}^n \bar{d}_{\text{DI}}^{n,r} (\mathbf{V}^n)^T \mathbf{\Gamma}^{\frac{1}{2}} \tag{27}$$

It is easy to see that the covariance matrix corresponding to the Tikhonov regularization is given by

$$C_{\text{Tik}}^{\text{post}} = \frac{1}{\alpha} \mathbf{\Gamma} - \frac{1}{\alpha} \mathbf{\Gamma}^{\frac{1}{2}} \mathbf{V}^n \bar{d}_{\text{Tik}}^{n,r} (\mathbf{V}^n)^T \mathbf{\Gamma}^{\frac{1}{2}}, \tag{28}$$

where both $\bar{d}_{\text{DI}}^{n,r}$ and $\bar{d}_{\text{Tik}}^{n,r}$ are diagonal matrices given in Table 1. Note that we have used α as the magnitude of the regularization to study the robustness and accuracy of all methods. If not needed, α can be straightforwardly absorbed into $\mathbf{\Gamma}$, and hence σ_i^2 ; in that case α is simply replaced by 1 everywhere it appears (including those in Table 1). As can be seen, $\bar{d}_{\text{Tik}}^{n,r}(i, i)$ is always non-negative for all i , while $\bar{d}_{\text{DI}}^{n,r}(i, i)$ is negative when $\sigma_i^2 < \alpha$ for $i \leq r$. That is, while the Tikhonov posterior uncertainty, $C_{\text{Tik}}^{\text{post}}$ (Bayesian posterior with standard Gaussian prior), is always smaller than the prior uncertainty $\mathbf{\Gamma}$ no matter how much informed the data is, the DI posterior uncertainty could be higher than the prior counterpart if the data supports this. In other words, standard (or typical) Gaussian priors do not allow the data to increase the uncertainty and hence are prone to producing overconfident results (see section “Applications to Imaging Problems”). The DI prior, on the other hand, takes the parameter-to-observable map (the proxy to the data) into account, and thus along parameter directions that are more data-informed, i.e. $\sigma_i^2 \geq \alpha$, the posterior uncertainty is reduced relative to the prior uncertainty. Along parameter directions that are less data-informed, i.e., $\sigma_i^2 < \alpha$, the posterior uncertainty increases relative to the prior uncertainty.

Applications to Imaging Problems

Image Deblurring

One typical inverse problem in imaging is image deblurring. Given some blurry image, we want to recover the true, sharp image. To understand the deblurring process, we must first understand how an image becomes blurred in the first place.

A simple and effective mathematical model of the blurring process is convolution of a sharp image with a blurring kernel. This blurring kernel is often described mathematically as a point spread function (PSF). The PSF describes how energy from a point source (i.e., a single pixel) is *smear*ed out among neighboring pixels, resulting in a blur.

Since convolution is a linear operation, it can be expressed mathematically as

$$\mathcal{A}X_{true} = \mathbf{B} \quad (29)$$

where \mathcal{A} is the blurring (convolution) operator acting on the true image $X_{true} \in \mathbb{R}^{m_1 \times m_2}$ resulting in the blurred image $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$. By stacking (or *vectorizing*) the columns of X_{true} , we can write (29) as a linear algebraic equation. Let us denote by \mathbf{x}_{true} the vectorized true image and by \mathbf{y} the vectorized blurred image, i.e.,

$$\mathbf{x}_{true} = \text{vec}(X_{true}) \in \mathbb{R}^{m_1 m_2}, \quad \mathbf{y} = \text{vec}(\mathbf{B}) \in \mathbb{R}^{m_1 m_2}$$

Also, since \mathcal{A} is a linear operator acting on a vector, it has a matrix representation denoted by $\mathbf{A} \in \mathbb{R}^{m_1 m_2 \times m_1 m_2}$. Finally, (29) becomes

$$\mathbf{A}\mathbf{x}_{true} = \mathbf{y} \quad (30)$$

Note that while this notation is convenient for manipulating mathematically, it is not efficient to construct the two-dimensional convolution matrix. \mathbf{A} is a large sparse matrix, which, for large problems, cannot be stored in memory. Even on problems small enough to fit in memory, it is computationally expensive to explicitly construct this matrix. Fortunately, there are efficient methods for computing spectral decompositions of the matrices arising from convolution operators using the fast Fourier transform and discrete cosine transform. While interesting in their own right, these implementation details are not necessary for the following discussion. For a detailed treatment of image deblurring problems and algorithms, the interested reader is encouraged to consult (Hansen et al. 2006).

For all examples considered in this chapter

$$\mathbf{A} = \lambda^2 \mathbf{I}, \quad \text{and } \mathbf{\Gamma} = \mathbf{I},$$

where λ is the noise level (the standard deviation).

Since truncated SVD (TSVD) and Tikhonov are spectral filtering methods, the regularized solution using these methods can be written using the following common form:

$$\mathbf{x}_{filt} = \sum_{i=1}^p \phi_i \frac{\mathbf{U}_i^T \mathbf{y}}{\sigma_i} \mathbf{V}_i, \quad (31)$$

where ϕ_i is usually called the *filter factor* as it has the effect of filtering (damping) when ϕ_i is close to 0. It can be shown that the filter factor for rank- r TSVD is given by

$$\phi_i = \begin{cases} 1, & i \leq r \\ 0, & \text{otherwise.} \end{cases}$$

Likewise, the filter factor for Tikhonov regularization is given by

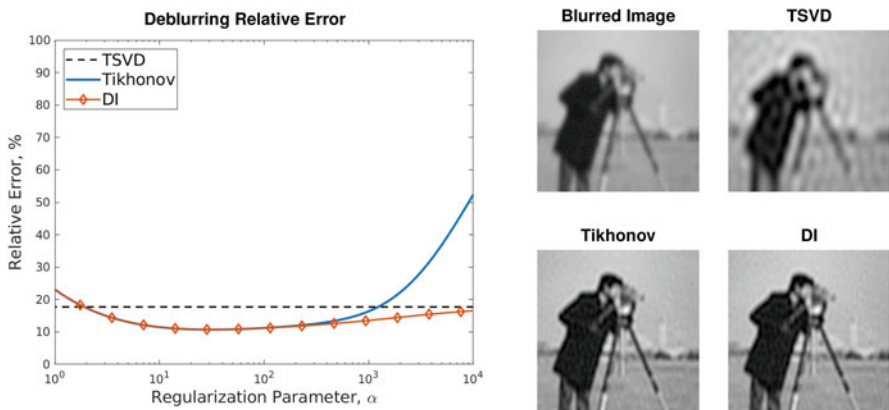
$$\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha}$$

As discussed in section “[A Data-Informed Regularization \(DI\) Approach](#)”, the DI method with rank- r approximation removes regularization on the first r directions \mathbf{V}_i , $1 \leq i \leq r$, while being the same as Tikhonov on the other directions. For $\mathbf{\Gamma} = \mathbf{I}$ and $\mathbf{x}_0 = 0$ the DI solution (see Lemma 1) can be written in the filtered form as

$$\phi_i = \begin{cases} 1, & i \leq r \\ \frac{\sigma_i^2}{\sigma_i^2 + \alpha}, & \text{otherwise.} \end{cases}$$

Remark 4. It should be emphasized that the DI method also shares the same spectral decomposition form in this case because $\mathbf{\Gamma} = \mathbf{I}$ and $\mathbf{x}_0 = 0$. When $\mathbf{\Gamma} \neq \mathbf{I}$, singular vectors of $\mathbf{A}^{-\frac{1}{2}}\mathbf{A}$ do not necessarily diagonalize both \mathbf{A} and $\mathbf{\Gamma}$ simultaneously. In other words, the filtered form (31) is not valid for the DI approach unless \mathbf{U} and \mathbf{V} are singular vectors of $\mathbf{A}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}^{\frac{1}{2}}$ and $\mathbf{x}_0 = 0$. When $\mathbf{x}_0 \neq 0$, there is an additional term contributed from \mathbf{x}_0 as shown in the DI solution given in Lemma 1.

We can see here again that (1) when $r \rightarrow 0$, DI approaches Tikhonov; (2) when $\alpha \ll \sigma_i$ for $i \leq r$, Tikhonov is close to DI; and 3) when $\alpha \rightarrow \infty$, DI converges to TSVD. This can be clearly seen in Fig. 5a for a deblurring problem in which we plot the relative error between the deblurred images and the original ones for $m_1 = m_2 = 128$, $\lambda = 0.01$, $r = 400$, and a wide range of α . For the under-regularization regime, i.e., $\alpha < 1$, which should be avoided, the regularization is not sufficient to suppress the oscillations due to the high-frequency modes for both Tikhonov and DI methods, resulting in inaccurate reconstructions. For reasonable-to-over-regularization regimes, i.e., $\alpha > 1$, DI is the best compared to both Tikhonov and TSVD method as it combines the advantages from both sides. That is: (1) DI behaves similar to Tikhonov for reasonable (but small) regularization and outperforms Tikhonov in reasonable-to-over-regularization regimes; and (2) compared to TSVD, DI is more accurate for reasonable regularization parameters as it maintains the benefits of keeping useful information from all parameter directions while avoiding potential errors caused by over-regularization. Consequently, the DI



(a) Effect of noise on DI and Tikhonov solutions

(b) Deblurring results with $\alpha = 100$



(c) Deblurring results with $\alpha = 1000$

(d) Deblurring results with $\alpha = 5000$

Fig. 5 Deblurring results for $m_1 = m_2 = 128$, $\lambda = 0.01$, $r = 400$. (a) relative error between deblurred images and the truth for a range of regularization parameter $\alpha \in [1, 10^4]$. (b) the DI deblurred image with $\alpha = 100$. (c) the DI deblurred image with $\alpha = 1000$. (d) the DI deblurred image with $\alpha = 5000$

error is the smallest of the three methods discussed for all $\alpha > 10^3$, and DI is robust with respect to the regularization parameter.

In Fig. 5b are the deblurred images for $\alpha = 100$ corresponding to the smallest deblurring error for both DI and Tikhonov. As can be seen, the Tikhonov result is similar to the DI one, while the truncated SVD result is blurry as it removes (putting infinite regularization on) useful information in directions V_i for $i > r$. Figure 5c, d show the deblurred images for $\alpha = 1000$ and $\alpha = 5000$, respectively, corresponding to cases where DI outperforms both Tikhonov and TSVD (see Fig. 5a). Indeed, the DI deblurred image has higher quality.

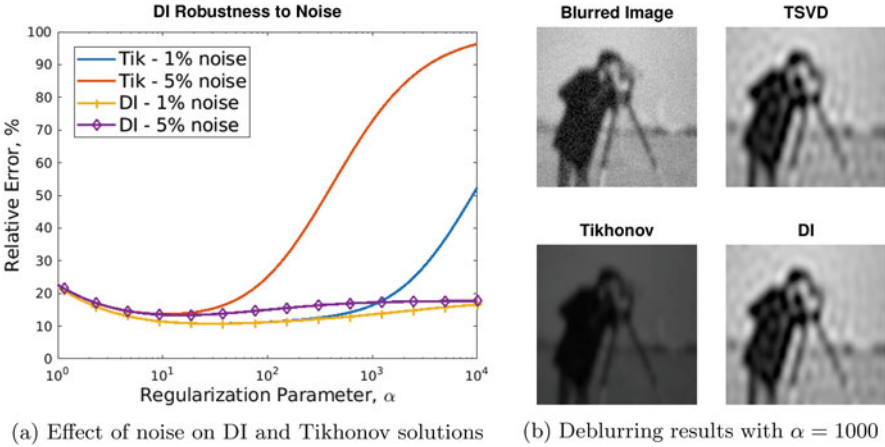


Fig. 6 Deblurring results for $m_1 = m_2 = 128$, $\lambda = 0.05$, $r = 400$. (a) relative error of DI and Tikhonov solutions with respect to true solution for noise levels of 1% and 5% and $\alpha \in [1, 10^4]$. (b) the DI, Tikhonov, and TSVD deblurred images with $\alpha = 1000$

In order to see if the DI method is sensitive to noise, we now consider the case with $\lambda = 5\%$ noise. Deblurring accuracy for this case (purple) is shown in Fig. 6a together with the accuracy for the case of 1% noise (yellow). As can be seen, the solution quality of the DI method does not degrade significantly due to the presence of noise. Compare this to the difference seen in the Tikhonov method (red and blue curves) with the increase in noise level, we can see that the solution quality of the Tikhonov method degrades rapidly in the presence of noise. It can also be seen that Tikhonov regularization becomes more sensitive to the choice of α as the noise increases. Since the DI method regularizes only the data-uninformed directions, which also contain much of the noise, increasing the noise level has little effect on the solution quality.

For the rest of this section, we consider the more challenging cases with $\lambda = 5\%$ noise. To make the problem even more challenging, we consider images with missing pixels to simulate more interesting cases when images are damaged or incomplete. Figure 7 show the deblurring results using DI, TSVD, and Tikhonov (Tik) regularizations for damaged images with $m_1 = m_2 = 128$, $r = 400$. The first column contains four scenarios with 10% random data, 25% random data, 50% random data, and 100% data, all with noise. Note that we plot the damaged images by filling the missing data with 0. The second column contains the corresponding TSVD deblurring results. The last four columns contain the results from DI and Tikhonov with $\alpha = 10$ and 20. As can be observed, all methods are able to deblur and at the same time recover the true image quite well even with only 10% data. Both DI and Tikhonov yield clearer images compared to TSVD. The Tikhonov results are “darker,” especially with $\alpha = 20$, indicating over-regularization, while the DI images are insensitive to regularization parameter as the data-informed modes are

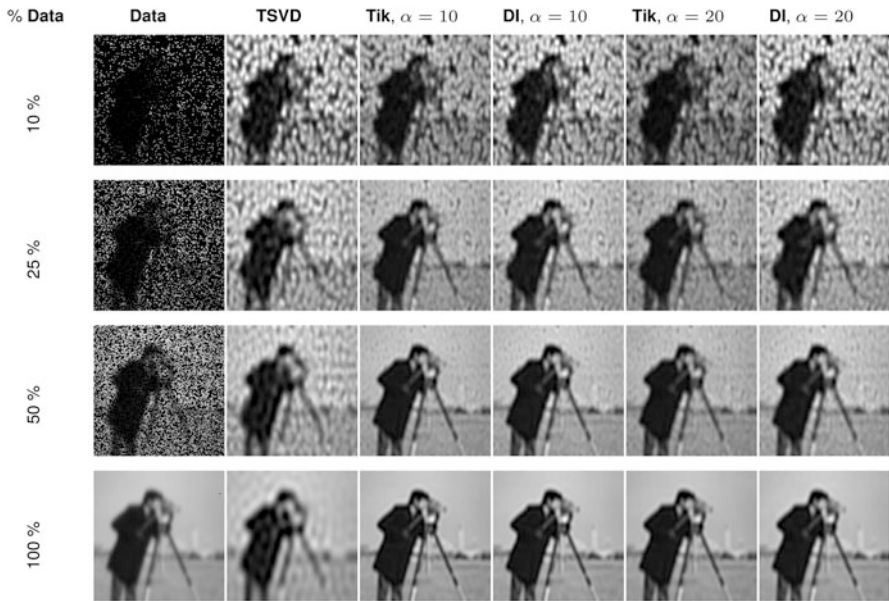


Fig. 7 Deblurring results using DI, TSVD, and Tikhonov (Tik) regularizations for damaged images with $m_1 = m_2 = 128$, $\lambda = 0.05$, $r = 400$. The first column consists of four scenarios with 10%, 25%, 50%, and 100% data. The second column is the corresponding TSVD deblurring results. The last four columns are the results from DI and Tikhonov with $\alpha = 10$ and 20

left untouched. Indeed, Fig. 8 clearly demonstrates these expected results for larger regularization parameters ($\alpha = 50$ and $\alpha = 100$).

Recall that the goal of sections “A Statistical Data-Informed (DI) Inverse Framework” and “Statistical Properties” is to gain insights into statistical properties of the DI prior. For linear parameter-to-observable maps—which are the cases for this chapter—with Gaussian observational noise, the posterior is also a Gaussian. As a result, the result at the end of section “Statistical Properties” also allows us to use the posterior covariances (27) and (28) to estimate the uncertainty in the corresponding inverse solutions. Since the posterior for either Tikhonov or DI prior is Gaussian, its diagonal contains the marginal pixel-wise variances, which can be used as a measure of uncertainty for each pixel. Clearly this does not take into account the correlation among pixels, but is straightforward to have a glimpse of uncertainty in high-dimensional (128^2 -dimensional) spaces. We now study the uncertainty estimation in the solution of deblurring problems.

To begin, it is important to distinguish the following two cases:

- *Case I*: using only rank- r DI regularization in which rank- r approximation for the pseudo-inverse $\left(\Lambda^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\Lambda^{-\frac{1}{2}}\right)^\dagger$ is done as we have presented. The DI

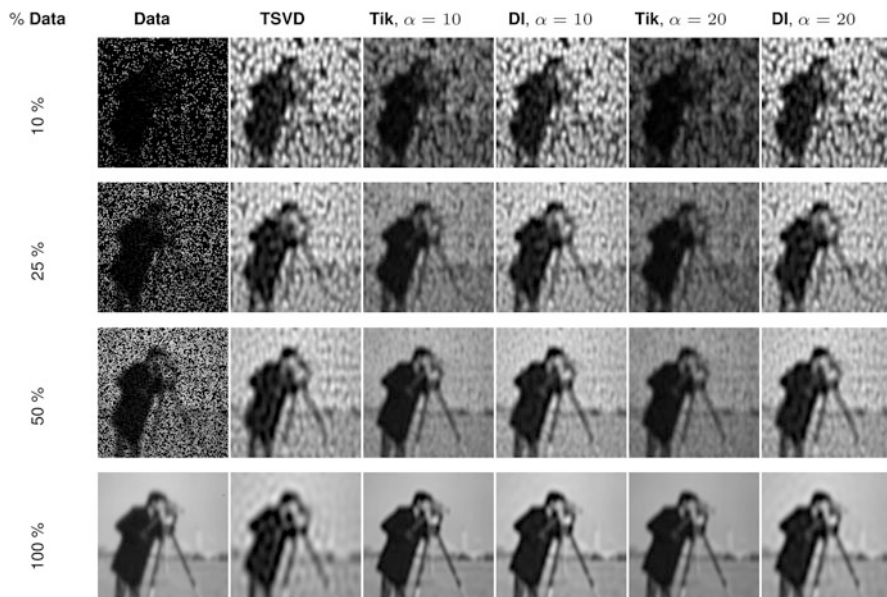


Fig. 8 Deblurring results using DI, TSVD, and Tikhonov (Tik) regularizations for damaged images with $m_1 = m_2 = 128$, $\lambda = 0.05$, $r = 400$. The first column consists of four scenarios with 10%, 25%, 50%, and 100% data. The second column is the corresponding TSVD deblurring results. The last four columns are the results from DI and Tikhonov with $\alpha = 50$ and 100

posterior covariance (27) thus involves the second and third columns in Table 1, and a rank- n SVD (13) is needed.

- *Case II*: performing rank- r low-rank approximation of the posterior covariance in addition to rank- r DI regularization. This amounts to using only the second column of Table 1 for the DI posterior covariance in (27). This case is typically more practical for large-scale problems as only a rank- r SVD (19) is needed.

In Fig. 9a are the minimum pixel-wise variances for four scenarios with 10% random data, 25% random data, 50% random data, and 100% random data for *Case II*. As can be seen, the uncertainty corresponding to the case of missing data is lower than the uncertainty for full data case! We expect the opposite, that is, more available (supposedly) informative data is expected to lead to lower uncertainty in the inverse solution. The observation is twofold: first, care needs to be taken for *Case II* results as rank- r approximation may not provide accurate uncertainty; second, for 10% data case, when $r > 500$ the uncertainty is larger compared to the full data case. This suggests that r needs to be sufficiently large for an accurate uncertainty estimation, and this will be confirmed in the discussion below for *Case I* in which we use the full rank (rank- n) decomposition (13). The criteria for estimating such a value of r are a subject for future research. (At the moment of writing this chapter, we have not yet found such a criteria.)

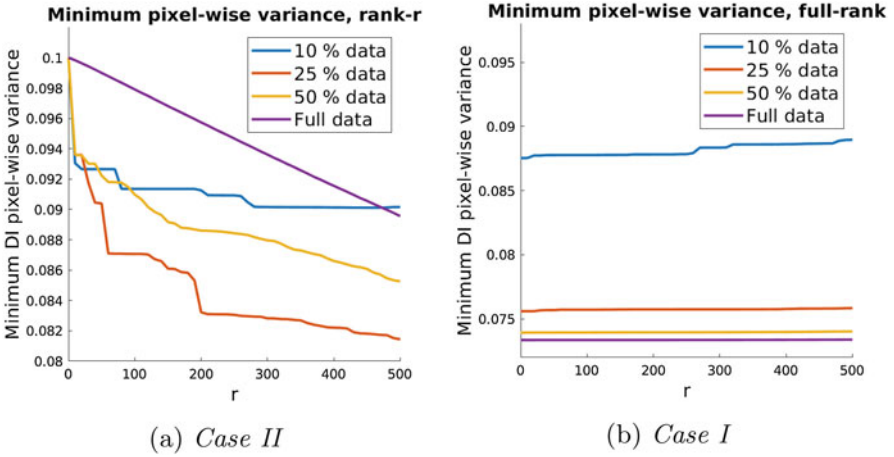


Fig. 9 Rank- r DI posterior pixel-wise uncertainty using rank- n SVD decomposition (*Case I* with both the second and third columns of Table 1) and using rank- r SVD decomposition (*Case II* with only the second column of Table 1)

We next discuss the results for *Case I*. Again, this requires a rank- n SVD (13), where n is the rank of A , to compute (27) using Table 1. Figure 9b shows that the minimum uncertainty for any missing data case is higher than the full data case regardless of any value of r in rank- r DI regularization. As also expected, the uncertainty scales inversely with the amount of available data, i.e., the more informative data we have, the smaller the uncertainty in the inverse solution. Note that the result and the conclusion for the largest pixel-wise variances are similar and hence omitted here.

We now compare the DI and Tikhonov posterior uncertainty estimations. Since *Case I*, though more expensive, provides more accurate uncertainty estimation, it is used for computing DI posterior pixel-wise variances. To be fair, we also use the full decomposition for Tikhonov regularization. In other words, the following comparison is based on (27) and (28) and Table 1. As discussed above in Figs. 6a and 7, $\alpha = 10$ corresponds to a case in the region where DI and Tikhonov give nearly the same reconstructions (in fact Tikhonov slightly over-regularizes), so let us start with this case first. Figure 10 shows that the DI posterior has higher pixel-wise variance than the Tikhonov posterior. This is consistent with the result and the discussion of Table 1 and Fig. 7, that is, the Tikhonov posterior is not only over-regularizing but also overconfident. For both methods, regions of higher uncertainty are visually discernible where data is missing. In the case of 100% data, the result is the same, namely, Tikhonov uncertainty estimation subjectively is less than the DI uncertainty estimation. In this case, the uncertainty estimate is not very interesting: both DI and Tikhonov have approximately uniform uncertainty everywhere as we have data everywhere. We next consider the case with $\alpha = 1000$ where Tikhonov significantly over-regularizes (see Fig. 6b). Figure 11, shows that while Tikhonov is

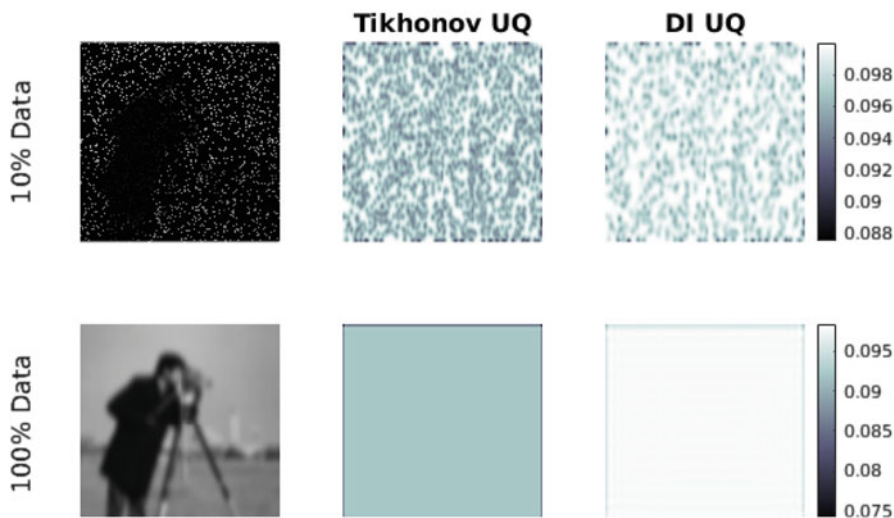


Fig. 10 Visualization of pixel-wise variance estimates for the deblurring problem with $\lambda = 0.05$, $r = 400$, and $\alpha = 10$. In the left column are the noisy images with 10% data and 100% data. In the second column are the Tikhonov uncertainty estimates for 10% data (top) and 100% data (bottom). Likewise, the third column contains the DI uncertainty estimates for 10% data (top) and 100% data (bottom)

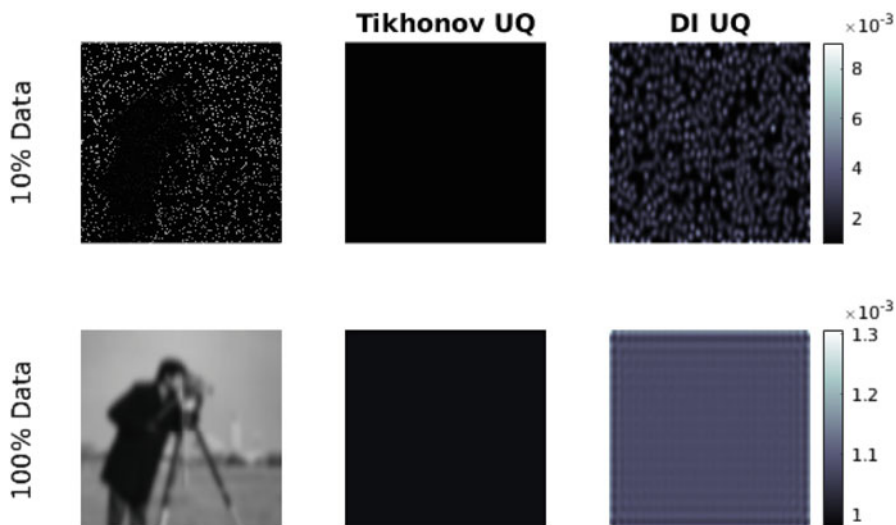


Fig. 11 Visualization of pixel-wise variances for the deblurring problem with $\lambda = 0.05$, $r = 400$, and $\alpha = 1000$. In the left column are the noisy images with 10% data and 100% data. In the second column are the Tikhonov uncertainty estimates for 10% data (top) and 100% data (bottom). The third column contains the DI uncertainty estimates for 10% data (top) and 100% data (bottom)

uniformly (very) overconfident, i.e., having small posterior uncertainty everywhere, DI gives informative UQ results. The latter can be clearly seen for the case with 10% data in which the uncertainty is higher for missing pixels. This implies that the DI priors could provide more useful UQ results than the Tikhonov (standard Gaussian) ones.

Image Denoising

We can extend the idea of data-informed (DI) regularization to the image denoising problem. Since noise typically resides in the high-frequency portion of the image, denoising can be performed by applying spectral filtering techniques directly to the noisy image. These noisy high-frequency modes are also the *less informative* modes in the DI setting. Taking the SVD of the noisy image, \mathbf{X}_{noisy} , we have

$$\mathbf{X}_{noisy} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \sum_i \sigma_i \mathbf{U}_i \mathbf{V}_i^T,$$

The denoised image can be obtained by “filtering” the noise as

$$\mathbf{X}_{filt} = \mathbf{U} \mathbf{\Sigma}^{filt} \mathbf{V}^T = \sum_i \phi_i \sigma_i \mathbf{U}_i \mathbf{V}_i^T,$$

where $\mathbf{\Sigma}^{filt}$ is the diagonal matrix with $\Sigma_{ii}^{filt} = \phi_i \sigma_i$. The filter factors ϕ_i are the same as those defined for the deblurring case. For a numerical demonstration, we pick a noisy image (Hansen et al. 2006) with 20% noise (see the top-left sub-figure of Fig. 12a). Shown in Fig. 12a are denoised results using DI with $r = 20$ and $\alpha = 100$, TSVD with $r = 20$, and Tikhonov with $\alpha = 100$. Though the difference in the results is not clearly visible, the DI has smaller error compared the other two methods. This can be verified in Fig. 12b where the relative error between the denoised image and the true one for a wide range of “regularization parameter” $\alpha \in [10^{-2}, 10^4]$ is presented. Clearly, we would not choose $\alpha < 1$ as these correspond to under-regularization. For $\alpha > 1$, DI is the best compared to both Tikhonov and TSVD method as it combines the advantages from both methods. Indeed, the DI error is smallest for all $\alpha > 1$, and DI is robust with regularization parameter.

X-Ray Tomography

In the previous two examples, we have been able to implement spectral filtering methods directly by introducing filter factors which effectively modified the singular values to minimize the impact of noise on the inversion process. (Recall that the DI method also shares the same spectral decomposition form in this case because

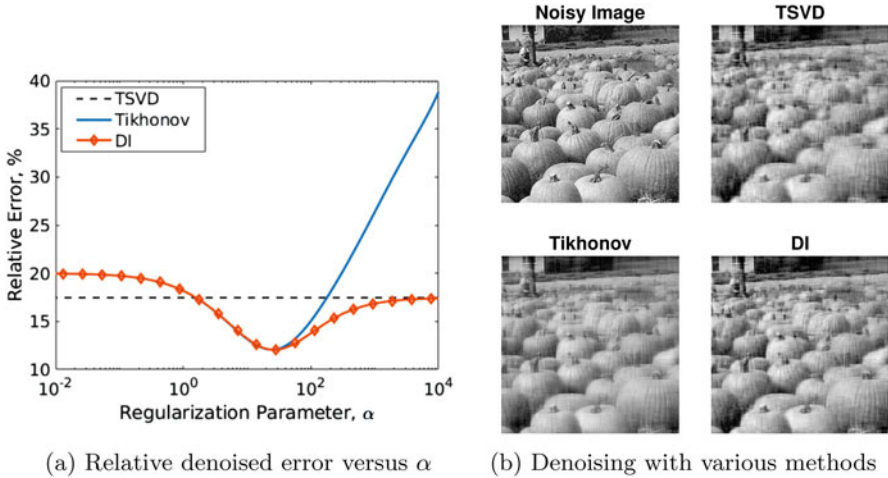


Fig. 12 Denoising with DI, Tikhonov, and TSVD methods. (a) The relative error between the denoised image and the true one for a wide range of “regularization parameter”. The DI error is smallest for all $\alpha > 1$ (corresponding reasonable to over-regularization regimes). (b) Denoised results using DI with $r = 20$ and $\alpha = 100$, TSVD with $r = 20$, and Tikhonov with $\alpha = 100$

$\Gamma = I$ and $\mathbf{x}_0 = 0$.) Each method relied on computing a full factorization of $\Lambda^{-\frac{1}{2}} \mathbf{A}$ and then applying filters. While this is an effective and straightforward method to solve small-to-moderate inverse problems that helps provide insight into each approach, it can be cumbersome or even computationally infeasible to compute full factorizations for large-scale problems. It is not uncommon that inverse problems arising in imaging applications can lead to very large matrix operators. Indeed, we have seen even in the toy image deblurring problem in section “Image Deblurring” that matrix size of 16384×16384 is significantly large, and we have employed more sophisticated methods to compute the factorization of the convolution operator. For many problems, however, such efficient factorizations may not exist, or it is computationally prohibitive to compute a full factorization.

One way to overcome the challenge of factorizing large matrices is to solve the optimality condition (20) iteratively. Since \mathbf{H} is symmetric positive definite, we choose the conjugate gradient (CG) method (see, e.g. Shewchuk 1994 and the references therein) which requires only matrix-vector products, which in turn avoids forming any matrices (including \mathbf{A} or \mathbf{H}) completely. We consider two variants: (a) using CG to solve for (20), that is, we still require rank- r approximation of the DI regularization, and (b) using CG to solve for (18), that is, a rank- r approximation of the DI regularization is not required. In this case we use a least-squares optimization method to compute the pseudo-inverse $\left(\Lambda^{-\frac{1}{2}} \mathbf{A} \Gamma \mathbf{A}^T \Lambda^{-\frac{1}{2}}\right)^\dagger$ acting on a vector for each CG iteration.

The detailed computational procedure for the a)-variant is given in Algorithm 1. Note that the viability of this method for large-scale problems relies on

the availability of a randomized eigensolver to compute eigenvectors of $\mathbf{A}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}\mathbf{A}^T\mathbf{A}^{-\frac{1}{2}}$ (and thus right singular vectors of $\mathbf{A}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}^{\frac{1}{2}}$) which does not require explicit construction of a matrix, only access to matrix-vector products.

Algorithm 1 Data-informed inversion using randomized eigensolver and CG

Input: Data \mathbf{y} , number of eigenvectors r , prior \mathbf{x}_0 , prior covariance matrix $\mathbf{\Gamma}$, noise covariance matrix \mathbf{A} , regularization parameter α

- 1: Define $\mathbf{F} := \mathbf{A}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Gamma}^{\frac{1}{2}}$.
- 2: Create functions to compute matrix-vector products $\mathbf{F}\mathbf{x}$ and $\mathbf{F}^T\mathbf{x}$.
- 3: Compute the first r eigenvectors (\mathbf{V}^r) of $\mathbf{F}^T\mathbf{F}$ using a randomized eigensolver.
- 4: Solve linear equation (20), i.e.,

$$\mathbf{\Gamma}^{-\frac{1}{2}}\left[\mathbf{F}^T\mathbf{F} + \alpha(\mathbf{I} - \mathbf{V}_r\mathbf{V}_r^T)\right]\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{x} = \mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{F}^T\mathbf{y} + \alpha\mathbf{\Gamma}^{-\frac{1}{2}}(\mathbf{I} - \mathbf{V}_r\mathbf{V}_r^T)\mathbf{\Gamma}^{-\frac{1}{2}}\mathbf{x}_0$$

using the conjugate gradient method.

To demonstrate the effectiveness of this approach for the DI method, we choose to solve the inverse problem of reconstructing an image from X-ray measurements. The forward model of generating X-ray measurements, \mathbf{A} , is given by the Radon transform, and \mathbf{A}^T is given by the inverse Radon transform. A more detailed description of the X-ray tomography inverse problem is given in Mueller and Siltanen (2012) (and the references therein). The problem setup in this section exactly follows the setup given in Mueller and Siltanen (2012). We use the MATLAB Image Processing Toolbox to compute the product of the Radon transform \mathbf{A} and its inverse \mathbf{A}^T with a vector. Results using Algorithm 1 for a popular 256×256 phantom image with 256 measurement angles are shown in Fig. 13 for various values of the regularization parameter α and the rank r . Each row contains the results for each regularization parameter with different values of r . The corresponding values for α and r can also be found in the rows and columns of Table 2. Note that below each figure is the relative error of the corresponding reconstruction and the actual phantom image. These relative errors are collected in Table 2 for clarity. Note that for the last two images on the last row of Fig. 13, CG does not converge, and this issue is still under investigation. Other than that the observations are similar to the previous section. That is, compared to Tikhonov, DI is robust to the regularization parameter, and it is at least as good as Tikhonov regardless of the values of regularization parameter α and rank r .

Next we present the detailed computational procedure for the b)-variant in Algorithm 2. In order to compare variant b) with variant a), we compute the relative error of the reconstruction and the true image for various values of regularization parameter α . From the results in Fig. 13, we choose $r = 200$ to balance the accuracy and the cost of the eigensolver. The result is in Fig. 14, which shows that the b)-variant (red curve) is at least as good as the a)-variant (blue curve) while not requiring low-rank approximations. Indeed, to demonstrate this, we pick $\alpha = 100$

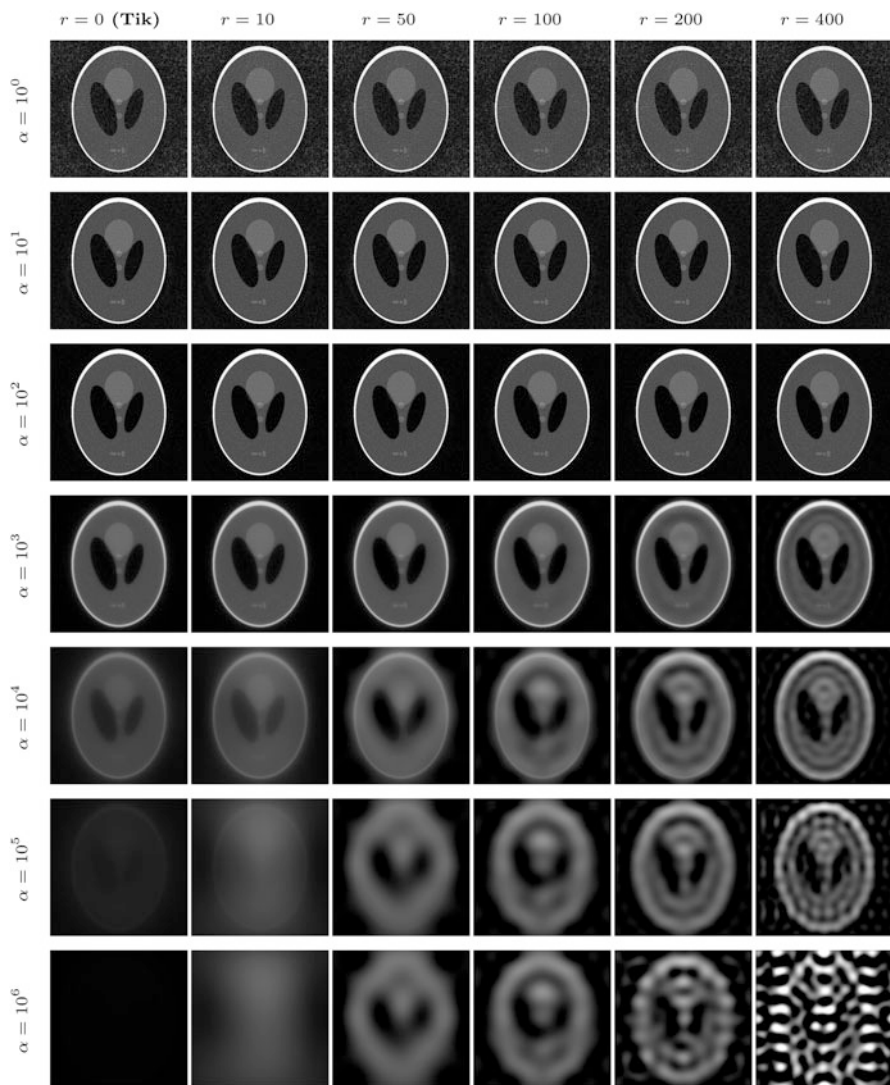


Fig. 13 DI reconstructions for various values of the regularization parameter α and the rank r . Each row contains the results for each regularization parameter with different values of r . The corresponding values for α and r can be found in the rows and columns of Table 2 along with the relative error between the reconstructed image and the true phantom

for which Fig. 14 shows that both variants give similar reconstruction quality, and the reconstruction from both variants is shown in Fig. 15. As can be seen, the result from the b)-variant looks much clearer, which is expected in this case, as $r = 200$ is not sufficient to capture all the data-informed modes for the a)-variant. *By using*

Table 2 Comparison of the relative errors of the DI solution estimate for various regularization parameters α and various values for r . The noise level here is $\lambda = 1\%$

α	Relative Error, %					
	$r = 0$ (Tik)	$r = 10$	$r = 50$	$r = 100$	$r = 200$	$r = 400$
1	33.52	33.52	33.52	33.52	33.52	33.52
10	31.73	31.73	31.73	31.73	31.73	31.73
100	24.44	24.45	24.45	24.45	24.45	24.45
1000	29.81	29.80	29.72	29.66	29.51	29.09
10^4	58.76	58.52	56.93	55.92	54.03	50.43
10^5	81.77	77.10	70.33	67.78	63.84	57.84
10^6	96.09	81.29	72.44	69.50	81.80	299.73

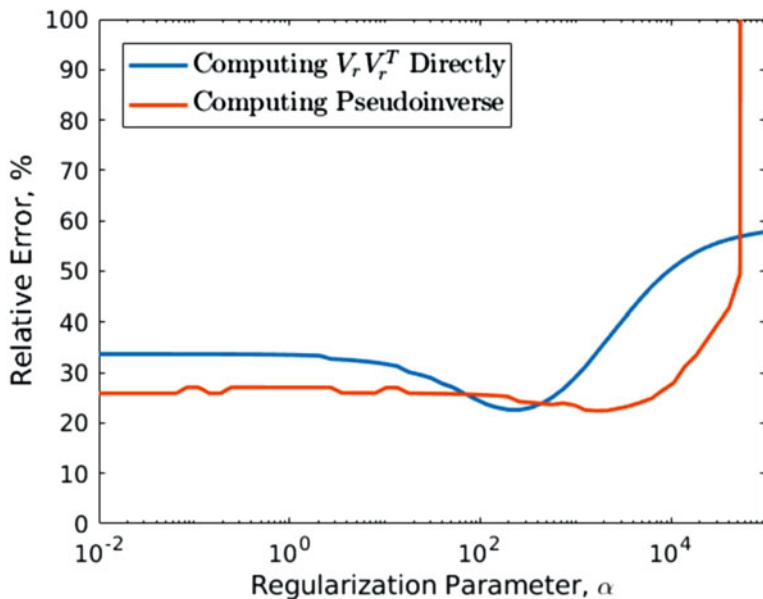


Fig. 14 A comparison between variant (b) (red curve) and variant (a) with $r = 200$ (blue curve). Here, we compute the relative error of the reconstruction and the truth image for various values of regularization parameter α

the pseudoinverse formulation, we can still get excellent results while avoiding the computation of a large factorization.

Conclusions

We have presented a new regularization technique called data-informed (DI) regularization that, though with disintegration origin, can be viewed as a combination of the classical truncated SVD and Tikhonov regularization. In particular, the DI

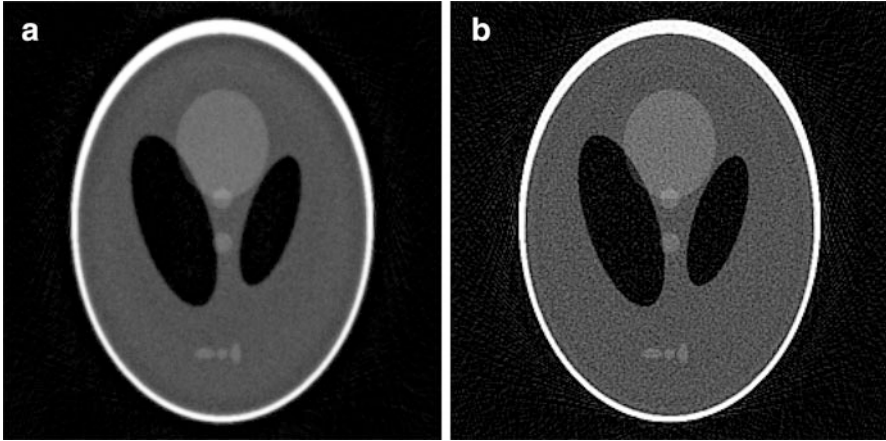


Fig. 15 X-Ray tomography reconstruction with 1% noise and $\alpha = 100$: (a) the result from the a)-variant with $r = 200$ and (b) the result from the b)-variant

Algorithm 2 Data-informed inversion using nested CG

Input: Data y , number of eigenvectors r , prior x_0 , prior covariance matrix Γ , noise covariance matrix Λ , regularization parameter α

- 1: Define $F = \Lambda^{-\frac{1}{2}} A \Gamma^{\frac{1}{2}}$.
- 2: Create functions to compute matrix-vector products Fx and $F^T x$.
- 3: Solve linear equation (18), i.e.,

$$\Gamma^{-\frac{1}{2}} \left[F^T F + \alpha (I - F^T (F F^T)^\dagger F) \right] \Gamma^{-\frac{1}{2}} x = \Gamma^{-\frac{1}{2}} F^T y + \alpha \Gamma^{-\frac{1}{2}} (I - F^T (F F^T)^\dagger F) \Gamma^{-\frac{1}{2}} x_0$$

using the conjugate gradient method. For each CG iterations, compute the product of $F^T (F F^T)^\dagger F \Gamma^{-\frac{1}{2}}$ with any vector x using matrix-free Algorithm 3.

Algorithm 3 Compute the product of $F^T (F F^T)^\dagger F \Gamma^{-\frac{1}{2}}$ with any vector using optimization

Input: functions to compute Fx and $F^T x$, current estimate of x , prior covariance matrix Γ

- 1: Compute $b = F \Gamma^{-\frac{1}{2}} x$.
- 2: Using conjugate gradient method, solve linear equation

$$F F^T z = b.$$

- 3: Return $F^T z$.
-

approach does not pollute the data-informed modes and regularizes only less data-informed ones. As a direct consequence, the DI approach is at least as good as the Tikhonov method for any value of the regularization parameter, and it is

more accurate than the TSVD (for reasonable regularization parameter). Due to the blending of these two classical methods, DI is expected to be robust with regularization parameter, and this is verified numerically. We have shown that DI is a regularization strategy. The DI approach has an interesting statistical interpretation, that is, it transforms both the data distribution (i.e., the likelihood) and prior distribution (induced by Tikhonov regularization) to the same Gaussian distribution whose covariance matrix is diagonal, and the diagonal elements are exactly the singular values of a composition of the prior covariance matrix, the forward map, and the noise covariance matrix. In other words, DI finds the modes that are most equally data-informed and prior-informed and leaves these modes untouched so that the inverse solution receives the best possible (balanced) information from both prior and the data. Furthermore, the DI approach takes the data uncertainty into account and hence can avoid overconfident uncertainty estimation. To demonstrate and to support our deterministic and statistical findings, we have presented various results for popular computer vision and imaging problems including deblurring, denoising, and X-ray tomography.

References

- Antoulas, A.C.: *Approximation of Large-Scale Systems*. SIAM, Philadelphia (2005)
- Babacan, S.D., Mancera, L., Molina, R., Katsaggelos, A.K.: Non-convex priors in bayesian compressed sensing. In: 2009 17th European Signal Processing Conference, pp. 110–114 (2009)
- Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.* **18**, 2419–2434 (2009)
- Boley, D.: Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs. *SIAM J. Optim.* **23**, 2183–2207 (2013)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2010)
- Chartrand, R., Wohlberg, B.: A nonconvex admm algorithm for group sparsity with sparse groups. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6009–6013 (2013)
- Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3869–3872 (2008)
- Colton, D., Kress, R.: *Integral Equation Methods in Scattering Theory*. Wiley (1983)
- Colton, D., Kress, R.: *Inverse Acoustic and Electromagnetic Scattering*, 2nd edn. Applied Mathematical Sciences, Vol. 93. Springer, Berlin/Heidelberg/New-York/Tokyo (1998)
- Franklin, J.N.: Well-posed stochastic extensions of ill-posed linear problems. *J. Math. Anal. Appl.* **31**, 682–716 (1970)
- Goldstein, T., Osher, S.: The slit Bregman method for L1-regularized problems. *SIAM J. Imag. Sci.* **2**, 323–343 (2009)
- Golub, G., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. **21**, 215–223 (1979)
- Gugercin, S., Antoulas, A.C.: A survey of model reduction by balanced truncation and some new results. *Int. J. Control.* **77**, 748–766 (2004)
- Hansen, P.C.: Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM J. Sci. Stat. Comput.* **11**, 503–518 (1990)
- Hansen, P.C.: Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Rev.* **34**, 561–580 (1992)

- Hansen, P.C., Nagy, J.G., O’Leary, D.P.: *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia (2006)
- Hansen, P.C., O’Leary, D.P.: The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.* **14**, 1487–1503 (1993)
- Kaipio, J., Somersalo, E.: *Statistical and Computational Inverse Problems*, vol. 160 of Applied Mathematical Sciences. Springer, New York (2005)
- Kirsch, A.: *An Introduction to the Mathematical Theory of Inverse Problems*, 2nd edn. Applied Mathematical Sciences, Vol. 120. Springer, New-York (2011)
- Lasanen, S.: *Discretizations of generalized random variables with applications to inverse problems*, Ph.D. thesis, University of Oulu (2002)
- Lehtinen, M.S., Päivärinta, L., Somersalo, E.: Linear inverse problems for generalized random variables. *Inverse Prob.* **5**, 599–612 (1989)
- Morozov, V.A.: On the solution of functional equations by the method of regularization. *Soviet Math. Dokl.* (1966)
- Mueller, J.L., Siltanen, S.: *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, Philadelphia (2012)
- Nikolova, M.: Weakly constrained minimization: Application to the estimation of images and signals involving constant regions. *J. Math. Imaging Vision* **21**, 155–175 (2004)
- Nikolova, M.: Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. *Multiscale Model. Simul.* **4**, 960–991 (2005) (electronic)
- Piironen, P.: *Statistical measurements, experiments, and applications*, Ph.D. thesis, Department of Mathematics and Statistics, University of Helsinki (2005)
- Ramirez-Giraldo, J., Trzasko, J., Leng, S., Yu, L., Manduca, A., McCollough, C.H.: Nonconvex prior image constrained compressed sensing (ncpiccs): Theory and simulations on perfusion ct. *Med. Phys.* **38**, 2157–2167 (2011)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**, 259–268 (1992)
- Shewchuk, J.R.: An introduction of the conjugate gradient method without the agonizing pain, Carnegie Mellon University (1994). <https://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>
- Stuart, A.M.: Inverse problems: A Bayesian perspective. *Acta Numerica* **19**, 451–559 (2010). <https://doi.org/10.1017/S0962492910000061>
- Tarantola, A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia (2005)
- Tikhonov, A.N., Arsenin, V.A.: *Solution of Ill-posed Problems*. Winston & Sons, Washington, DC (1977)



Randomized Kaczmarz Method for Single Particle X-Ray Image Phase Retrieval

36

Yin Xian, Haiguang Liu, Xuecheng Tai, and Yang Wang

Contents

Introduction	1274
The Phase Retrieval Problem	1274
Challenges of X-Ray Data Processing	1275
Phase Retrieval with Noisy or Incomplete Measurements	1276
Outline	1276
Background: Phase Retrieval and Stochastic Optimization	1277
Phase Retrieval	1277
Stochastic Optimization and the Kaczmarz Method	1278
Variance-Reduced Randomized Kaczmarz (VR-RK) Method	1279
Application: Robust Phase Retrieval of the Single-Particle X-Ray Images	1281
Synthetic Single-Particle Data Recovery Experiment	1281
Recovery Efficiency Under Constraints	1282
Results of the PR772 Dataset	1283
Conclusion	1284
Appendix	1284
References	1286

Y. Xian (✉)

TCL Research Hong Kong, Hong Kong, SAR, China

e-mail: polinexian@tcl.com

H. Liu

Microsoft Research-Asian, Beijing, China

e-mail: haiguangliu@microsoft.com

X. Tai

Hong Kong Center for Cerebro-cardiovascular Health Engineering (COCHE), Shatin, Hong Kong, China

e-mail: xtai@hkcoche.org

Y. Wang

Hong Kong University of Science and Technology, Hong Kong, SAR, China

e-mail: yangwang@ust.hk

Abstract

In this chapter, we investigate phase retrieval algorithm for the single-particle X-ray imaging data. We present a variance-reduced randomized Kaczmarz (VR-RK) algorithm for phase retrieval. The VR-RK algorithm is inspired by the randomized Kaczmarz method and the Stochastic Variance Reduce Gradient Descent (SVRG) algorithm. Numerical experiments show that the VR-RK algorithm has a faster convergence rate than randomized Kaczmarz algorithm and the iterative projection phase retrieval methods, such as the hybrid input output (HIO) and the relaxed averaged alternating reflections (RAAR) methods. The VR-RK algorithm can recover the phases with higher accuracy, and is robust at the presence of noise. Experimental results on the scattering data from individual particles show that the VR-RK algorithm can recover phases and improve the single-particle image identification.

Keywords

Stochastic optimization · Variance reduction · Phase retrieval · Randomized Kaczmarz algorithm

Introduction

The Phase Retrieval Problem

The mathematical formulation of phase retrieval is solving a set of quadratic equations. Methods to solve the phase retrieval problem can be classified into two categories: convex and non-convex approaches. Convex methods, like *PhaseLift* (Candès et al. 2013), convert the quadratic system equation to a linear system equation through a matrix-lifting technique. The *PhaseMax* method (Goldstein and Studer 2017; Bahmani and Romberg 2017) operates in the original signal space rather than lifting it to a higher dimensional space. It replaces the non-convex constraints with inequality constraints that define convex sets. The convex approaches have good recovery guarantees, but their computational complexities are usually high when the dimension of the signals is large.

On the other hand, the non-convex approaches turn the phase retrieval into an optimization problem. The most popular class of methods is based on alternate projection, such as the hybrid input output (HIO) method (Bauschke et al. 2003), the error reduction (ER) method (Fienup and Wackerman 1986), and the relaxed averaged alternating reflections (RAAR) method (Luke 2004). These methods are iterative projection methods, since they involve iterative projections onto the constraint sets. Unlike the convex approaches, convergence is not guaranteed for these algorithms, and stagnation may occur due to nonuniqueness of the solution (Fienup and Wackerman 1986). A unified evaluation of these iterative projection algorithms can be found in the paper of Marchesini (2007). Recently, a method called Wirtinger flow (Candès et al. 2015) is proposed. It works well with spectral method for

initialization. The follow-up works include the truncated Wirtinger flow (Chen and Candès 2017), truncated amplitude flow (Wang et al. 2017), and reshaped Wirtinger flow (Zhang and Liang 2016). These methods have less computational complexities and have theoretical convergence guarantees.

The randomized Kaczmarz algorithm is introduced to solve the phase retrieval problem by Wei (2015). The randomized Kaczmarz method can be viewed as a special case of the stochastic gradient descent (SGD) (Needell et al. 2014). For the phase retrieval problem, the method is essentially SGD for the amplitude flow objective. It was shown numerically that the method outperforms the Wirtinger flow and the ER method (Wei 2015). The convergence rate of the randomized Kaczmarz method for the linear system is studied in the paper of Strohmer and Vershynin (2009). The theoretical justification of using randomized Kaczmarz method for phase retrieval has been presented in the paper of Tan and Vershynin (2019).

Challenges of X-Ray Data Processing

The structure of biological macromolecules is the key to understand the living cell function and behavior. The Protein Data Bank (PDB) (Bernstein et al. 1977) currently has more than 173,110 structures, but many structures of biological molecules and their complexes have not been determined. The cryo-electron microscopy (Cryo-EM) and the X-ray crystallography have been successfully applied in this field. The X-ray crystallography has solved about 90% of these structures. However, growing high-quality crystals of biomolecules is challenging, especially for biologically functional molecules. Therefore, determining structures from single molecules are appealing.

The use of the X-ray free electron lasers (XFEL) is a recent development in structure biology. The idea behind this method is to record the instantaneous elastic scattering from an ultrashort pulse. The pulse is so brief that it terminates before the onset of radiation damage (“diffract before destroy”) (Liu and Spence 2016). With this application, the single-particle imaging becomes possible, even at room temperature. It allows one to understand the structures and dynamics of macromolecules.

The difference between the Cryo-EM and the X-ray crystallography is that the Cryo-EM data includes phase information of the structural factors, while the X-ray crystallographic diffraction data only provide amplitude information but lack phase information (Wang and Wang 2017; Scheres 2012). The illustrations and data processing examples are shown in the paper of Sorzano et al. (2004), Xian et al. (2018), and Gu et al. (2020). In order to solve the biological structures, the phase information is essential. It is normally obtained by experimental or computational means.

The challenges of XFEL single-particle imaging also include the following: (i) the signal-to-noise ratio (SNR) is low, and the information is influenced by noise; (ii) the orientation of each sample particle is unknown, leading to the difficulty in data merging and 3D reconstruction; (iii) conformational heterogeneity places a hurdle for single-particle identification and reconstruction (Wang and Wang 2017).

In this chapter, we investigate the phase retrieval algorithms of the XFEL data. The baseline for a good phase retrieval algorithm is its robustness against noises and the incompleteness of information (Shi et al. 2019).

Phase Retrieval with Noisy or Incomplete Measurements

The number of photons detected by the optical sensor is of Poisson distribution. For the phase retrieval problem contaminated by the Poisson noise, or has incomplete magnitude information, the prior information is crucial to process the data. Research for imposing prior information to image processing is shown in the literature (Le et al. 2007; Zhang et al. 2012; Hunt et al. 2018).

In order to better reconstruct the data, one can consider a variational model by introducing a total variation (TV) regularization, which is widely used in imaging processing community. TV regularization can enable recovery of signals from incomplete or limited measurements. The alternating direction of multipliers method (ADMM) (Glowinski and Le Tallec 1989; Wu and Tai 2010) and the split Bregman method (Goldstein and Osher 2009) is usually applied to solve the TV-regularization problem. They have been applied in the phase retrieval problem (Chang et al. 2016, 2018; Bostan et al. 2014; Li et al. 2016).

Besides TV regularization, Tikhonov regularization is another important smoothing techniques in variational image denoising. It is often applied in noise removal. The phase retrieval problem with a Tikhonov regularization has been solved by the Gauss-Newton method (Seifert et al. 2006; Sixou et al. 2013; Langemann and Tasche 2008; Ramos et al. 2019). Considering the sparsity constraints, the fixed point iterative approach (Fornasier and Rauhut 2008; Tropp 2006; Ma et al. 2018) has been applied for the problem with nonlinear joint sparsity regulation.

Outline

In this chapter, we further advance the convergence speed of the randomized Kaczmarz method for phase retrieval. The idea comes from the fact that the randomized Kaczmarz method is a weighted SGD, and the convergence rate of SGD is slower because of the random sampling variance. Therefore, reducing the sampling variance can improve the convergence rate of the randomized Kaczmarz method. Inspired by the stochastic variance reduce gradient (SVRG) method (Johnson and Zhang 2013), we present the variance-reduced randomized Kaczmarz method (VR-RK) for single-particle X-ray imaging phase retrieval. Considering the sparsity constraint and generality of the problem, we present the VR-RK method under both the L_1 and the L_2 constraints for computational analysis. Numerical results on the virus data show that the VR-RK method can recover information with higher accuracy at a faster convergence rate. It helps recover the lost information due to the beam stop for blocking the incidence X-ray beam.

The rest of the chapter is organized as follows. In the section “[Background: Phase Retrieval and Stochastic Optimization](#),” we give a general overview of phase retrieval and stochastic optimization. In the section “[Variance-Reduced Randomized Kaczmarz \(VR-RK\) Method](#),” the proposed variance-reduced randomized Kaczmarz method, and its variation under L_1 and L_2 constraints are presented. The evaluation of the algorithm is shown in the “[Application: Robust Phase Retrieval of the Single-Particle X-Ray Images](#)” section, and the single-particle X-ray image data are tested. The “[Conclusion](#)” section concludes the chapter.

Background: Phase Retrieval and Stochastic Optimization

Phase Retrieval

Formulation of the phase retrieval problem is as follows:

$$\min_x \sum_{k=1}^m (y_k - |\langle \mathbf{a}_k, \mathbf{x} \rangle|^2)^2. \quad (1)$$

where \mathbf{y} is the measurement, \mathbf{x} is the signal that need to be recovered, and \mathbf{a}_k is the measurement operating vector. In the setting of forward X-ray scattering imaging at the far field, \mathbf{a}_k is a Fourier vector, and \mathbf{y} is a diffraction pattern of the target. The problem in phase retrieval is the limitation of optical sensors, which measures only the intensity.

The loss function of Eq. (1) is expressed as the squared difference between measurement intensities and the modelled intensities. It is a system of quadratic equation, and therefore, it is a non-convex problem.

To solve Eq. (1), the alternate projection methods are often used, such as HIO, ER, and RAAR methods as mentioned previously. These algorithms can be expressed in the form of fixed-point equation. They can be implemented jointly to better avoid local minima.

When the loss function is expressed as the squared loss of amplitudes, the formulation can be written as:

$$\min_x \sum_{k=1}^m (\sqrt{y_k} - |\langle \mathbf{a}_k, \mathbf{x} \rangle|)^2. \quad (2)$$

To solve Eq. (2), it is possible to apply the amplitude flow algorithm (Wang et al. 2017), which is essentially a gradient descent algorithm that can converge under good initialization.

Stochastic Optimization and the Kaczmarz Method

The phase retrieval problem can be solved by stochastic optimization approaches. For the problem:

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{k=1}^m f_k(\mathbf{x}), \quad (3)$$

the gradient descent method updating rule is: $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{t_k}{m} \sum_{k=1}^m \nabla f_k(\mathbf{x}_k)$, where t_k is the step size at each iteration and m is the number of samples, or the number of measurements in the phase retrieval setting. The gradient descent is expensive, and it requires evaluation of n derivatives at each iteration. To reduce the computational cost, the SGD is proposed:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f_{i_k}(\mathbf{x}_k) \quad (4)$$

where i_k is an index chosen uniformly in random from $\{1, \dots, m\}$ at each iteration. The computational cost is $1/m$ of the standard gradient descent. The SVRG is proposed to reduce variance of SGD and has a faster convergence rate (Johnson and Zhang 2013). It is operated in epochs. In each epoch, the updating process is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \left(\nabla f_{i_k}(\mathbf{x}_k) - \nabla f_{i_k}(\bar{\mathbf{x}}) + \frac{1}{m} \sum_{i=1}^m \nabla f_i(\bar{\mathbf{x}}) \right) \quad (5)$$

where η is the step size, and $\bar{\mathbf{x}}$ is a snapshot value in each epoch (Johnson and Zhang 2013).

The Kaczmarz method is a well-known iterative method for solving a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. The classical Kaczmarz method sweeps through the rows in \mathbf{A} in a cyclic manner and projects the current estimate onto a hyperplane associated with the row of \mathbf{A} to get the new estimate. The randomized Kaczmarz method randomly chooses the row for projection in each iteration:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{b_{i_k} - \langle \mathbf{a}_{i_k}, \mathbf{x}_k \rangle}{\|\mathbf{a}_{i_k}\|_2^2} \mathbf{a}_{i_k} \quad (6)$$

where \mathbf{a}_{i_k} is the row of \mathbf{A} . The randomized Kaczmarz can be viewed as a reweighted SGD with importance sampling for the least squares problem (Needell et al. 2014):

$$F(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \frac{1}{2} \sum_{i=1}^m (\mathbf{a}_i^T \mathbf{x} - b_i)^2. \quad (7)$$

The randomized Kaczmarz algorithm is essentially stochastic gradient descent for the amplitude flow problem in Eq. (2). This suggests that the acceleration schemes for SGD, such as the variance-reduced approach, can be applied to the algorithm and improve phase retrieval.

Variance-Reduced Randomized Kaczmarz (VR-RK) Method

Define $b_{i_k} = \sqrt{y_{i_k}}$; the formulation of Eq. (2) can be written as:

$$\min_{\mathbf{x}} \sum_{k=1}^m (b_k - |\langle \mathbf{a}_k, \mathbf{x} \rangle|)^2. \quad (8)$$

The update scheme for randomized Kaczmarz for the phase retrieval objective of Eq. (8), according to the paper of Tan and Vershynin (2019), is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{a}_{i_k} \quad (9)$$

where

$$\eta_k = \frac{\text{sign}(\langle \mathbf{a}_{i_k}, \mathbf{x}_k \rangle) b_{i_k} - \langle \mathbf{a}_{i_k}, \mathbf{x}_k \rangle}{\|\mathbf{a}_{i_k}\|_2^2}$$

i_k is drawn independently and identically distributed (i.i.d.) from the index set $\{1, 2, \dots, m\}$ with the probability

$$g_k = \frac{\|\mathbf{a}_{i_k}\|^2}{\|\mathbf{A}\|_F^2}. \quad (10)$$

The VR-RK method is inspired by the randomized Kaczmarz method and the SVRG method. It is proposed originally to solve the linear system equation (Jiao et al. 2017). Let $f_i(\mathbf{x}) = \frac{1}{2}(\mathbf{a}_i^T \mathbf{x} - b_i)^2$, and let

$$h_i(\mathbf{x}) = \frac{f_i(\mathbf{x})}{g_i} = \frac{1}{2}(\mathbf{a}_i^T \mathbf{x} - b_i)^2 \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{a}_i\|^2} \quad (11)$$

then,

$$\nabla h_i(\mathbf{x}) = (\mathbf{a}_i^T \mathbf{x} - \text{sign}(\mathbf{a}_i^T \mathbf{x}) b_i) \mathbf{a}_i \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{a}_i\|^2} \quad (12)$$

Let $\mu_i(\mathbf{x}) = \nabla h_i(\mathbf{x})$, and s be the size of the epoch. The variance-reduced randomized Kaczmarz algorithm for phase retrieval is shown in Algorithm 1.

Algorithm 1 Variance-reduced randomized Kaczmarz (VR-RK)

Initialize $\mu_i(\bar{\mathbf{x}}) = 0$, and $\bar{\boldsymbol{\mu}} = 0$, specify \mathbf{A} , \mathbf{b} , s .

At steps $k = 1, 2, \dots$, if $k \bmod s = 0$, then

$$\bar{\mathbf{x}} = \mathbf{x}_k \text{ and } \bar{\boldsymbol{\mu}} = \mu(\mathbf{x}_k)$$

Pick index i uniformly at random according to (10).

Update \mathbf{x}_k by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{m}{\|\mathbf{A}\|_F^2} (\mu_{i_k}(\mathbf{x}_k) - \mu_i(\bar{\mathbf{x}}) + \bar{\boldsymbol{\mu}})$$

$$\text{where } \bar{\boldsymbol{\mu}} = \frac{1}{m} \sum_{i=1}^m \nabla h_i(\bar{\mathbf{x}})$$

Considering the generality of the problem, and L_2 constraint is imposed, the objective function is:

$$\min_{\mathbf{x}} \frac{1}{2} \sum_{k=1}^m (b_k - |\langle \mathbf{a}_k, \mathbf{x} \rangle|)^2 + \gamma \|\mathbf{x}\|_2. \quad (13)$$

Applying the randomized Kaczmarz method, according to Hefny et al. (2017), the updating process becomes:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{(\mathbf{a}_{i_k}^T \mathbf{x}_k - \text{sign}(\mathbf{a}_{i_k}^T \mathbf{x}_k) b_{i_k}) \mathbf{a}_{i_k} + \gamma \mathbf{x}_k}{\|\mathbf{a}_{i_k}\|^2 + \gamma} \quad (14)$$

In the VR-RK setting, the updating process is:

$$\nabla c_{i_k}(\mathbf{x}_k) = \frac{(\mathbf{a}_{i_k}^T \mathbf{x}_k - \text{sign}(\mathbf{a}_{i_k}^T \mathbf{x}_k) b_{i_k}) \mathbf{a}_{i_k} + \gamma \mathbf{x}_k}{\|\mathbf{a}_{i_k}\|^2 + \gamma} \quad (15)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \nabla c_{i_k}(\mathbf{x}_k) + \nabla c_{i_k}(\bar{\mathbf{x}}) - \frac{1}{m} \sum_{i=1}^m \nabla c_i(\bar{\mathbf{x}}) \quad (16)$$

For the consideration of the sparsity, the L_1 instead of the L_2 constraint can be imposed; then the objective function becomes:

$$\min_{\mathbf{x}} \frac{1}{2} \sum_{k=1}^m (b_k - |\langle \mathbf{a}_k, \mathbf{x} \rangle|)^2 + \lambda \|\mathbf{x}\|_1. \quad (17)$$

To deal with this formula, the majorization-minimization (MM) technique and the C-PRIME method (Qiu and Palomar 2017) are employed. It is shown that the problem is equivalent to:

$$\min_{\mathbf{x}} \left(C \|\mathbf{x} - \mathbf{d}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right) \quad (18)$$

where C is a constant and $C \geq \rho_{\max}(\mathbf{A}^H \mathbf{A})$, ρ_{\max} is the largest eigenvalue of a matrix, and \mathbf{d} is the constant vector that is defined as:

$$\mathbf{d} := \mathbf{x}_k - \frac{1}{C} \mathbf{A}^H (\mathbf{A} \mathbf{x}_k - \mathbf{b} \odot e^{j\angle(\mathbf{A} \mathbf{x}_k)}). \quad (19)$$

Above, the notation \odot is the element-wise Hadamard product of two vectors, and \angle is the phase angle. The close form solution of \mathbf{x} is:

$$\mathbf{x}^* = e^{j\angle(\mathbf{d})} \odot \max \left\{ |\mathbf{d}| - \frac{\lambda}{2C} \mathbf{1}, \mathbf{0} \right\}.$$

Application: Robust Phase Retrieval of the Single-Particle X-Ray Images

In this section, we present numerical results of phase retrieval of the single-particle X-ray imaging data.

Synthetic Single-Particle Data Recovery Experiment

The first experiment is to test the reconstruction efficiency of the virus data, as shown in Fig. 1. The image size of Fig. 1a is 755×755 pixels, and the pixel values are normalized to $[0,1]$. The diffraction pattern (Fig. 1b) is created by taking the Fourier transform of Fig. 1a. In this experiment, X-ray scattering signals are mainly observed at low resolutions, corresponding to low frequencies in Fourier space. A gap is placed in the center of the diffraction pattern to allow the incident beam to pass through, to avoid damaging or saturating detector sensors. The gap results in an information loss at low-frequency regime, as shown in Fig. 1c. The low-frequency information corresponds to the overall shape of the object. Without which, it poses a challenge for reconstruction.

We reconstruct the sample virus image from the diffraction pattern with detector gap in Fig. 1c. The VR-RK, randomized Kaczmarz, HIO, and RAAR methods are tested in the MATLAB platform. In order to reconstruct the data, a reference signal is used as a priori for preprocessing as described in the paper of Barmherzig et al. (2019), and the numerical iteration is then performed. Comparison of convergence rates and the relative square errors is shown in Fig. 2 and Table 1. The relative square error is defined by: $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 / \|\mathbf{x}\|^2$, where \mathbf{x} is the ground truth image and $\hat{\mathbf{x}}$ is the reconstructed image. The experiment shows that the VR-RK algorithm has a faster convergence rate and a better reconstruction accuracy compared with the randomized Kaczmarz algorithm and the iterative projection algorithms.

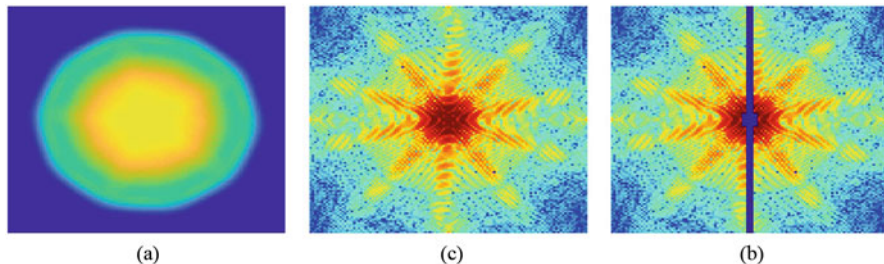


Fig. 1 Virus sample particle and its diffraction patterns (Li 2016). (a) Virus particle 2D projection imaging in real space. (b) Simulated X-ray data. (c) The simulated data with a gap. The size of pixels in the gap is 409

Fig. 2 Comparison of convergence rate

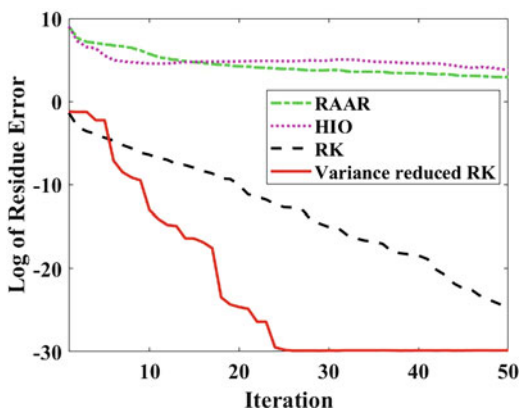


Table 1 Reconstruction error comparison

	VR-RK	RK	RAAR	HIO
Error	1.7540e-12	6.8635e-12	0.0307	0.1313

Recovery Efficiency Under Constraints

To further illustrate the convergence rate, we compare the VR-RK algorithm and the randomized Kaczmarz algorithm under L_1 and L_2 constraints on reconstructing the virus sample data. The cost function changes per iteration are shown in Fig. 3. From the figure, the loss function decays faster in VR-RK than randomized Kaczmarz method.

Considering that the single-particle X-ray imaging data are influenced by the Poisson noise, we examine the reconstruction accuracy at various noise levels, with ϵ from 0.005 to 0.1, and the measurement under the noise: $\mathbf{y} = |\mathbf{Ax}|^2(1 + \epsilon)$.

Table 2 shows the relative square error of reconstruction using different phase retrieval algorithms in various noise levels. From Table 2, we can see that the VR-RK method outperforms other algorithms under noise.

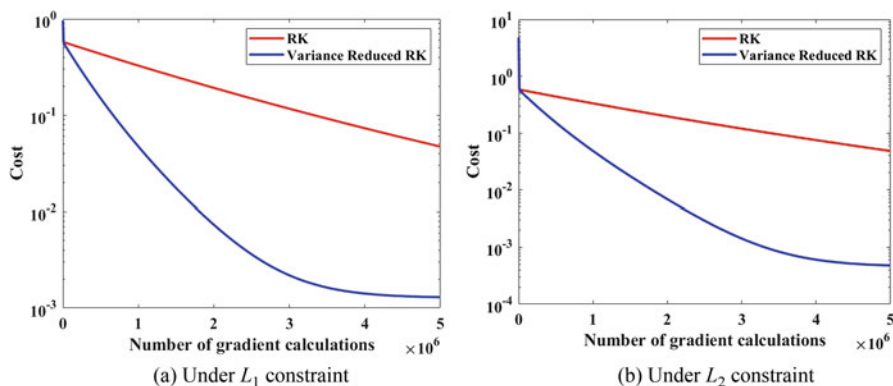


Fig. 3 Comparison of convergence rate. (a) Under L_1 constraint. (b) Under L_2 constraint

Table 2 Relative square error comparison

	VR-RK- L_2	VR-RK- L_1	RAAR	HIO
$\epsilon = 0.1$	0.3687	0.3685	1.2502	0.7315
$\epsilon = 0.05$	0.2438	0.2432	0.8775	0.5398
$\epsilon = 0.01$	0.1007	0.1013	0.3860	0.3150
$\epsilon = 0.005$	0.0707	0.0712	0.2703	0.2130

Results of the PR772 Dataset

We test the VR-RK algorithm on the PR772 particle dataset (Reddy et al. 2017). The image size is 256×256 pixels, and the pixel values are scaled to the range of $[0, 255]$. Illustration of the diffraction pattern of the single-particle data is shown in Fig. 4a and e.

For this dataset, the shrinkwrap method is applied to obtain a tight object support (Shi et al. 2019; Marchesini et al. 2003), and the square root of the diffraction intensities is used as a reference for the missing pixels during numerical iteration. A recovery example is shown in Fig. 4, and more recovery examples are presented in the supplementary materials.

We use the VR-RK algorithm and the RAAR and HIO methods to recover the data and classify the single-particle scattering pattern data and the non-single-particle scattering pattern data. We use the VR-RK for computation. There are 497 samples with labels in the validation set (Shi et al. 2019). Among them, 208 are single-particle samples, and 289 are non-single-particle samples. We use ISOMAP for data compression and clustering and KNN for classification. We use fourfold cross-validation. The VR-RK has the best result. The AUCs of the binary classification results are listed as follows (Table 3).

From the results, we can see that the VR-RK method can help recover the data and improve classification rate.

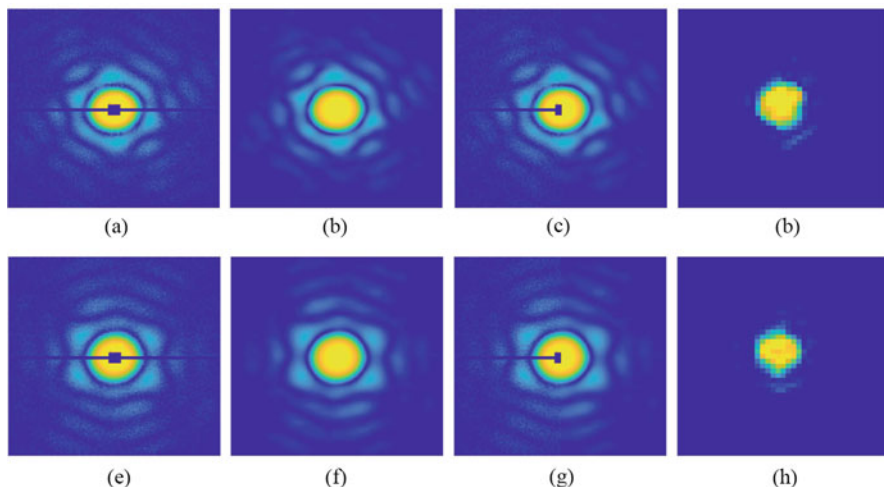


Fig. 4 PR772 single-particle scattering pattern phase retrieval. (a) and (e) are two single-particle diffraction patterns; (b) and (f) are the recovered diffraction patterns of (a) and (e), respectively; (c) and (g) show the comparison of the original and the recovered diffraction patterns, the left half is the original, and the right half is the recovered; (d) and (h) are the real-space images reconstructed using VR-RK algorithm from (a) and (e)

Table 3 AUC of binary classification

	VR-RK	RAAR	HIO
AUC	0.9501	0.9069	0.9231

Conclusion

In this chapter, we present the variance-reduced randomized Kaczmarz (VR-RK) method for XFEL single-particle phase retrieval. The VR-RK method is inspired by the randomized Kaczmarz method and the SVRG method. It is proposed in order to accelerate the convergence speed of the algorithm. Numerical results show that the VR-RK method has faster convergence rate and better accuracy under noises. Experiments on PR772 single-particle X-ray imaging data show that the VR-RK method can help recover and classify particles.

Appendix

For the PR772 dataset, further examples of phase retrieval recovery are shown here. Figure 5a and b are examples of 25 diffraction pattern sample reconstructions. Figure 5c shows the corresponding real-space recovered images.

Figures 6 and 7 are examples of 100 diffraction pattern samples reconstruction. Figure 8 shows the corresponding real space recovered images.

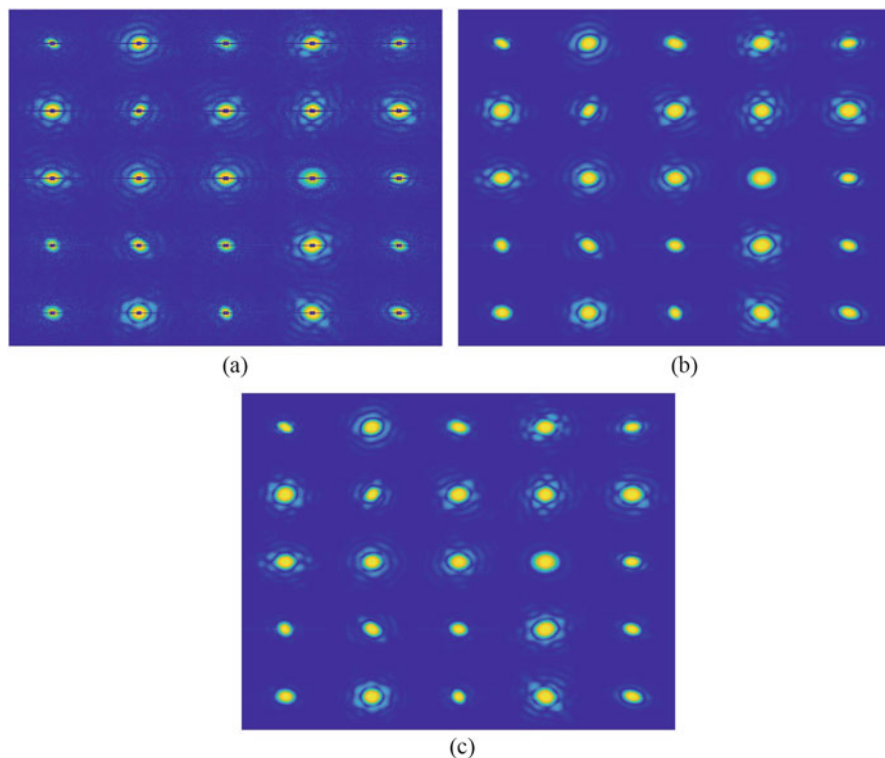
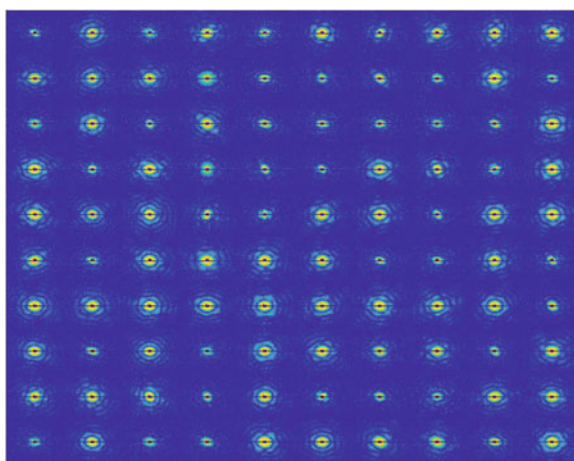


Fig. 5 Phase retrieval of the PR772 dataset. (a) Original data diffraction pattern illustrations. (b) Recovered image diffraction pattern illustrations. (c) Recovered real-space data illustrations

Fig. 6 Original data diffraction pattern illustrations



Acknowledgments Tai is supported by NSFC/RGC Joint Research Scheme (N_HKBU214/19), Initiation Grant for Faculty Niche Research Areas (RC-FNRA-IG/19-20/SCI/01) and CRF (C1013-21GF).

Fig. 7 Recovered image diffraction pattern illustrations

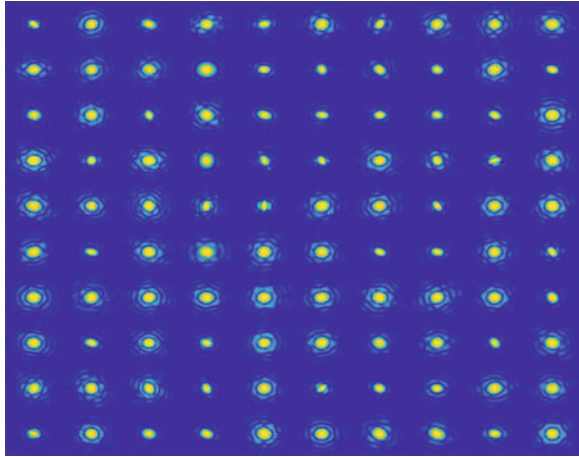
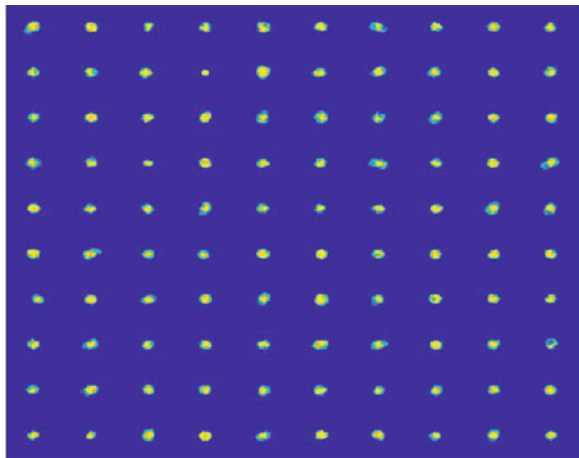


Fig. 8 Recovered real-space data illustrations



References

- Bahmani, S., Romberg, J.: Phase retrieval meets statistical learning theory: a flexible convex relaxation. In: *Artificial Intelligence and Statistics*, pp. 252–260. PMLR (2017)
- Barmherzig, D., Sun, J., Li, P., Lane, T.J., Candès, E.: Holographic phase retrieval and reference design. *Inverse Probl.* **35**(9), 094001 (2019)
- Bauschke, H., Combettes, P., Luke, R.: Hybrid projection–reflection method for phase retrieval. *JOSA A* **20**(6), 1025–1034 (2003)
- Bernstein, F., Koetzle, T., Williams, G., Meyer Jr, E., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T., Tasumi, M.: The protein data bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.* **112**(3), 535–542 (1977)
- Bostan, E., Froustey, E., Rappaz, B., Shaffer, E., Sage, D., Unser, M.: Phase retrieval by using transport-of-intensity equation and differential interference contrast microscopy. In: *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 3939–3943. IEEE (2014)

- Candès, E., Strohmer, T., Voroninski, V.: Phaselift: exact and stable signal recovery from magnitude measurements via convex programming. *Commun. Pure Appl. Math.* **66**(8), 1241–1274 (2013)
- Candès, E., Li, X., Soltanolkotabi, M.: Phase retrieval via Wirtinger flow: theory and algorithms. *IEEE Trans. Inf. Theory* **61**(4), 1985–2007 (2015)
- Chang, H., Lou, Y., Ng, M., Zeng, T.: Phase retrieval from incomplete magnitude information via total variation regularization. *SIAM J. Sci. Comput.* **38**(6), A3672–A3695 (2016)
- Chang, H., Lou, Y., Duan, Y., Marchesini, S.: Total variation–based phase retrieval for poisson noise removal. *SIAM J. Imag. Sci.* **11**(1), 24–55 (2018)
- Chen, Y., Candès, E.: Solving random quadratic systems of equations is nearly as easy as solving linear systems. *Commun. Pure Appl. Math.* **70**(5), 822–883 (2017)
- Fienup, J., Wackerman, C.: Phase-retrieval stagnation problems and solutions. *JOSA A* **3**(11), 1897–1907 (1986)
- Fornasier, M., Rauhut, H.: Recovery algorithms for vector-valued data with joint sparsity constraints. *SIAM J. Numer. Anal.* **46**(2), 577–613 (2008)
- Glowinski, R., Le Tallec, P.: *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia (1989)
- Goldstein, T., Osher, S.: The split bregman method for l1-regularized problems. *SIAM J. Imag. Sci.* **2**(2), 323–343 (2009)
- Goldstein, T., Studer, C.: Convex phase retrieval without lifting via phasemax. In: *International Conference on Machine Learning*, pp. 1273–1281. PMLR (2017)
- Gu, H., Xian, Y., Unarta, I., Yao, Y.: Generative adversarial networks for robust Cryo-EM image denoising. *arXiv preprint arXiv:2008.07307* (2020)
- Hefny, A., Needell, D., Ramdas, A.: Rows versus Columns: Randomized Kaczmarz or Gauss–Seidel for Ridge Regression. *SIAM J. Sci. Comput.* **39**(5), S528–S542 (2017)
- Hunt, X., Reynaud-Bouret, P., Rivoirard, V., Sansonnet, L., Willett, R.: A data-dependent weighted LASSO under poisson noise. *IEEE Trans. Inf. Theory* **65**(3), 1589–1613 (2018)
- Jiao, Y., Jin, B., Lu, X.: Preasymptotic convergence of randomized Kaczmarz method. *Inverse Probl.* **33**(12), 125012 (2017)
- Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. *Adv. Neural Inf. Process. Syst.* **26**, 315–323 (2013)
- Langemann, D., Tasche, M.: Phase reconstruction by a multilevel iteratively regularized gauss-newton method. *Inverse Probl.* **24**(3), 035006 (2008)
- Le, T., Chartrand, R., Asaki, T.: A variational approach to reconstructing images corrupted by poisson noise. *J. Math. Imag. Vis.* **27**(3), 257–263 (2007)
- Li, F., Abascal, J., Desco, M., Soleimani, M.: Total variation regularization with split Bregman-based method in magnetic induction tomography using experimental data. *IEEE Sens. J.* **17**(4), 976–985 (2016)
- Li, P.: EE368 project: phase processing with a priori. <http://github.com/leeneil/adm> (2016)
- Liu, H., Spence, J.: XFEL data analysis for structural biology. *Quant. Biol.* **4**(3), 159–176 (2016)
- Luke, R.: Relaxed averaged alternating reflections for diffraction imaging. *Inverse Probl.* **21**(1), 37 (2004)
- Ma, C., Wang, K., Chi, Y., Chen, Y.: Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In: *International Conference on Machine Learning*, pp. 3345–3354. PMLR (2018)
- Marchesini, S.: Invited article: a unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* **78**(1), 011301 (2007)
- Marchesini, S., He, H., Chapman, H., Hau-Riege, S., Noy, A., Howells, M., Weierstall, U., Spence, J.: X-ray image reconstruction from a diffraction pattern alone. *Phys. Rev. B* **68**(14), 140101 (2003)
- Needell, D., Ward, R., Srebro, N.: Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Adv. Neural Inf. Process. Syst.* **27**, 1017–1025 (2014)
- Qiu, T., Palomar, D.: Undersampled sparse phase retrieval via majorization–minimization. *IEEE Trans. Sig. Process.* **65**(22), 5957–5969 (2017)

- Ramos, T., Grønager, B., Andersen, M., Andreassen, J.: Direct three-dimensional tomographic reconstruction and phase retrieval of far-field coherent diffraction patterns. *Phys. Rev. A* **99**(2), 023801 (2019)
- Reddy, H., Yoon, C., Aquila, A., Awel, S., Ayer, K., Barty, A., Berntsen, P., Bielecki, J., Bobkov, S., Bucher, M.: Coherent soft X-ray diffraction imaging of coliphage PR772 at the Linac coherent light source. *Sci. Data* **4**(1), 1–9 (2017)
- Scheres, S.: RELION: implementation of a bayesian approach to cryo-em structure determination. *J. Struct. Biol.* **180**(3), 519–530 (2012)
- Seifert, B., Stolz, H., Donatelli, M., Langemann, D., Tasche, M.: Multilevel Gauss–Newton methods for phase retrieval problems. *J. Phys. A: Math. General* **39**(16), 4191 (2006)
- Shi, Y., Yin, K., Tai, X., DeMirci, H., Hosseinizadeh, A., Hogue, B., Li, H., Ourmazd, A., Schwander, P., Vartanyants, I.: Evaluation of the performance of classification algorithms for XFEL single-particle imaging data. *IUCrJ* **6**(2), 331–340 (2019)
- Sixou, B., Davidou, V., Langer, M., Peyrin, F.: Absorption and phase retrieval with Tikhonov and joint sparsity regularizations. *Inverse Probl. Imag.* **7**(1), 267 (2013)
- Sorzano, C., Marabini, R., Velázquez-Muriel, J., Bilbao-Castro, J., Scheres, S., Carazo, J., Pascual-Montano, A.: XMIPP: a new generation of an open-source image processing package for electron microscopy. *J. Struct. Biol.* **148**(2), 194–204 (2004)
- Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.* **15**(2), 262–278 (2009)
- Tan, Y., Vershynin, R.: Phase retrieval via randomized Kaczmarz: theoretical guarantees. *Inf. Infer. J. IMA* **8**(1), 97–123 (2019)
- Tropp, J.: Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Sig. Process.* **86**(3), 589–602 (2006)
- Wang, G., Giannakis, G., Eldar, Y.: Solving systems of random quadratic equations via truncated amplitude flow. *IEEE Trans. Inf. Theory* **64**(2), 773–794 (2017)
- Wang, H., Wang, J.: How cryo-electron microscopy and X-ray crystallography complement each other. *Protein Sci.* **26**(1), 32–39 (2017)
- Wei, K.: Solving systems of phaseless equations via kaczmarz methods: A proof of concept study. *Inverse Probl.* **31**(12), 125008 (2015)
- Wu, C., Tai, X.: Augmented lagrangian method, dual methods, and split bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imag. Sci.* **3**(3), 300–339 (2010)
- Xian, Y., Gu, H., Wang, W., Huang, X., Yao, Y., Wang, Y., Cai, J.: Data-driven tight frame for Cryo-EM image denoising and conformational classification. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 544–548. IEEE (2018)
- Zhang, H., Liang, Y.: Reshaped wirtinger flow for solving quadratic system of equations. *Adv. Neural Inf. Process. Syst.* **29**, 2622–2630 (2016)
- Zhang, X., Lu, Y., Chan, T.: A novel sparsity reconstruction method from poisson data for 3d bioluminescence tomography. *J. Sci. Comput.* **50**(3), 519–535 (2012)



A Survey on Deep Learning-Based Diffeomorphic Mapping

37

Huilin Yang, Junyan Lyu, Roger Tam, and Xiaoying Tang

Contents

Introduction	1291
Background and Motivation	1291
Diffeomorphic Mapping	1291
Problem Statement and Framework Overview	1293
Deep Learning-Based Methods	1295
Related Deep Network Introduction	1296
Convolutional Neural Networks	1298
Fully Convolutional Network	1300
U-Net	1300
Autoencoders	1302

Huilin Yang and Junyan Lyu contributed equally with all other contributors.

H. Yang

Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China
School of Biomedical Engineering, The University of British Columbia, Vancouver, BC, Canada
e-mail: huiliny1@student.ubc.ca

J. Lyu

Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China
Queensland Brain Institute, The University of Queensland, St Lucia, QLD, Australia
e-mail: 12063003@mail.sustech.edu.cn

R. Tam

School of Biomedical Engineering, The University of British Columbia, Vancouver, BC, Canada
e-mail: roger.tam@ubc.ca

X. Tang (✉)

Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, China
e-mail: tangxy@sustech.edu.cn

Recurrent Neural Networks and Long Short-Term Memory Networks	1302
Unsupervised Methods	1303
Loss Function	1303
CNN-Based Methods	1305
VAE-Based Methods	1307
More Related Works	1309
Supervised Methods	1310
Loss Function	1310
CNN-Based Methods	1311
More Related Works	1314
Discussion and Future Direction	1314
Achievements and Applications	1314
Challenges	1315
Future Directions	1316
Conclusions	1316
References	1317

Abstract

Diffeomorphic mapping is a specific type of registration methods that can be used to align biomedical structures for subsequent analyses. Diffeomorphism not only provides a smooth transformation that is desirable between a pair of biomedical template and target structures but also offers a set of statistical metrics that can be used to quantify characteristics of the pair of structures of interest. However, traditional one-to-one numerical optimization is time-consuming, especially for 3D images of large volumes and 3D meshes of numerous vertices. To address this computationally expensive problem while still holding desirable properties, deep learning-based diffeomorphic mapping has been extensively explored, which learns a mapping function to perform registration in an end-to-end fashion with high computational efficiency on GPU. Learning-based approaches can be categorized into two types, namely, unsupervised and supervised. In this chapter, recent progresses on these two major categories will be covered. We will review the general frameworks of diffeomorphic mapping as well as the loss functions, regularizations, and network architectures of deep learning-based diffeomorphic mapping. Specifically, unsupervised ones can be further subdivided into convolutional neural network (CNN)-based methods and variational autoencoder-based methods, according to the network architectures, the corresponding loss functions, as well as the optimization strategies, while supervised ones mostly employ CNN. After summarizing recent achievements and challenges, we will also provide an outlook of future directions to fully exploit deep learning-based diffeomorphic mapping and its potential roles in biomedical applications such as segmentation, detection, and diagnosis.

Keywords

Diffeomorphic mapping · Deep learning · Unsupervised · Supervised

Introduction

Background and Motivation

In biomedical analysis, it is usually necessary and important to put biomedical manifolds of interest from different individuals into a common coordinate system for further analyses. Registration plays the role of putting two objects of interest into a common coordinate system, and it is usually a necessary pre-processing step before performing statistical analyses of anatomy. Diffeomorphic mapping provides one-to-one as well as smooth correspondences across different objects of interest, which serves as a powerful registration tool and has been successfully applied to a variety of biomedical applications (Louis et al. 2018; Tian et al. 2020; Debavelaere et al. 2020; Tang et al. 2019; Jiang et al. 2018; Yang et al. 2017a). However, most existing applications are based on a traditional numerical optimization scheme, which is time-consuming and could cost up to several hours for registering a single pair of 3D images or 2D meshes. In addition, registering only one template-and-target pair in a single optimization course could not learn any information from all available objects. In such context, utilizing deep learning to tackle this registration task has been extensively explored. Once the mapping function is obtained at the training phase, the network can perform registration within a few seconds given a pair of template and target, wherein only a forward pass is needed. This chapter aims at a comprehensive survey of recent progresses on deep learning-based diffeomorphic mapping methods addressing the two aforementioned issues: first, the low computational efficiency of traditional schemes that only optimize one template-and-target pair during a single optimization course and second, traditional schemes could not learn and utilize information of other available objects in the optimization process.

Diffeomorphic Mapping

Several specific properties are intuitively desirable for transformations across anatomical manifolds of interest from different individuals. First, the transformation is desired to be one-to-one, namely, an element in the template anatomy is supposed to have unique correspondence in the target anatomy. This property ensures the existence and uniqueness of the correspondence between the two anatomical manifolds of interest. In addition, it is critical that the deformed manifold obtained from the transforming process is close to the target anatomy, to ensure the accuracy of the transformation. Furthermore, since the folding of the deformation field over itself can destroy neighborhood structure which is essential for the study of anatomy, the transformation should be able to preserve the topology of template manifold before and after deformation. In this way, originally connected sets are still connected, and originally disconnected ones stay disconnected. As such, diffeomorphic mapping is of considerable interest in this regard. As shown in Fig. 1, in a diffeomorphic setting,

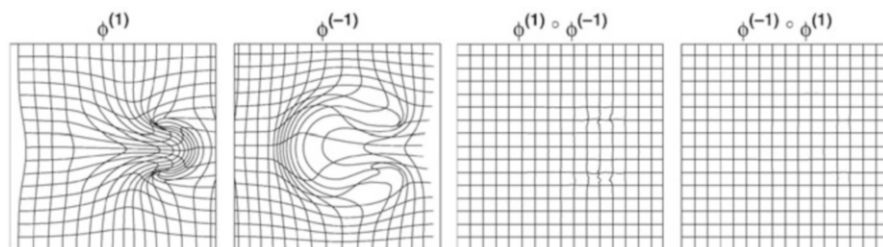


Fig. 1 Demonstration of inversion and composition in a diffeomorphic setting. Left-most: a forward deformation. Second: the corresponding inverse deformation. Both forward and inverse transformations are one-to-one. The last two: compositions of the forward and inverse transformations. (Taken from Ashburner 2007)

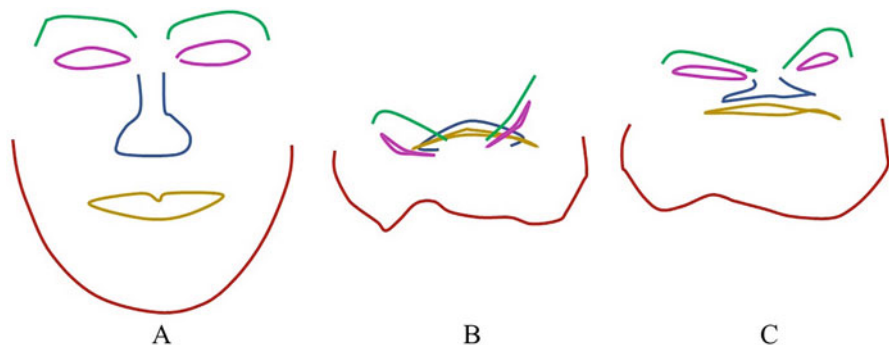


Fig. 2 Illustration of the topology-preserving property of diffeomorphic mapping. (a) is the original face, (b) is the deformed face obtained through a non-diffeomorphic mapping, and (c) is the deformed face obtained through a diffeomorphic mapping

both forward transformation and the corresponding inverse transformation are smooth and unfolding, and compositions of the forward and inverse transformations are very close to identity. Figure 2 illustrates the topology-preserving property of diffeomorphic mapping. The relative locations of different facial organs are well preserved after a diffeomorphic deformation, whereas those obtained from a non-diffeomorphic one are completely destroyed and unrecognizable.

Considering the aforementioned advantages, diffeomorphic mapping works well for mapping and analyzing anatomical information via various medical imaging media. Below, we will briefly introduce two widely used conventional diffeomorphic mapping methods: large deformation diffeomorphic metric mapping (LDDMM) (Beg et al. 2005; Vaillant et al. 2007; Glaunes et al. 2008) and stationary velocity field (SVF) (Arsigny et al. 2006; Modat et al. 2012). Most of the deep learning-based diffeomorphic mapping methods that we will cover in this chapter are based on the frameworks of the two methods.

Large Deformation Diffeomorphic Metric Mapping

LDDMM is a classical suite of algorithms within the academic discipline of computational anatomy, which provides not only a diffeomorphic mapping but also a geodesic metric induced on the group of diffeomorphisms. Under the LDDMM setting, manifolds to be registered could be volumes (Beg et al. 2005), curves (Glaunes et al. 2008), currents and surfaces (Vaillant and Glaunes 2005), landmarks (Joshi and Miller 2000), varifolds (Charon and Trouvé 2013), and tensors (Cao et al. 2006). The template manifold is mapped onto the target one by defining and solving a variational problem through a conditional ordinary differential equation (ODE) (Beg et al. 2005), wherein the diffeomorphism is obtained by minimizing a squared-error matching function between the deformed template and the target. To ensure diffeomorphism, the transforming flow should satisfy the Lagrangian and Eulerian specifications associated with the ODE. The diffeomorphism group is equipped with time-varying speed flows, with vector fields absolutely integrable in the Sobolev norm.

Stationary Vector Field

SVF-based diffeomorphic mapping (Arsigny et al. 2006) generalizes the principal logarithm to nonlinear geometric deformations. It is similar to the Log-Euclidean framework for tensors (Arsigny et al. 2005) in an infinite-dimensional way and aims at computing various statistics of general diffeomorphisms. SVF is defined by a stationary ODE in which the exponential of a vector field is the flow at time 1, which is solved based on nonlinear generalization of the “scaling and squaring” method. Different from LDDMM with time-varying speed flows, SVF provides flows of vector fields with stationary speed and a way to compute typical Euclidean statistics on diffeomorphisms via logarithms.

Problem Statement and Framework Overview

Given a moving object m and a fixed object f , it is preferable to find a diffeomorphic one-to-one correspondence between them so as to put them into a same reference system for further analyses. A general framework of diffeomorphic mapping for images or shapes can be, respectively, framed with the following objective functions:

$$J_{f,m}(v_t) = \min_{v_t: \dot{\phi}_t = v_t(\phi_t), \phi_0 = id} \gamma R(\phi_1) + D(\phi_1^{-1} \circ m, f), \quad (1)$$

$$J_{f,m}(v_t) = \min_{v_t: \dot{\phi}_t = v_t(\phi_t), \phi_0 = id} \gamma R(\phi_1) + D(\phi_1 \cdot m, f), \quad (2)$$

where $R(\phi_1)$ is a regularization term that ensures the mapping’s smoothness and diffeomorphism property. The second term quantifies the overall discrepancy between the deformed moving object $\phi_1^{-1} \circ m$ (images) or $\phi_1 \cdot m$ (shapes) and the target/fixed object f . For simplicity, we denote as $\phi_1 \cdot m$ for both images

and shapes in all subsequent contexts. After transformation, it is supposed that the deformed moving object $\phi_1 \cdot m$ should be very close to the fixed object f . v_t is the velocity of the transformation with respect to time t . We name it as a dynamic velocity field method when the velocity of the mapping varies across time and as a stationary velocity field method when the velocity of the mapping stays static during transformation. When $t = 0$, the registration field is identity such that $\phi_0 \cdot m = m$, and the optimal registration field ϕ_1 is obtained at $t = 1$. γ is a weight ranging from 0 to 1, serving as a trade-off coefficient between the regularization term and the overall discrepancy term. Increasing γ imposes more weight on the registration field enforcing a smoother transformation, whereas decreasing γ puts more attention on the discrepancy term making the deformed moving object closer to the fixed object.

It should be noted that, in traditional methods, we get only one optimal registration field at the time $t = 1$ after optimizing the objective function with respect to a pair of moving object and fixed object. At the top panel of Fig. 3, we present the flowchart of the typical optimization scheme in traditional methods. There are a variety of methods that can be categorized into this category, including large LDDMM (Beg et al. 2005; Vaillant et al. 2007; Glaunes et al. 2008) and SVF (Modat et al. 2012). During the past decade, they have been extensively and successfully applied to various biomedical applications (Tang et al. 2019; Jiang et al. 2018; Yang et al. 2015, 2017a; Bossa et al. 2010). Nevertheless, since these methods all make use of traditional optimization schemes and biomedical data usually have large size especially for 3D data such as MRI and CT, it usually takes such methods up to several hours to process one pair of objects of interest. In order

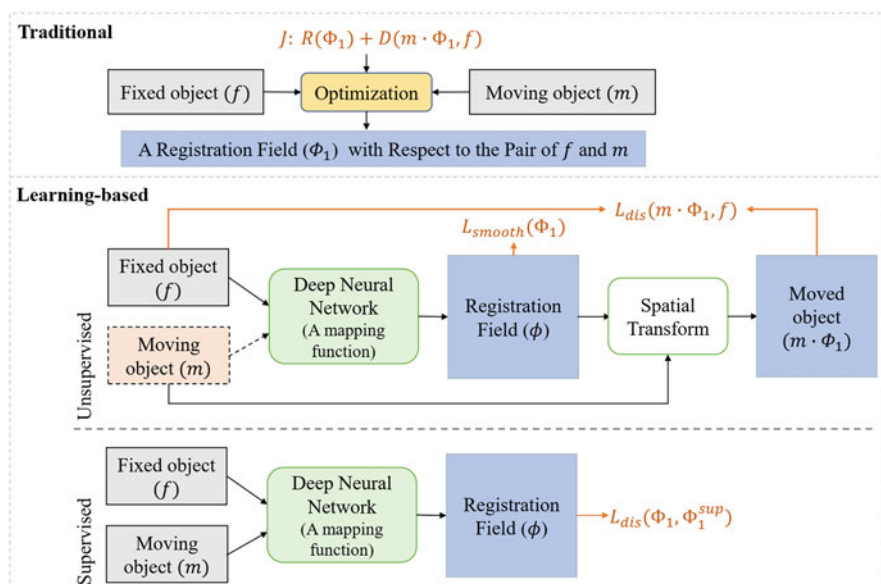


Fig. 3 An overview of traditional and deep learning-based registration methods. The top panel shows the flowchart of traditional methods, and the bottom panel shows the flowcharts of two types of deep learning-based methods (unsupervised ones and supervised ones)

to address this problem, recent researches have started to focus on learning the registration field through deep learning. Endowed by powerful functional properties and computational efficiency on GPU of deep neural networks, the registration time has been largely reduced and is capable of predicting not only one but multiple registration fields. Once training has been finished, a pair of even large-size objects can be processed within several seconds. According to the learning style, deep learning-based diffeomorphic mapping can be categorized into two major classes, namely, unsupervised methods and supervised methods.

Deep Learning-Based Methods

Deep learning-based methods can be divided into unsupervised ones and supervised ones according to whether they require labels from traditional methods. A brief summary of key information for deep learning-based methods is illustrated in Fig. 4. Details will be described in the following subsections.

Unsupervised Methods

Unsupervised methods refer to such a kind of approach that trains a deep neural network without any information of registration fields obtained through traditional methods. It usually directly minimizes the discrepancy between the deformed moving object and the fixed object together with regularization on the registration field. In the upper part of the bottom panel in Fig. 3, we show the flowchart of unsupervised methods, which takes as inputs a fixed object and a moving object and

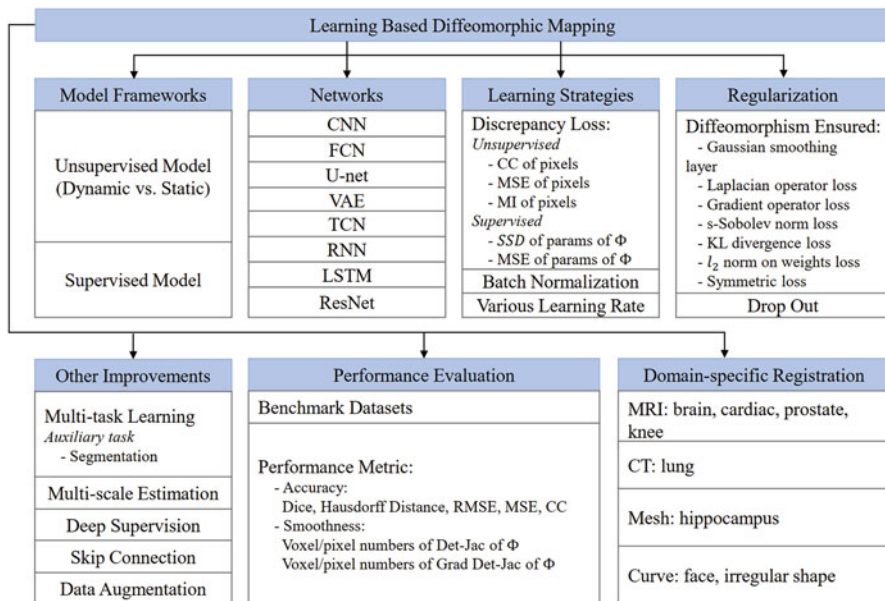


Fig. 4 A summary of deep learning-based diffeomorphic mapping

then feeds them into a deep neural network to predict the corresponding registration field. The subsequent spatial transform module takes the predicted registration field and the moving object as inputs to perform diffeomorphic deformation yielding the deformed moving object (also called moved object). Some methods only need to input fixed objects, and the moving object (in dashed box in Fig. 3) can be estimated during training. The whole training phase makes use of only two kinds of information, namely, the fixed objects and some methods the corresponding moving objects. Regularization is imposed on the registration field through the loss term $L_{\text{smooth}}(\phi_1)$, and measurement of similarity is conducted through minimizing the discrepancy loss $L_{\text{dis}}(m \cdot \phi_1, f)$. Unsupervised methods mimic the traditional optimization scheme except that they aim at predicting diffeomorphisms between a set of template-and-target pairs instead of between only one pair of template and target within one single training course.

Supervised Methods

Compared with unsupervised methods, supervised methods usually take three kinds of information as inputs including the fixed objects, the moving objects, and parameterizations of the corresponding registration fields such as the velocity or momentum acquired from performing traditional methods. This means that we first need to conduct traditional diffeomorphic registrations on all pairs of moving-and-fixed objects to obtain the corresponding registration fields, called ϕ_1^{sup} , as ground truth. We then use them together with all moving-and-fixed pairs of objects as training materials. The lower part of the bottom panel in Fig. 3 shows the training flow of supervised methods. The loss function $L_{\text{dis}}(\phi_1, \phi_1^{\text{sup}})$ minimizes the discrepancy between the predicted registration field ϕ_1 and the pre-obtained “ground truth” registration field ϕ_1^{sup} . Supervised methods assume the registration fields obtained through the traditional optimization scheme are optimal and try to make the learning-based predictions as close to them as possible.

The remainder of this chapter is organized as follows. Related deep network introduction and summary will be described in section “[Related Deep Network Introduction](#)”. Unsupervised methods will be reviewed in section “[Unsupervised Methods](#)”, followed by a survey of supervised methods in section “[Supervised Methods](#)”. We will also cover current achievements and related applications in section “[Discussion and Future Direction](#)”. After reviewing existing works, potential emerging topics and future directions will be elaborated. Finally, conclusion of this survey will be organized in section “[Conclusions](#)”.

Related Deep Network Introduction

Leveraged by deep learning and neural networks, diffeomorphic mapping can be achieved in an efficient manner. Related neural network types that have been employed in learning-based diffeomorphic mapping approaches surveyed in this chapter are summarized in Fig. 4, and the specific approaches together with their corresponding adopted networks are, respectively, listed in Table 1 for unsupervised

Table 1 Summary of all reviewed unsupervised learning-based diffeomorphic mapping methods. (*ROI* region of interests, *CNN* convolutional neural network *FCN* fully convolutional network, *CVAE* conditional variational autoencoder, *VAE* variational autoencoder, *RNN* recurrent neural network, *TCN* temporal convolutional network, *ResNet* residual neural network, *SVF* static velocity field, *RDMM* region-specific diffeomorphic metric mapping, *LDDMM* large deformation diffeomorphic metric mapping, *NLCC* normalized local cross-correlation, *MSE* mean squared error, *SSD* sum of squared difference, *CD* the Chamfer distance, *EMD* the Earth mover’s distance, d the displacement of the predicted registration field, v the velocity of the predicted registration field, *LOC* local orientation consistency, *KL divergence* Kullback-Leibler divergence, *OMT* optimal mass transport, *MRI* magnetic resonance imaging, *CT* computed tomography)

Reference	Network	Velocity	Similarity metric	Regularity term	Modality	ROI	Others
Balakrishnan et al. (2019)	U-Net	Static	NLCC; MSE	∇^2 on d	MRI	Brain	Auxiliary segmentation; instance-specific fine-tuning
Balakrishnan et al. (2018)	U-Net	Static	NLCC	∇^2 on d	MRI	Brain	–
Dalca et al. (2018)	U-Net	Static	MSE	∇^2 on inverse of covMatrix of v	MRI	Brain	Uncertainty analysis
Dalca et al. (2019a)	U-Net	Static	MSE	∇^2 on inverse of covMatrix of v	MRI	Brain	Surface information
Mok and Chung (2020a)	FCN	Static	NLCC	∇ on v ; LOC (Mok and Chung 2020a)	MRI	Brain	Symmetric map & loss
Krebs et al. (2019)	CVAE	SVF	Symmetric NLCC	Gaussian smoothing layer on v	MRI	Cardiac	Symmetric loss; from healthy to pathological cases
Bône et al. (2019)	VAE	Dynamic	Norm on vector valued mesh metric	KL divergence	MRI; mesh	Brain; hippocampus	Current-splating layer for meshes
Bône et al. (2019)	VAE	SVF	Likelihood probability	3-Sobolev norm (Zhang and Fletcher 2019) on v	MRI; mesh	Face; hippocampus	Current-splating layer for meshes; private dataset

(continued)

Table 1 (continued)

Han et al. (2020)	CNN	SVF	NLCC	Differential operator on v	MRI	Brain	Brain with tumors
Shen et al. (2019a)	U-Net	SVF	NLCC	Differential operator on v	MRI	Knee	Symmetric loss
Louis et al. (2019)	RNN	Dynamic	SSD	Gaussian smoothing layer on v	MRI	Brain	Force small variance on latent space
Niethammer et al. (2019)	CNN	SVF	NLCC	OMT on multi-Gaussian kernel weights	MRI	Brain	OMT on local deformation
Krebs et al. (2021)	CVAE TCN	SVF	SSD	Gaussian smoothing layer on v	MRI	Cardiac	Regularity on both spatial and temporal domains
Mok and Chung (2020b)	CNN	SVF	NLCC	∇^2 on v	MRI	Brain	Coarse-to-fine; Laplacian pyramid framework
Shen et al. (2019b)	CNN	RD-MM	Multi-kernel NLCC	∇^2 on v ; OMT regularity	CT	Lung; knee	Multi-kernel NLCC similarity metric
Hoffmann et al. (2020)	U-Net	SVF	Soft Dice	∇ on d	MRI	Brain	Train purely with synthetic data
Amor et al. (2021)	Res-Net	LDD-MM	CD; EMD	LDDMM regularity on v	Mesh	Cortex; heart; liver; femur; hand	Train on one pair of data

methods and Table 2 for supervised methods. In this section, we will introduce in detail several main types of deep neural network (DNN) architectures that have been adopted in existing diffeomorphic mapping approaches.

Convolutional Neural Networks

Convolutional neural networks (CNNs) have made impressive progress in computer vision tasks including image recognition (Krizhevsky et al. 2012), object detection (Liu et al. 2020), and semantic segmentation (Lateef and Ruichek 2019). As shown in Fig. 5, a CNN typically consists of convolutional layers, pooling layers, activation layers, and fully connected layers. A convolutional layer contains a set of learnable

Table 2 Summary of all reviewed supervised learning-based diffeomorphic mapping methods. (*ROI* region of interests, *CNN* convolutional neural network, *FCN* fully convolutional network, *LSTM* long short-term memory module, *LDDMM* large deformation diffeomorphic metric mapping, *SSD* sum of squared difference, *MSE* mean squared error, v the velocity of the predicted registration field, *MRI* magnetic resonance imaging)

Reference	Network	Velocity	Similarity metric	Regularity term	Modality	ROI	Others
Yang et al. (2017c)	U-shape CNN	Static	SSD	LDDMM regularity on v	MRI	Brain	A correction network for momenta
Rohé et al. (2017)	U-Shape FCN	Static	SSD	–	MRI	Cardiac	Data augmentation; no regularity
Wang and Zhang (2020b)	CNN	LDDMM	SSD	Fourier domains of v ; l_2 norm on network weights	MRI	Brain	Process in Fourier domain; two networks for real and imaginary parts separately
Ding et al. (2019)	CNN	LDDMM	SSD	∇^2 on v	MRI	Brain	Longitudinal registration study
Kwitt and Niethammer (2017)	CNN	LDDMM	SSD	∇^2 on v	MRI	Brain	–
Wang and Zhang (2020a)	CNN	LDDMM	MSE	l_2 norm on network weights	MRI	Brain	–
Pathan and Hong (2018)	LSTM; CNN	LDDMM	MSE	Momentum sequence	MRI	Brain	–
Krebs et al. (2017)	CNN	Dynamic	SSD	Fuzzy action control	MRI	Prostate	Reinforcement learning

kernels operating on local input regions to extract features. A pooling layer performs linear or nonlinear downsampling on feature maps to reduce spatial resolution and summarize local information. An activation layer can be the sigmoid function, the hyperbolic tangent function (Tanh), or the rectified linear unit (ReLU) (Sibi et al. 2013), introducing nonlinearity to CNNs. A fully connected layer is the same as multi-layer perceptron, providing final classification or regression predictions. By stacking these layers hierarchically, CNNs gain large receptive fields and thus can exploit and capture scale-invariant and translation-invariant features. Several novel CNN architectures including VGGNet (Simonyan and Zisserman 2014), Inception

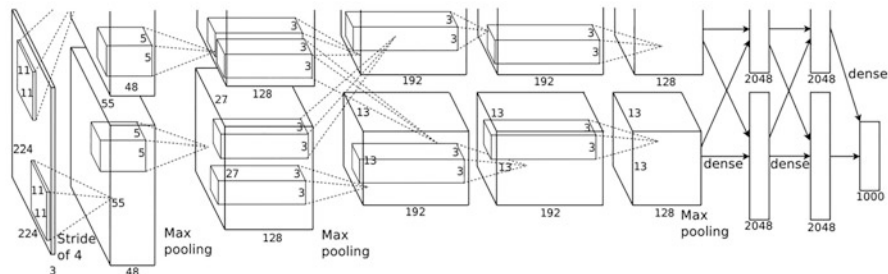


Fig. 5 A typical hierarchy of convolutional neural networks. (Taken from Krizhevsky et al. 2012)

(Szegedy et al. 2017), ResNet (He et al. 2016), and DenseNet (Iandola et al. 2014) have been proposed since 2012, after AlexNet achieved a big breakthrough on ImageNet classification; the error rate was reduced by half (Krizhevsky et al. 2012).

Fully Convolutional Network

Fully convolutional network (FCN) is proposed in 2015 and is originally designed for semantic segmentation. FCN first inputs images to an arbitrary backbone network to produce feature maps, the backbone network of which can be AlexNet or VGGNet. It then applies deconvolutional layers (Zeiler et al. 2010), which simply reverse the forward and backward pass of convolution to upsample feature maps. The upsampled dense-pixel feature maps are subsequently sent to a 1×1 convolution layer with desired channel dimensions to get pixel-level spatial predictions. As a result, FCN is able to take an input of arbitrary size and produce an output of the same size in an end-to-end manner, whereas previous CNNs cannot.

In order to refine spatial details, multi-scale feature maps from previous convolutional layers in the backbone network are deconvoluted and yield additional predictions. By aggregating those predictions, FCN can combine semantic (high-level) information from coarse-scale predictions and appearance (low-level) information from fine-scale predictions and thus further boost the final precision. As a result, FCN demonstrates state-of-the-art performance on PASCAL VOC, NYUDv2, and SIFT Flow. FCN and its variants have also been applied to medical image segmentation tasks such as lung segmentation (Kaul et al. 2019), whole brain segmentation (Roy et al. 2018), and retinal vessel segmentation (Lyu et al. 2019).

U-Net

Inspired by FCN, Ronneberger et al. (2015) propose a novel encoder-decoder network, named U-Net, for biomedical image segmentation. U-Net has a symmetrical

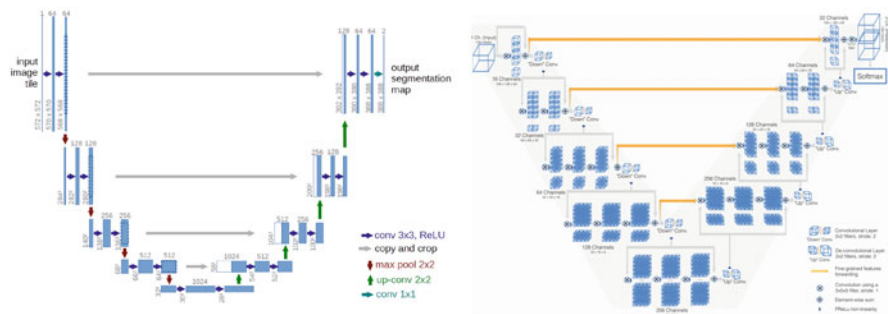


Fig. 6 Illustrations of U-Net (left) and V-Net (right). (Taken from Ronneberger et al. 2015 and Milletari et al. 2016)

U-shaped architecture with a contracting path and an expanding path, as illustrated in Fig. 6. The contracting path performs repetitions of two 3×3 convolutions followed by ReLU and 2×2 max pooling to encode contextual information, while the expanding path performs repetitions of two 3×3 convolutions with ReLU and 2×2 deconvolution to gradually decode feature maps and recover spatial resolution. Finally, a 1×1 convolution projects the feature maps to output space.

The feature maps from the contracting path are, respectively, skip connected to the corresponding feature maps from the expanding path with the same spatial resolutions. Compared with FCN that aggregates multi-level features in the output space, U-Net fuses these features via concatenation in the feature space. This enables U-Net to propagate more information from previous layers to subsequent layers, contributing to better gradient flows and faster convergence speed in the training course. Moreover, its light and compact architecture allows U-Net to better converge on biomedical image datasets which typically have fewer labeled training samples and higher spatial resolutions than natural images. U-Net is evaluated on the EM segmentation challenge (Arganda-Carreras et al. 2015) and the ISBI cell tracking challenge (Ulman et al. 2017), outperforming previous methods by large margins.

The success of U-Net promotes the development of its extensions. One of its most well-known variants is V-Net proposed by Milletari et al. in 2016. V-Net extends U-Net to work on volumetric medical image segmentation. There are several modifications other than simply changing 2D convolutions to 3D convolutions. V-Net learns additional residual functions at each stage: the input of each stage is element-wisely added to the corresponding output. This operation effectively simplifies the network and alleviates the vanishing gradient issue and thus solves the convergence problem on volumetric CNNs. In addition, $5 \times 5 \times 5$ convolutions are adopted to replace the original $3 \times 3 \times 3$ convolutions to gain larger receptive fields. Max pooling is replaced with strided convolution to perform parametric downsampling, introducing more nonlinearities. V-Net has been used to perform prostate segmentation on MRI volumes in a fast and accurate manner (Milletari et al. 2016).

Autoencoders

Autoencoders are first proposed for dimensionality reduction (Wang et al. 2014). An autoencoder learns an approximation of the identity mapping through an encoder-decoder structure: an encoder extracts the latent space representation of the input, and a decoder reconstructs the input with the extracted vector. The encoder and decoder can be multi-layer perceptrons, CNNs, or any feed-forward neural networks. Supervised by the identity loss (mean absolute error or mean square error) between the input and the output, the latent vector is able to provide a compact expression of the input in a lower-dimensional space. Autoencoders have been applied to principal component analysis (Kramer 1991), image denoising (Gondara 2016), and anomaly detection (Zhou and Paffenroth 2017).

Variational autoencoder (VAE) (Kingma and Welling 2013) is one of the most important variants of autoencoders. Unlike a traditional autoencoder, a VAE assumes that the latent space fits a certain probability distribution, such as a Gaussian distribution, and estimates the parameters of this probability distribution from the input data. Therefore, the approximated distribution of the latent space of a VAE matches the input space closer than that of a traditional autoencoder. In addition to minimizing the identity loss between the input and the output, a regularization term of Kullback-Leibler (KL) divergence between the desired distribution and the predicted distribution is used to train a VAE.

Recurrent Neural Networks and Long Short-Term Memory Networks

CNNs and the other aforementioned networks are unable to handle input sequences of various lengths and thus cannot model the temporal correlations within sequences. Recurrent neural networks (RNNs) (Karpathy et al. 2015) are proposed to solve this problem and have been widely used to process text, video, and time series. At each timestamp, a RNN collects the previous hidden state vector and the current input vector to update the current hidden state and produces output by sending the current hidden state vector to a feed-forward network.

However, RNNs suffer from vanishing or exploding gradients as the sequences grow longer (Karpathy et al. 2015), resulting in poor performance on capturing long-term dependencies. Long short-term memory (LSTM) (Karpathy et al. 2015) is explicitly designed to address such long-term dependency issue. LSTM introduces three gates to protect and control the cell state and the hidden state: the forget gate is used to determine how much information in the previous cell state should be kept; the input gate is used to collect useful information from the current input and the previous hidden state and add them to the filtered previous cell state so as to update the current cell state; and the output gate is used to output a filtered informative vector, namely, the current hidden state, from the updated cell state. All these three types of gates take the previous hidden state as well as the current input as inputs for calculations of their corresponding filter coefficients (Fig. 7).

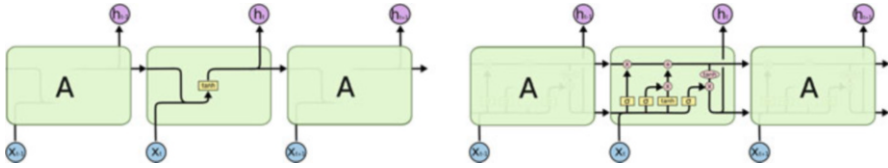


Fig. 7 Architectures of a RNN (left) model and a LSTM (right) model. (Taken from Karpathy et al. 2015)

Unsupervised Methods

In this section, we will survey the literature on methods of unsupervised deep learning-based diffeomorphic mapping. The goal of training an unsupervised network is to ensure it can predict the corresponding registration fields and perform diffeomorphic deformations when pairs of to-be-registered objects are given.

We will start the chapter with loss function and several representative similarity metrics. We then proceed to a variety of regularization approaches for diffeomorphic mapping, and finally several CNN-based (specifically U-Net-based and VAE-based) methods as well as more related works will be introduced.

Loss Function

In a deep learning-based unsupervised diffeomorphic mapping framework, the typical loss function is also composed of two parts, namely, the similarity term and the regularization term. However, the optimization procedure of deep learning-based methods is completely different from that of traditional methods. A typical loss function can be written as follows:

$$L^{\text{unsup}} = L_{\text{sim}}(m \cdot \phi_1, f) + \gamma L_{\text{reg}}(\phi_1), \quad (3)$$

where L_{sim} is the similarity term measuring the difference between the deformed moving objects $m \cdot \phi_1$ and the fixed objects f and L_{reg} is the regularization term imposing certain constraint on the registration fields ϕ_1 to make them diffeomorphic. In the process of minimizing the loss function, the set of deformed moving objects is increasingly closer to the set of fixed objects, and the corresponding registration fields are becoming smoother. γ is a trade-off factor between the similarity term and the regularization term. A too large γ will result in inadequate registration fields that cause highly inaccurate registrations, whereas a too small γ will lead to overly flexible registration fields that might be irregular and not diffeomorphic anymore. In practice, γ is usually empirically chosen.

Similarity Metrics

For different data types, there are different metrics to quantify the similarity between the moved objects $m \cdot \phi_1$ and the fixed objects f . For image data, mean squared error

(MSE), normalized local cross-correlation (NLCC), and mutual information (MI) are often employed. MSE is computed by averaging squared pixel-wise (2D)/voxel-wise (3D) image intensity differences, which can be expressed as

$$MSE(m \cdot \phi_1, f) = \frac{1}{|\Omega|} \sum_{p \in \Omega} [m \cdot \phi_1(p) - f(p)]^2. \quad (4)$$

In MSE, p indexes image pixels or voxels and Ω represents the whole image. Since loss function is minimized to train the deep network, a small MSE is desired to yield a good alignment result. The similarity loss term with MSE can be directly written as $L_{\text{sim}}(m \cdot \phi_1, f) = MSE((m \cdot \phi_1), f)$. Unlike MSE that measures a global difference, NLCC quantifies local cross-correlation and is commonly termed as CC, which is computed over the whole image. CC can be written as

$$\begin{aligned} CC(m \cdot \phi_1, f) &= \sum_{p \in \Omega} NLCC \\ &= \sum_{p \in \Omega} \frac{\left(\sum_{p_i} ([m \cdot \phi_1](p_i) - [m \cdot \phi_1](\bar{p})) (f(p_i) - f(\bar{p})) \right)^2}{\left(\sum_{p_i} (f(p_i) - f(\bar{p}))^2 \right) \left(\sum_{p_i} ([m \cdot \phi_1](p_i) - [m \cdot \phi_1](\bar{p}))^2 \right)} \end{aligned} \quad (5)$$

where $f(\bar{p})$ and $[m \cdot \phi_1](\bar{p})$ denote images with local mean intensities, $f(\bar{p}) = \frac{1}{n^d} \sum_{p_i} f(p_i)$, with p_i iterates over a n^2 area (2D) or a n^3 volume (3D) around p . A higher CC indicates a better alignment, yielding the loss function: $L_{\text{sim}}(m \cdot \phi_1, f) = -CC(m \cdot \phi_1, f)$.

For shape data such as landmarks, curves, or meshes, l_2 norm or norm of differences on manifold-based vector-valued metrics (Vaillant et al. 2007; Glaunes et al. 2008) is usually taken as the similarity term.

Regularization for Diffeomorphic Mapping

In order to make the registration field diffeomorphic, imposing regularization on it is necessary to ensure the smoothness of the deformation. Typically, there are three types of vector fields parameterizing the registration field: displacement field, velocity field, and momentum field. Displacement directly gives the length and direction that the moving object should move. Integral of velocity over time gives the displacement. Momentum usually is a dual space of velocity, and thus velocity can be computed from momentum. These three types of vector fields characterize the registration field in different forms. Regularizations are usually conducted on displacement or velocity in a way of minimizing their differential spaces (such as the first-order or the second-order derivatives) with a vector norm. Let \mathbf{u} denote displacement or velocity; then L_{reg} can be written as

$$L_{\text{reg}}(\phi) = \sum_{p \in \Omega} \|\nabla \mathbf{u}(p)\|^2, \quad (6)$$

or

$$L_{\text{reg}}(\phi) = \sum_{p \in \Omega} \|\nabla^2 \mathbf{u}(p)\|^2, \quad (7)$$

where ∇ is the gradient operator. Other than formulating the regularization term as part of the loss function, employing a smoothing filter like Gaussian convolution right behind the layer for predicting displacement or velocity is also a useful way to smooth the registration field. Applying a Gaussian convolution layer is equivalent to imposing a diffusion-like regularization prior on the predicted velocity or displacement (Krebs et al. 2019).

CNN-Based Methods

Unsupervised learning-based diffeomorphic mapping has been extensively exploited since 2018. The most frequently used deep networks are CNN-based models including FCN and U-Net. We will detail several representative works in this subsection as well as briefly introduce other related works in section “[More Related Works](#)”.

VoxelMorph VoxelMorph (Balakrishnan et al. 2019) takes one moving image and one fixed image as inputs and feeds them into a U-shape CNN to predict displacements \mathbf{u} through a function $g_{\theta}(f, m)$ with network parameters θ . Then, a variant of spatial transform layer (Jaderberg et al. 2015) is applied to deform the moving image using the predicted displacement to register it onto the fixed image. Laplacian regularization is imposed on the predicted displacement. After finishing training VoxelMorph, a mapping that can predict a set of pairwise deformation fields across the population of interest is learned.

VoxelMorph has been extensively validated on 3731 T1-weighted brain MRI scans from eight publicly available datasets. All MRI data go through standard pre-processing steps (also applicable to other methods surveyed in this chapter), including affine spatial normalization and brain extraction using FreeSurfer (Fischl 2012). Considerable reduction on the registration time is achieved by VoxelMorph while holding comparable registration accuracy compared to the two classical methods (around 0.77 Dice for each of the three compared methods and 0.608 Dice for affine alignment). Both atlas-based registration and subject-to-subject registration are evaluated to validate the effectiveness of VoxelMorph. In addition, auxiliary tasks such as segmentation are also investigated in VoxelMorph. Extensive experiments show that incorporating auxiliary tasks in the training procedure is beneficial for the registration accuracy.

VoxelMorph-diff Unlike VoxelMorph which predicts deterministic displacement, VoxelMorph-diff (Dalca et al. 2018) assumes that each registration field between a pair of images fits a normal distribution and learns a generative model that is

able to quantify the registration uncertainty. VoxelMorph-diff employs a similar framework as VoxelMorph does, but it predicts the mean and covariance matrix of the velocity field instead of the displacement. A posterior probability of multivariate normal distribution with a diagonal covariance matrix is imposed on the velocity field for a pair of given moving-and-fixed images, followed by seven squaring and scaling layers (Arsigny et al. 2006) to compute the deformation field ϕ_1 . Then a subsequent spatial transform layer deforms the moving image using the obtained deformation field ϕ_1 . The model is trained by optimizing the variational lower bound of KL divergence, making the predicted posterior probability distribution approximate the true posterior probability distribution. MSE is employed as the similarity metric in the loss function. A newly defined Laplacian operator on the inverse of the predicted covariance matrix of the velocity field with hyperparameter λ is employed for regularization. In the testing phase, given a pair of images, the predicted mean of the velocity field can be used as the optimal sample for the subsequent deformation process. Furthermore, numerous samples could be drawn from the learned distribution to evaluate the registration uncertainty.

For fair comparisons, the same datasets and settings as Balakrishnan et al. (2018) are used. VoxelMorph-diff outperforms VoxelMorph and ANTS SyN in terms of all metrics: 0.753, 0.75, and 0.75 on Dice; 0.7, 18096, and 6505 on the averaged number of voxels whose Jacobian determinants are less than or equal to 0; and 0.451s, 0.554s, and – on GPU (NVIDIA TitanX). Particularly, only VoxelMorph-diff can quantify the registration uncertainty. Experimental results indicate higher uncertainty appears at anatomical boundaries, while lower uncertainty appears at regions that are relatively far from anatomical boundaries.

SYMNet In order to ensure the preferable diffeomorphic mapping properties, SYMNet (Mok and Chung 2020a) is proposed. SYMNet presents a symmetric image registration method that maximizes the similarity between images with respect to the space of diffeomorphic mappings. In addition, it simultaneously estimates both forward and backward transformations with an additional local orientation consistency regularization term (Mok and Chung 2020a) that forces local deformations to be consistent and smooth. Specifically, SYMNet takes images X and Y as inputs; they are fed into a U-shape FCN to predict two symmetric velocity fields v_{XY} and v_{YX} . Meanwhile, it takes the negative of v_{XY} and v_{YX} , respectively, obtaining $-v_{XY}$ and $-v_{YX}$. After performing scaling and squaring (Arsigny et al. 2006) on each of these four velocity fields, four deformation fields $\phi_{XY}^{(0.5)}$, $\phi_{XY}^{(-0.5)}$, $\phi_{YX}^{(0.5)}$, and $\phi_{YX}^{(-0.5)}$ are yielded. Then, $\phi_{XY}^{(0.5)}$ and $\phi_{YX}^{(-0.5)}$ are composed and applied to X via a diffeomorphic spatial transformer derived from Jaderberg et al. (2015) to obtain $\phi_{XY}^{(1)}(X)$. $\phi_{YX}^{(0.5)}$ and $\phi_{XY}^{(-0.5)}$ are composed and applied to Y to obtain $\phi_{YX}^{(1)}(Y)$. $\phi_{XY}^{(0.5)}$ is applied to X yielding $\phi_{XY}^{(0.5)}(X)$, and $\phi_{YX}^{(0.5)}$ is applied to Y yielding $\phi_{YX}^{(0.5)}(Y)$. Thus, the similarity term consists of two parts: $L_{\text{sim}} = L_{\text{mean}} + L_{\text{pair}}$, in which $L_{\text{mean}} = -CC(\phi_{XY}^{(0.5)}(X), \phi_{YX}^{(0.5)}(Y))$ and $L_{\text{pair}} = -CC(\phi_{XY}^{(1)}(X), Y) - CC(\phi_{YX}^{(1)}(Y), X)$. For regularization purposes, SYMNet employs three terms: L_{Jdet}

measures the averaged number of voxels whose Jacobian determinants of the deformation field are less than 0, serving as a local orientation consistency regularity. L_{reg} measures the l_2 norm on the gradients of v_{XY} and v_{YX} across all voxels serving as a global smoothness regularity. L_{mag} measures the averaged discrepancy between the l_2 norm of v_{XY} and that of v_{YX} , explicitly guaranteeing the magnitudes of the two predicted symmetric velocity fields to be (approximately) the same. Both L_{mean} and L_{mag} enforce the mapping and the corresponding inverse mapping to be symmetric.

Comparing SYMNet with ANTs SyN (Avants et al. 2008), VoxelMorph (Balakrishnan et al. 2019), and VoxelMorph-diff (Dalca et al. 2018) are conducted via atlas-based registration using 425 T1-weighted brain MRI scans from OASIS (Foteno et al. 2005). Different from the original experimental settings in VoxelMorph and VoxelMorph-diff, all learning-based methods involved in the comparisons are trained by pairwise registrations of all image pairs in the training set. The average Dice scores for SYMNet, VoxelMorph, VoxelMorph-diff, and ANTs SyN are, respectively, 0.738, 0.707, 0.693, and 0.680 (0.567 for affine only), and the corresponding numbers of voxels whose Jacobian determinants are less than or equal to 0 are, respectively, 0.471, 0.588, 346.712, and 0.047. The running time is 0.414s for SYMNet, 0.695 s for VoxelMorph, and 0.517 s for VoxelMorph-diff on a NVIDIA GTX 1080Ti GPU and 1039 s for ANTs SyN on an Intel Core i7-7700 CPU. SYMNet achieves the best performance on the evaluated dataset. Ablation studies successfully validate the effectiveness of the local orientation-consistency loss proposed by SYMNet.

VAE-Based Methods

Different from CNN-based methods which usually directly estimate the parameterizations (displacement, velocity, or momentum) of the registration field, VAE-based methods estimate a latent space that encodes the deformation space through an encoder and predict the velocity field through a decoder. Subsequent layers deform the moving image to reconstruct the input image (the fixed image) with the predicted velocity field, yielding the deformed moving image. Furthermore, the template, namely, the moving image, can be simultaneously estimated together with the latent space in the training phase. Two representative works will be described in detail, and more related works will be briefly covered in section “[More Related Works](#)”.

ProbDR (Krebs et al. 2019) models registration in a probabilistic and generative framework by applying a conditional variational autoencoder (CVAE) with multi-scale deformations, denoted as ProbDR. ProbDR assumes that the transformations for a to-be-registered population could be represented using a compact low-dimensional latent space (follows a multivariate unit Gaussian distribution with spherical covariance) and assumes the velocity of the deformation could be decoded from this latent space. Given a pair of moving-and-fixed images, the corresponding low-dimensional representation in the latent space can be estimated and fed into the decoder for calculating the velocity field of the deformation, which is then smoothed

by a Gaussian convolution layer to ensure the diffeomorphism property. After that, the moving image is fed into a dense warping layer implemented via STN together with the calculated velocity field to acquire the finally deformed moving image. ProbDR concurrently conducts registration in a multi-scale fashion to further boost the performance. The moving image at each scale is fed into the corresponding deconvolution layer of the same resolution in the decoder to learn more geometry-invariant representations in the latent space. A Boltzmann distribution likelihood with symmetric NLCC is employed as the posterior probability distribution of the input images given the moving image and the corresponding predicted latent space. Moreover, the trained decoder network can be used to sample and transport new deformations in the following way: sampling latent representations from the previously predicted mean and covariance and then applying the sampled representations to the new moving images through subsequent networks. To be noticed, although VoxelMorph-diff also learns a generative model, ProbDR learns a much more compact low-dimensional representation of the deformation field instead of predicting the velocity field which is of a much higher dimension.

Extensive evaluations are conducted on 3D intra-subject registration using 334 cardiac cine-MRIs. The training set are randomly shifted, rotated, scaled, and mirrored as data augmentation. It should be noted that the aforementioned methods are all trained without data augmentation. Comparisons are conducted with LCC-demons (Lorenzi et al. 2013), ANTs SyN (Avants et al. 2008), and VoxelMorph (Balakrishnan et al. 2019). ProbDR obtains the best results with respect to Dice, Hausdorff distance, and the averaged number of voxels whose Jacobian determinants are less than or equal to 0, respectively, being 0.812, 7.3 mm, and 1.4. VoxelMorph gets the best RMSE being 0.24, while ProbDR obtains a RMSE of 0.30. A five-disease classification accuracy of 83% is obtained by ProbDR when using the eight most discriminative components from canonical correlation analysis. ProbDR also demonstrates how to perform deformation transport from healthy to disease without inter-subject registration for pre-processing, which is needed by the other three methods.

LRShape Another method that learns low-dimensional representations of diffeomorphic mapping is proposed by Bone and published in 2019 (Bône et al. 2019), denoted as LRShape. Unlike all of the aforementioned methods that mainly focus on image data and prediction of static velocity field, LRShape focuses on shape data such as curves and surfaces and predicts a time-varying velocity field. A current-splatting layer (Durrleman 2010; Gori et al. 2017) that allows neural network architectures to process meshes is presented in this work. In contrast to ProbDR that takes both moving and fixed objects as the inputs, LRShape only takes the fixed shape as the input and estimates the template (namely, the moving shape) jointly with a low-dimensional representation. Specifically, the fixed shape is fed into the current-splatting layer to transform shape data into image type. Then the current-splatting expression is passed through an encoder to estimate the latent space that encodes deformations of the population of interest. The velocity field can be obtained as the output of the decoder and is applied to the

estimated template to (approximately) reconstruct the input fixed shape. Noticeably, the template is the same for all fixed shapes. A s -Sobolev equivalent norm is adopted as the regularization term on the registration field to encourage smooth deformation.

Evaluations are conducted on the ADNI database (Jack et al. 2008). Compared with principal geodesic analysis (PGA) (Zhang and Fletcher 2014) on reconstruction (on training data) and generalization (on testing data unseen in the training phase), residuals of the hippocampus show that LRShape is better at reconstruction, while PGA is better at generalization. When using the learned 3D latent representations from PGA and LRShape as inputs for classifications of three classes, healthy control (HC: 54 cases), mild cognitive impairment (MCI: 53 cases), and Alzheimer's disease (AD: 53 cases), accuracies of 61.3% versus 58.8% for classifying CN/MCI/AD, 85.0% versus 84.1% for classifying CN/AD, 67.3% versus 67.3% for classifying CN/MCI, and 68.9% versus 71.7% for classifying MCI/AD are obtained from LRShape and PGA.

More Related Works

In addition to these detailedly described methods, there are a number of other related methods also built under unsupervised frameworks. Han et al. (2020) explores a CNN-based learning approach to register images with brain tumors to an atlas. It learns appearance mappings from images with tumors to the atlas and simultaneously predicts the corresponding transformations to the atlas space. Shen et al. (2019a) propose a method that jointly learns affine and diffeomorphic mappings through an end-to-end U-Net. In addition to the regular similarity and regularity terms, it is supervised by an additional symmetric loss. Riemannian manifold learning in association with a statistical task of longitudinal trajectory analyses is studied in Louis et al. (2019), which adopts a RNN to properly process the sequence of longitudinal data. Niethammer et al. (2019) jointly optimize over momenta and the parameters in a CNN of predicting regularizer, constructing a metric such that diffeomorphic transformations can be ensured in the continuum. A deep Laplacian pyramid image registration framework (Mok and Chung 2020b) is proposed in 2020, which is able to solve the optimization problem of image registration in a coarse-to-fine fashion within the space of diffeomorphism. Krebs et al. (2021) recently proposes learning a probabilistic motion model from image sequences for spatio-temporal registration. It encodes motion in a low-dimensional probabilistic space (a motion matrix), enabling various motion tasks such as simulation and interpolation of realistic motion patterns for faster data acquisition and data augmentation. This work is a variant of Krebs et al. (2019) by introducing a novel Gaussian process prior and employing a temporal convolutional network (Lea et al. 2016) for the temporal sequences. Another work Hinkle et al. (2018) aims to create atlas using a form of autoencoder, in which the encoder maps an image to a transformation and the decoder interpolates a deformable template to reconstruct the input. Shen et al. (2019b) describe a region-specific diffeomorphic mapping that

allows for spatial-varying regularization advected via the estimated spatio-temporal velocity field, the framework of which is built based on CNN. Aiming to remove image-data dependency for learning-based methods, Hoffmann et al. (2020) exploit a new direction that leverages a generative model for diverse label maps and images, which exposes the networks to a wide range of variabilities during training. Besides, Detlefsen et al. (2018) employ continuous piecewise-affine-based (CPAB) (Freifeld et al. 2017) diffeomorphic mapping in the tasks of classifying digital numbers and face verification via CNN and show better results over methods without involving diffeomorphic mapping. Amor et al. (2021) proposes a method that uses deep residual networks (He et al. 2016) to implement LDDMM on surfaces and conducts evaluations on a variety of region of interests (ROIs) including the cortex, heart, liver, femur, and hand. Related information of all of the reviewed unsupervised learning works is organized in Table 1.

Supervised Methods

In this section, literature on supervised deep learning-based diffeomorphic mapping methods will be surveyed. The goal of training a supervised network is to obtain the parameterization of the registration field obtained through performing pairwise registration via traditional numerical optimization methods. This parameterization could be momentum, velocity, or displacement.

We will start reviewing the loss function with the most commonly used similarity metrics and several representative regularization ways to ensure diffeomorphism. After that, a variety of CNN-based methods as well as more related works will be covered.

Loss Function

For supervised learning-based diffeomorphic mapping, the loss function also consists of a similarity term and a regularization term. However, the similarity term is completely different from that in an unsupervised method. The typical loss function can be written as follows:

$$L^{\text{sup}} = L_{\text{sim}}(\mathbf{u}^{\text{sup}}, \mathbf{u}) + \gamma L_{\text{reg}}(\mathbf{u}), \quad (8)$$

where L_{sim} is the similarity term that measures the difference between the parameterization \mathbf{u} of the predicted deformation and \mathbf{u}^{sup} obtained from conducting one-to-one registration utilizing traditional methods. L_{reg} is the regularization term that imposes certain constraint on the parameterization of the deformation. When minimizing the loss function, the set of the estimated parameters of the deformation is increasingly closer to the set of the ground truth \mathbf{u}_{sup} , and the registration field is

progressively smoother. γ is a trade-off factor between the similarity term and the regularization term, which behaves similarly as the unsupervised one.

Similarity Metrics

So far, supervised learning-based diffeomorphic mappings are focused on image data and usually employ sum of squared difference (SSD), also called MSE, as the similarity metric:

$$SSD(u^{\text{sup}}, u) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \|u^{\text{sup}} - u\|^2, \quad (9)$$

where p indexes image pixels or voxels and Ω represents the whole image. u and u^{sup} , respectively, represent the predicted parameterization and the one obtained from traditional methods. Since the loss function is minimized to train the framework, a small SSD is desired to yield a good alignment. The similarity loss term with SSD can be directly written as $L_{\text{sim}}(u^{\text{sup}}, u) = SSD(u^{\text{sup}}, u)$.

Regularization for Diffeomorphic Mapping

In a learning-based supervised framework, the LDDMM regularity term (Beg et al. 2005; Glaunes et al. 2008) is usually used when the prediction aims to obtain the registration field of LDDMM. l_2 norm on weights of the networks is also employed for smooth deformation purposes. In addition, a differential operator similar to Eq. 6 or Eq. 7 conducting regularization on momentum or velocity is also a common choice for ensuring diffeomorphism.

CNN-Based Methods

Quicksilver (Yang et al. 2016) proposes a fast predictive image registration method in 2016 which focuses only on atlas-based registration. A later version (Yang et al. 2017b) extends the former work to multi-modal image registration. Quicksilver (Yang et al. 2017c) is an enhanced version of the two previous works. It is a patch-based learning framework that mimics LDDMM by (approximately) predicting LDDMM's momentum through neural networks instead of employing traditional LDDMM. The predicted momentum is constrained by a LDDMM regularity term so as to ensure smooth mapping. Concretely, two patches of size $15 \times 15 \times 15$ of the same location, respectively, taken from the moving image and the fixed image are fed into the framework to learn feature maps, which encode spatial and contextual information of the inputs. The feature maps are subsequently passed through three independent decoding branches with identical network structure to predict the corresponding momentum at the three axes. SSD is employed as the similarity metric to train the network. An extra shooting procedure (Vialard et al. 2012) not included in the network is adopted to perform registration with the

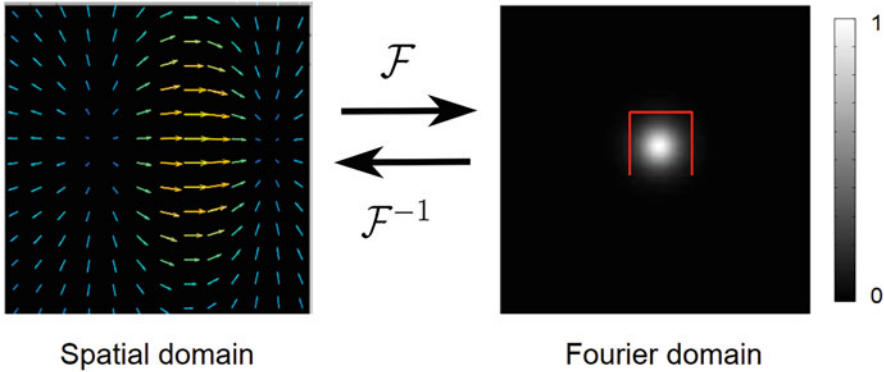


Fig. 8 An example of velocity field in spatial domain and Fourier domain. (Taken from Wang and Zhang 2020b)

predicted momentum. It is worth noting since the input patches are extracted from the whole MRI scans, a large stride 14 of the sliding window for the three axes is preferable considering the computational cost.

Besides, a probabilistic framework is presented to evaluate the registration uncertainty. It assumes the prior on the weights of each layer of the network is a diagonal matrix, each entry of which is drawn from a Bernoulli distribution (a way of drop out). A correction network is additionally proposed to further boost the registration accuracy. Specifically, the momentum predicted in the previous procedure is regarded as an initial prediction and is used to apply backward warp to the fixed patch. The moving patch and the backwardly warped fixed patch are subsequently fed into the correction network to estimate the residual momentum between the initial prediction and the true one (obtained from performing traditional LDDMM) with a residual connection. Results from LDDMM of traditional scheme implemented in PyCA (Singh et al. 2013) with GPU are employed to obtain the supervised labels. There are 3 types of evaluations, including atlas-to-image registration on 150 MRI scans from the OASIS longitudinal dataset (Fotenoš et al. 2005), image-to-image registration on 373 MRI scans from the OASIS longitudinal dataset (Fotenoš et al. 2005) for training and 2168 MRI scans from 4 datasets (LPBA40, IBSR18, MGH10, CUMC12) (Klein et al. 2009) for testing, and multi-modal registration (T1-weighted to T2-weighted) on 375 MRI scans from the IBIS 3D Autism Brain image dataset (Hazlett et al. 2017). Three metrics including target overlap (Yang et al. 2017c), number of voxels whose logarithm Jacobian determinant of the registration field are equal to or less than 0, and deformation errors (mm) are adopted for evaluation purposes. Comparisons are conducted with several related methods with respect to the three metrics.

SVF-Net Also in 2017, a U-shape FCN-based method is published. It takes as inputs pairs of moving and fixed images to predict SVF (Rohé et al. 2017) of the

registration, with SSD as the similarity metric. Unlike Quicksilver which employs independent branches for predicting momentum of each axis, SVF-Net instead estimates the velocity using a 4D map, the last dimension of which, respectively, represents the velocity in x , y , z axes. No explicit regularization is presented in the loss function of SVF-Net. The true velocity labels are obtained by using an iterative log-approximation scheme with the scaling and squaring approach (Arsigny et al. 2006). It starts with the displacement field defined on the whole image grid and parameterizes a transformation that maps a set of selected landmarks from the moving image to the corresponding fixed image.

Inter-patient registration is conducted on 187 segmented 3D MRI cardiac scans acquired from multiple clinical centers. Small translations in x and y axes are performed as data augmentation for the training data. Results with respect to four evaluation metrics (Dice, HD, NLCC, relative variance of Log-Jacobian) for SVF-Net and LCC Log-Demons (Lorenzi et al. 2013) on four ROIs are shown. When performing on a NVIDIA TitanX GPU, SVF-Net takes less than 0.03 s for one pair of registration.

DeepFLASH The aforementioned VAE-based methods in section “VAE-Based Methods” follow a same mechanism: first, they learn a low-dimensional representation of the deformation field; subsequently, a decoder restores the corresponding registration field from the learned compact representation; and finally, registration is performed with the restored registration field. Distinctively, DeepFLASH (Wang and Zhang 2020b) distinguishes itself by predicting a low-dimensional Fourier representation of the velocity field, based on the fact that the velocity field does not develop high frequencies in the Fourier domain, as illustrated in Fig. 8. Thus, the training time and memory consumption can be drastically saved compared to other learning-based methods. To be concrete, DeepFLASH performs Fourier transform on the input moving-and-fixed images and then feeds the real parts of the Fourier representation of the two images into a R_{net} so as to estimate the real part of the Fourier representation for the to-be-predicted velocity field. Meanwhile, the imaginary parts of the Fourier representation of the two images are fed into another network that is parallel to R_{net} , called I_{net} , to estimate the corresponding imaginary part. Structures of R_{net} and I_{net} are identical to each other and are built based on CNN. Once the real and the imaginary parts are obtained, the velocity and the corresponding registration field can be recovered from the predicted low-dimensional representations in the Fourier domain. SSD is employed as the similarity metric which measures the difference between the predicted velocity field in Fourier domain and the ground truth obtained from conducting VM-LDDMM (Singh et al. 2013) as well as Fourier transformation. l_2 norm on weights of the network is adopted serving as the regularity term in the loss function.

Experiments are conducted on 3200 public T1-weighted 3D brain MRI scans from ADNI (Jack et al. 2008), OASIS (Foteno et al. 2005), ABIDE (Di Martino et al. 2014), and LPBA40 (Shattuck et al. 2008) with 1000 subjects involved. Results are compared with three traditional methods, VM-LDDMM (Singh et al. 2013), ANTs SyN (Avants et al. 2008), and FLASH (Zhang and Fletcher 2019), as well as

two learning-based methods: Quicksilver (Yang et al. 2017c) and VoxelMorph (Balakrishnan et al. 2019). When comparing Dice scores among these methods, 0.780 for DeepFLASH, 0.774 for VoxelMorph, 0.762 for Quicksilver, 0.788 for FLASH, 0.770 for ANTs SyN, and 0.760 for VM-LDDMM are obtained. Considering the training time, DeepFLASH takes 14.1 h, VoxelMorph takes 29.7 h, and Quicksilver takes 31.4 h under the same conditions. However, both DeepFLASH and Quicksilver need extra time for acquiring the registration labels through conducting conventional methods before the training procedure. The registration time on NVIDIA GTX 1080Ti GPUs is, respectively, 0.273 s for DeepFLASH, 0.571 s for VoxelMorph, 0.760 s for Quicksilver, 53.4 s for FLASH, and 262 s for VM-LDDMM.

More Related Works

Besides, Krebs et al. (2017) explores training a reinforcement learning model with a large number of synthetically deformed image pairs and a small number of real inter-subject pairs through agent-based action learning. Pathan and Hong (2018) combine LSTM and CNN to learn a predictive regression model based on LDDMM for longitudinal images with missing data. Ding proposes a framework similar to Quicksilver, called FPSGR (Ding et al. 2019; Kwitt and Niethammer 2017), to approximate a simplified geodesic regression model so as to capture longitudinal brain changes. To be specific, FPSGR predicts initial momenta supervised by the geodesic distance between images. The geodesic regression can be solved by approximately performing pairwise image registrations between the first image and all subsequent images of the longitudinal data. FPSGR-derived correlations with clinical indicators are also analyzed. A work on arXiv (Wang and Zhang 2020a) first estimates the regularity parameters of the image registrations for given image pairs using a CNN. Afterward, a new two-stream CNN-based network is trained to estimate the mapping from image pairs to their corresponding regularity parameters, under the supervision of the estimated regularity parameters from the previous step. Table 2 lists the related information of all reviewed supervised learning-based works.

Discussion and Future Direction

Achievements and Applications

Heretofore, deep learning-based diffeomorphic mapping, whether unsupervised or supervised, can achieve comparable or even better results than the state-of-the-art traditional methods (Lorenzi et al. 2013; Avants et al. 2008; Singh et al. 2013) when performing registrations within the same underlyingly assumed population as the training data. Besides, the time consumed by each pair of registration is considerably reduced, thanks to the efficient parallel computations of GPU and the ability of deep networks to learn and store registration mappings. In addition, atlases can be conveniently generated by VAE-based methods. Registration uncertainty and

sampling new deformation as well as conducting deformation transport can also be achieved by training a probabilistic generative model. Furthermore, incorporating time sequence data and temporal convolutional networks can jointly predict registration fields for sequential data and perform progression analyses of diseases such as Alzheimer's disease. The aforementioned methods mainly focus on using deep neural networks to perform registration tasks in a diffeomorphic way. The works that focus on specific applications making use of these deep learning-based registration methods have also emerged recently.

Dalca et al. (2019b) propose a strategy that combines a conventional probabilistic atlas-based segmentation method with a deep learning-based registration method, being able to train a model for segmenting new testing MRI scans without any manually segmented images involved in the training phase. An efficient method for yielding either universal or conditional templates and jointly performing registration between images and templates is presented in Dalca et al. (2019a). In Evan et al. (2020), a model which learns to compute an attribute-specific spatial deformation is proposed. This model can deform a brain template in certain ways that take a wide range of ages, presence of diseases, and different genders into consideration. Cheng et al. (2020) focus on cortical surface registration utilizing unsupervised learning. Olut et al. (2020) use deformations obtained from deep registration models to conduct data augmentation. Specifically, it builds statistical deformation models based on unlabeled data using principal component analysis and subsequently uses the acquired statistical deformation space to augment training samples with labels.

Challenges

Nevertheless, there are still a variety of challenges presented to researchers. Learning-based techniques can only accurately register objects that come from the same population as in training. To be concrete, these methods can merely register images whose image contrast and geometric content are similar to those of the training data. This limitation comes from the inherent property of deep learning; it can only capture and store characteristics of data involved during training. For instance, when we use a deep registration network trained on T1-weighted MRI scans to register T2-weighted MRI scans or other modalities such as CT scans, the performance is usually inferior and much lower than that of performing registration between T1-weighted pairs. Besides, medical imaging scans of the same ROI obtained from different machines or different sites could be of various distributions even within the same modality. Thus, challenges still need to be solved to acquire the desirable property of conventional methods, namely, being able to register any type of data rather than only those involved in training. This limitation is also applicable to shape data. Furthermore, when comes to shape data, existing deep registration frameworks usually first transform shapes into image representations and then feed the transformed image representations into deep neural networks for subsequent procedures. However, this kind of image representation may deteriorate the resolution of the original shape. This is due to the fact that only values on the

grid are considered and no strong constraint of the original shape is involved in the network structures. Thus, how to design a more suitable deep registration framework for shape data remains a challenging topic to explore.

Future Directions

As a newly emerging topic, deep learning-based diffeomorphic mapping demonstrates promising potentials to improve or exploit in several directions. For cross-modality or cross-ROI registrations (train on one modality or ROI but perform registration on another modality or ROI), the recent approaches of domain adaptation (Sun et al. 2015; Wilson and Cook 2020) and domain generalization (Li et al. 2018; Zhou et al. 2021) might serve as potential solutions. To be specific, if training data from a new domain are available, domain adaptation methods can be used to fine-tune the deep registration networks so as to make the tuned networks applicable for the new data. On the contrary, domain generalization methods can serve as a technique to handle data from an unseen domain if training data are unavailable. Additionally, a generative adversarial network (Yi et al. 2019) might be used to improve the performance of the registration. A generator produces the deformation fields, while the discriminator evaluates whether they are good or not. The zero-sum game can significantly contribute to improving the quality and authenticity of the deformation fields. As for applications, deep learning-based registration frameworks, especially for surfaces and curves, can explicitly incorporate geometrical information into neural networks. This is potentially beneficial for other tasks such as more regular and smooth organ segmentations or more accurate landmark detections. As far as deep learning-based shape registration is concerned, an elaborately designed network suitable for handling meshes could further improve the registration performance.

Conclusions

In this chapter, we firstly describe the conventional diffeomorphic registration problem and its general objectivities. Afterward, several deep neural networks used in learning-based diffeomorphic mapping are briefly introduced. Subsequently, the general loss functions, similarity metrics, regularity terms, and recent works of deep registration frameworks, both unsupervised and supervised, are examined in detail. Several data types such as MRI, CT, surface, and curve are covered in these works. In addition, we summarize current achievements, applications, and challenges in this field. Finally, we provide several potential future directions to explore at the end of this chapter.

Acknowledgments This study was supported by the National Natural Science Foundation of China (62071210); the Shenzhen Basic Research Program (JCYJ20200925153847004, JCYJ20190809120205578); and the High-Level University Fund (G02236002). The authors would like to thank Yuanyuan Wei from the University of British Columbia for his help on this chapter.

References

- Amor, B.B., Arguillère, S., Shao, L.: Resnet-LDDMM: advancing the LDDMM framework using deep residual networks (2021). arXiv preprint arXiv:210207951
- Arganda-Carreras, I., Turaga, S.C., Berger, D.R., Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J.M., et al.: Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front. Neuroanat.* **9**, 142 (2015)
- Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and simple calculus on tensors in the log-Euclidean framework. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 115–122. Springer (2005)
- Arsigny, V., Commowick, O., Pennec, X., Ayache, N.: A log-Euclidean framework for statistics on diffeomorphisms. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 924–931. Springer (2006)
- Ashburner, J.: A fast diffeomorphic image registration algorithm. *Neuroimage* **38**(1), 95–113 (2007)
- Avants, B.B., Epstein, C.L., Grossman, M., Gee, J.C.: Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Med. Image Anal.* **12**(1), 26–41 (2008)
- Balakrishnan, G., Zhao, A., Sabuncu, M.R., Gutttag, J., Dalca, A.V.: An unsupervised learning model for deformable medical image registration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9252–9260 (2018)
- Balakrishnan, G., Zhao, A., Sabuncu, M.R., Gutttag, J., Dalca, A.V.: Voxelmorph: a learning framework for deformable medical image registration. *IEEE Trans. Med. Imaging* **38**(8), 1788–1800 (2019)
- Beg, M.F., Miller, M.I., Trounev, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **61**(2), 139–157 (2005)
- Bône, A., Louis, M., Colliot, O., Durrleman, S., Initiative, A.D.N., et al.: Learning low-dimensional representations of shape data sets with diffeomorphic autoencoders. In: *International Conference on Information Processing in Medical Imaging*, pp. 195–207. Springer (2019)
- Bossa, M., Zacur, E., Olmos, S., Initiative, A.D.N., et al.: Tensor-based morphometry with stationary velocity field diffeomorphic registration: application to ADNI. *Neuroimage* **51**(3), 956–969 (2010)
- Cao, Y., Miller, M.I., Mori, S., Winslow, R.L., Younes, L.: Diffeomorphic matching of diffusion tensor images. In: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pp. 67–67. IEEE (2006)
- Charon, N., Trounev, A.: The varifold representation of nonoriented shapes for diffeomorphic registration. *SIAM J. Imaging Sci.* **6**(4), 2547–2580 (2013)
- Cheng, J., Dalca, A.V., Fischl, B., Zöllei, L., Initiative, A.D.N., et al.: Cortical surface registration using unsupervised learning. *NeuroImage* **221**, 117161 (2020)
- Dalca, A.V., Balakrishnan, G., Gutttag, J., Sabuncu, M.R.: Unsupervised learning for fast probabilistic diffeomorphic registration. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 729–738. Springer (2018)
- Dalca, A.V., Rakic, M., Gutttag, J., Sabuncu, M.R.: Learning conditional deformable templates with convolutional networks (2019a). arXiv preprint arXiv:190802738
- Dalca, A.V., Yu, E., Golland, P., Fischl, B., Sabuncu, M.R., Iglesias, J.E.: Unsupervised deep learning for Bayesian brain MRI segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 356–365. Springer (2019b)
- Debavelaere, V., Durrleman, S., Allasonnière, S., Initiative, A.D.N.: Learning the clustering of longitudinal shape data sets into a mixture of independent or branching trajectories. *Int. J. Comput. Vis.* **128**, 2794–2809 (2020)
- Detlefsen, N.S., Freifeld, O., Hauberg, S.: Deep diffeomorphic transformer networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4403–4412 (2018)

- Di Martino, A., Yan, C.G., Li, Q., Denio, E., Castellanos, F.X., Alaerts, K., Anderson, J.S., Assaf, M., Bookheimer, S.Y., Dapretto, M., et al.: The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Mol. Psychiatry* **19**(6), 659–667 (2014)
- Ding, Z., Fleishman, G., Yang, X., Thompson, P., Kwitt, R., Niethammer, M., Initiative, A.D.N., et al.: Fast predictive simple geodesic regression. *Med. Image Anal.* **56**, 193–209 (2019)
- Durrleman, S.: Statistical models of currents for measuring the variability of anatomical curves, surfaces and their evolution. PhD thesis, Université Nice Sophia Antipolis (2010)
- Evan, M.Y., Dalca, A.V., Sabuncu, M.R.: Learning conditional deformable shape templates for brain anatomy. In: International Workshop on Machine Learning in Medical Imaging, pp. 353–362. Springer (2020)
- Fischl, B.: Freesurfer. *Neuroimage* **62**(2), 774–781 (2012)
- Fotenos, A.F., Snyder, A., Girton, L., Morris, J., Buckner, R.: Normative estimates of cross-sectional and longitudinal brain volume decline in aging and ad. *Neurology* **64**(6), 1032–1039 (2005)
- Freifeld, O., Hauberg, S., Batmanghelich, K., Fisher, J.W.: Transformations based on continuous piecewise-affine velocity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(12), 2496–2509 (2017)
- Glaunes, J., Qiu, A., Miller, M.I., Younes, L.: Large deformation diffeomorphic metric curve mapping. *Int. J. Comput. Vis.* **80**(3), 317–336 (2008)
- Gondara, L.: Medical image denoising using convolutional denoising autoencoders. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pp. 241–246. IEEE (2016)
- Gori, P., Colliot, O., Marrakchi-Kacem, L., Worbe, Y., Poupon, C., Hartmann, A., Ayache, N., Durrleman, S.: A Bayesian framework for joint morphometry of surface and curve meshes in multi-object complexes. *Med. Image Anal.* **35**, 458–474 (2017)
- Han, X., Shen, Z., Xu, Z., Bakas, S., Akbari, H., Bilello, M., Davatzikos, C., Niethammer, M.: A deep network for joint registration and reconstruction of images with pathologies. In: International Workshop on Machine Learning in Medical Imaging, pp. 342–352. Springer (2020)
- Hazlett, H.C., Gu, H., Munsell, B.C., Kim, S.H., Styner, M., Wolff, J.J., Elison, J.T., Swanson, M.R., Zhu, H., Botteron, K.N., et al.: Early brain development in infants at high risk for autism spectrum disorder. *Nature* **542**(7641), 348–351 (2017)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- Hinkle, J., Womble, D., Yoon, H.J.: Diffeomorphic autoencoders for LDDMM atlas building (2018)
- Hoffmann, M., Billot, B., Eugenio Iglesias, J., Fischl, B., Dalca, A.V.: Learning image registration without images (2020). arXiv e-prints arXiv:2004
- Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: implementing efficient convnet descriptor pyramids (2014). arXiv preprint arXiv:14041869
- Jack, C.R. Jr, Bernstein, M.A., Fox, N.C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P.J., Whitwell, J.L., Ward, C., et al.: The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging: Off. J. Int. Soc. Magn. Res. Med.* **27**(4), 685–691 (2008)
- Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks (2015). arXiv preprint arXiv:150602025
- Jiang, Z., Yang, H., Tang, X.: Deformation-based statistical shape analysis of the corpus callosum in mild cognitive impairment and Alzheimer’s disease. *Curr. Alzheimer Res.* **15**(12), 1151–1160 (2018)
- Joshi, S.C., Miller, M.I.: Landmark matching via large deformation diffeomorphisms. *IEEE Trans. Image Process.* **9**(8), 1357–1370 (2000)
- Karpathy, A., Johnson, J., Fei-Fei, L.: Visualizing and understanding recurrent networks (2015). arXiv preprint arXiv:150602078

- Kaul, C., Manandhar, S., Pears, N.: Focusnet: An attention-based fully convolutional network for medical image segmentation. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 455–458. IEEE (2019)
- Kingma, D.P., Welling, M.: Auto-encoding variational Bayes (2013). arXiv preprint arXiv:1312.6114
- Klein, A., Andersson, J., Ardekani, B.A., Ashburner, J., Avants, B., Chiang, M.C., Christensen, G.E., Collins, D.L., Gee, J., Hellier, P., et al.: Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *Neuroimage* **46**(3), 786–802 (2009)
- Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **37**(2), 233–243 (1991)
- Krebs, J., Mansi, T., Delingette, H., Zhang, L., Ghesu, F.C., Miao, S., Maier, A.K., Ayache, N., Liao, R., Kamen, A.: Robust non-rigid registration through agent-based action learning. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 344–352. Springer (2017)
- Krebs, J., Delingette, H., Mailhé, B., Ayache, N., Mansi, T.: Learning a probabilistic model for diffeomorphic registration. *IEEE Trans. Med. Imaging* **38**(9), 2165–2176 (2019)
- Krebs, J., Delingette, H., Ayache, N., Mansi, T.: Learning a generative motion model from image sequences based on a latent motion matrix. *IEEE Trans. Med. Imaging* **40**(5), 1405–1416 (2021)
- Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25**, 1097–1105 (2012)
- Kwitt, R., Niethammer, M.: Fast predictive simple geodesic regression. In: Third International Workshop DLMIA, p. 267 (2017)
- Lateef, F., Ruichek, Y.: Survey on semantic segmentation using deep learning techniques. *Neurocomputing* **338**, 321–348 (2019)
- Lea, C., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks: a unified approach to action segmentation. In: European Conference on Computer Vision, pp. 47–54. Springer (2016)
- Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: meta-learning for domain generalization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: a survey. *Int. J. Comput. Vis.* **128**(2), 261–318 (2020)
- Lorenzi, M., Ayache, N., Frisoni, G.B., Pennec, X., ADNI, et al.: LCC-demons: a robust and accurate symmetric diffeomorphic registration algorithm. *NeuroImage* **81**, 470–483 (2013)
- Louis, M., Charlier, B., Durrleman, S.: Geodesic discriminant analysis for manifold-valued data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 332–340 (2018)
- Louis, M., Couronné, R., Koval, I., Charlier, B., Durrleman, S.: Riemannian geometry learning for disease progression modelling. In: International Conference on Information Processing in Medical Imaging, pp. 542–553. Springer (2019)
- Lyu, J., Cheng, P., Tang, X.: Fundus image based retinal vessel segmentation utilizing a fast and accurate fully convolutional network. In: International Workshop on Ophthalmic Medical Image Analysis, pp. 112–120. Springer (2019)
- Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571. IEEE (2016)
- Modat, M., Daga, P., Cardoso, M.J., Ourselin, S., Ridgway, G.R., Ashburner, J.: Parametric non-rigid registration using a stationary velocity field. In: 2012 IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 145–150. IEEE (2012)
- Mok, T.C., Chung, A.: Fast symmetric diffeomorphic image registration with convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4644–4653 (2020a)
- Mok, T.C., Chung, A.C.: Large deformation diffeomorphic image registration with laplacian pyramid networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 211–221. Springer (2020b)

- Niethammer, M., Kwitt, R., Vialard, F.X.: Metric learning for image registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8463–8472 (2019)
- Olut, S., Shen, Z., Xu, Z., Gerber, S., Niethammer, M.: Adversarial data augmentation via deformation statistics. In: European Conference on Computer Vision, pp. 643–659. Springer (2020)
- Pathan, S., Hong, Y.: Predictive image regression for longitudinal studies with missing data (2018). arXiv preprint arXiv:180807553
- Rohé, M.M., Datar, M., Heimann, T., Sermesant, M., Pennec, X.: SVF-Net: learning deformable image registration using shape matching. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 266–274. Springer (2017)
- Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
- Roy, A.G., Navab, N., Wachinger, C.: Concurrent spatial and channel ‘squeeze & excitation’ in fully convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 421–429. Springer (2018)
- Shattuck, D.W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K.L., Poldrack, R.A., Bilder, R.M., Toga, A.W.: Construction of a 3d Probabilistic Atlas of Human Cortical Structures. *Neuroimage* **39**(3), 1064–1080 (2008)
- Shen, Z., Han, X., Xu, Z., Niethammer, M.: Networks for joint affine and non-parametric image registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4224–4233 (2019a)
- Shen, Z., Vialard, F.X., Niethammer, M.: Region-specific diffeomorphic metric mapping (2019b). arXiv preprint arXiv:190600139
- Sibi, P., Jones, S.A., Siddarth, P.: Analysis of different activation functions using back propagation neural networks. *J. Theor. Appl. Inf. Technol.* **47**(3), 1264–1268 (2013)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:14091556
- Singh, N., Hinkle, J., Joshi, S., Fletcher, P.T.: A vector momenta formulation of diffeomorphisms for improved geodesic regression and atlas construction. In: 2013 IEEE 10th International Symposium on Biomedical Imaging, pp. 1219–1222. IEEE (2013)
- Sun, S., Shi, H., Wu, Y.: A survey of multi-source domain adaptation. *Inf. Fusion* **24**, 84–92 (2015)
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
- Tang, X., Ross, C.A., Johnson, H., Paulsen, J.S., Younes, L., Albin, R.L., Ratnanather, J.T., Miller, M.I.: Regional subcortical shape analysis in premanifest Huntington’s disease. *Hum. Brain Map.* **40**(5), 1419–1433 (2019)
- Tian, L., Puett, C., Liu, P., Shen, Z., Aylward, S.R., Lee, Y.Z., Niethammer, M.: Fluid registration between lung CT and stationary chest tomosynthesis images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 307–317. Springer (2020)
- Ulman, V., Maška, M., Magnusson, K.E., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., et al.: An objective comparison of cell-tracking algorithms. *Nat. Methods* **14**(12), 1141–1152 (2017)
- Vaillant, M., Glaunes, J.: Surface matching via currents. In: Biennial International Conference on Information Processing in Medical Imaging, pp. 381–392. Springer (2005)
- Vaillant, M., Qiu, A., Glaunès, J., Miller, M.I.: Diffeomorphic metric surface mapping in subregion of the superior temporal gyrus. *NeuroImage* **34**(3), 1149–1159 (2007)
- Vialard, F.X., Risser, L., Rueckert, D., Cotter, C.J.: Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation. *Int. J. Comput. Vis.* **97**(2), 229–241 (2012)
- Wang, J., Zhang, M.: Deep learning for regularization prediction in diffeomorphic image registration (2020a). arXiv preprint arXiv:201114229

- Wang, J., Zhang, M.: Deepflash: an efficient network for learning-based medical image registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4444–4452 (2020b)
- Wang, W., Huang, Y., Wang, Y., Wang, L.: Generalized autoencoder: a neural network framework for dimensionality reduction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 490–497 (2014)
- Wilson, G., Cook, D.J.: A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol. (TIST)* **11**(5), 1–46 (2020)
- Yang, X., Li, Y., Reutens, D., Jiang, T.: Diffeomorphic metric landmark mapping using stationary velocity field parameterization. *Int. J. Comput. Vis.* **115**(2), 69–86 (2015)
- Yang, X., Kwitt, R., Niethammer, M.: Fast predictive image registration. In: Deep Learning and Data Labeling for Medical Applications, pp. 48–57. Springer, Cham (2016)
- Yang, H., Wang, J., Tang, H., Ba, Q., Yang, G., Tang, X.: Analysis of mitochondrial shape dynamics using large deformation diffeomorphic metric curve matching. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4062–4065. IEEE (2017a)
- Yang, X., Kwitt, R., Styner, M., Niethammer, M.: Fast predictive multimodal image registration. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pp. 858–862. IEEE (2017b)
- Yang, X., Kwitt, R., Styner, M., Niethammer, M.: Quicksilver: fast predictive image registration—a deep learning approach. *NeuroImage* **158**, 378–396 (2017c)
- Yi, X., Walia, E., Babyn, P.: Generative adversarial network in medical imaging: a review. *Med. Image Anal.* **58**, 101552 (2019)
- Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2528–2535. IEEE (2010)
- Zhang, M., Fletcher, P.T.: Bayesian principal geodesic analysis in diffeomorphic image registration. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 121–128. Springer (2014)
- Zhang, M., Fletcher, P.T.: Fast diffeomorphic image registration via fourier-approximated lie algebras. *Int. J. Comput. Vis.* **127**(1), 61–73 (2019)
- Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674 (2017)
- Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: a survey (2021). arXiv preprint arXiv:210302503

Part III
Shape Spaces and Geometric Flows



Alexis Arnaudon, Darryl Holm, and Stefan Sommer

Contents

Introduction	1326
Key Concepts from Shape Analysis	1328
Large Deformation Diffeomorphic Metric Mapping	1328
Metric and Variational Formulations	1329
Hamiltonian Systems and Noise	1330
Hamiltonian Systems and Landmark Dynamics	1330
Noise from a Statistical Physics Perspective	1332
Non-dissipative Stochastic Shape Models	1333
Riemannian Brownian Motion	1334
Lagrangian Noise	1335
Eulerian Noise	1336
Stochastic Euler-Poincaré Reduction and Its Infinite Dimensional Extension	1337
Reduction by Symmetry	1338
Stochastic EPDiff	1339
Other Stochastic Shape Models	1340
Stochastic Models in Shape Statistics	1340
Random Orbits with Time-Continuous Noise	1340
Noise Inference from Evolution of Moments	1341
Likelihood-Based Inference and Bridge Sampling	1342
Likelihood Maximisation and Automatic Differentiation	1343

A. Arnaudon

Department of Mathematics, Imperial College, London, UK

Blue Brain Project, École polytechnique fédéral de Lausanne (EPFL), Geneva, Switzerland

e-mail: alexis.arnaudon@epfl.ch; alexis.arnaudon@imperial.ac.uk

D. Holm

Department of Mathematics, Imperial College, London, UK

e-mail: d.holm@imperial.ac.uk

S. Sommer (✉)

Department of Computer Science (DIKU), University of Copenhagen, Copenhagen E, Denmark

e-mail: sommer@di.ku.dk

Applications and Extensions.....	1344
Conclusion and Outlook.....	1346
References.....	1346

Abstract

The chapter describes stochastic models of shapes from a Hamiltonian viewpoint, including Langevin models, Riemannian Brownian motions and stochastic variational systems. Starting from the deterministic setting of outer metrics on shape spaces and transformation groups, we discuss recent approaches to introducing noise in shape analysis from a physical or Hamiltonian point of view. We furthermore outline important applications and statistical uses of stochastic shape models, and we discuss perspectives and current research efforts in stochastic shape analysis.

Keywords

Shape analysis · Stochastic geometric mechanics · Hamiltonian systems · Langevin equations · Stochastic Euler-Poincaré equations

Mathematics Subject Classification (2010)

60G99 · 70H99 · 65C30

Introduction

Shape analysis is a vast topic that can be approached from many angles including geometry, analysis, statistics and numerical analysis. Shape modelling and analysis similarly finds application in a range of domains including biology, medical image analysis, computer vision, computer graphics and engineering.

The mathematical study of shapes often involves geometric methods due to the inherent nonlinearity of shape spaces. Examples include the setting of *inner* metrics (Bauer et al. 2014), where a Riemannian structure is defined directly on the shape space or the pattern theory pioneered by Grenander (1994) and advanced farther by, e.g. Miller, Christensen, Trouvé and Younes (Christensen et al. 1996; Grenander and Miller 1998; Younes 1998; Trouvé 1998) where sets of transformations of the shape domain are equipped with geometric structure. Specifically, in the latter approach, a right-invariant Riemannian metric is defined on a subgroup of the diffeomorphism group, whose action on shapes descends to a Riemannian metric on the shape space itself. Due to the transformation of the entire domain in which the shape resides, this class of metrics is denoted *outer* metrics.

In both settings, Riemannian geometric structure is defined on the shape space. Then, the optimal trajectory between two shapes, a matching of the shapes, is a

geodesic, and the dual of the Riemannian metric defines a Hamiltonian of which the geodesic satisfies Hamilton's equations. In the outer metric case, the Hamiltonian is defined on both the shape space and the transformation group, and due to the invariance of the metric, the concepts of momentum maps and symmetry reduction of the Hamiltonian flow have important roles (Holm et al. 2004; Bruveris et al. 2009).

The foregoing models treat shape transformation as *deterministic* smooth evolutions in the shape space. However, both from applied and theoretical perspectives, it has been of recent interest to generalise these models to admit *stochastic* transformations of shapes. For example, evolutionary biology incorporates stochasticity in the models of species change, organs may evolve stochastically during the development of a disease, and stochastic processes define probability distributions which can be used for statistical analysis.

Several stochastic shape models exist, each with different properties. With the variety of shape problems that require randomness, several stochastic frameworks must be available. It is therefore relevant to have several models, each with different properties. Here, we will focus on four models that are applicable to at least the simplest shape representation with landmarks. In Fig. 1, we illustrate these four models by solving an initial value problem forward from a configuration of 21 landmarks:

- (a) *Riemannian Brownian motion*: this noise corresponds to pure Brownian motion but on the shape manifold. It does not have any initial momenta and has little spatial correlation.
- (b) *Langevin dynamics*: landmarks are interpreted as interacting particles in a heat bath; it has an initial momenta and noise on the momentum, so more regular trajectories. The dissipation term slows down the landmark trajectories.

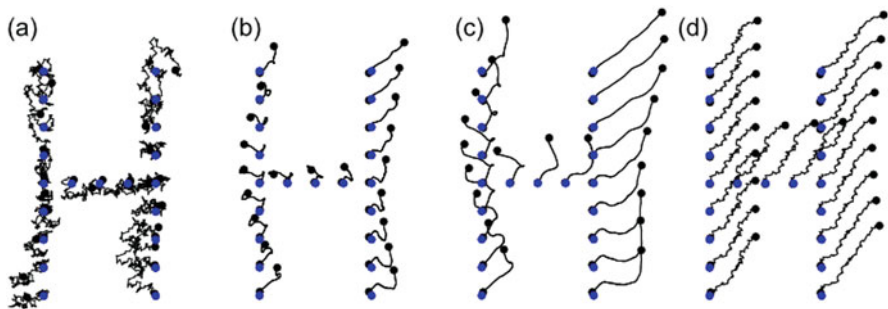


Fig. 1 Examples of stochastically evolving landmarks configurations from a H shape (in blue dots) up to time 1 (dark dots) with models surveyed in this chapter: (a) Riemannian Brownian motion (Sommer et al. 2017); (b) Lagrangian noise (Trounev and Vialard 2012); (c) Langevin dynamics (Marsland and Shardlow 2017); (d) Eulerian noise (Arnaudon et al. 2019a). See the text for more details on these noise models

- (c) *Lagrangian noise*: landmarks have their own intrinsic additive noise and initial momenta, as in the Langevin dynamics, but without dissipation; thus, trajectories move further in space for the same initial momenta. The spatial correlation of the trajectories results only from the interactions of landmarks.
- (d) *Eulerian noise*: the image has noise fields encoding spatially correlated noise (a grid of 4 by 4 Gaussian kernels uniformly covering the landmark trajectory space), on which landmarks move and interact. This model has initial momenta, and noisy trajectories, but with full control on the spatial correlation of the noise via the noise fields.

Chapter content. The chapter starts with a short outline of key concepts from non-stochastic shape analysis in section “[Key Concepts from Shape Analysis](#)” focusing on the outer metric viewpoint. We then review basic Hamiltonian dynamics with and without noise in [Hamiltonian Systems and Noise](#), with a statistical physics perspective; then, in section “[Non-dissipative Stochastic Shape Models](#)”, we describe three of four main stochastic models for landmarks dynamics, illustrated in Fig. 1. The last model, with Eulerian noise, can be extended to other types of shape spaces in a geometrical way, which is described in section “[Stochastic Euler-Poincaré Reduction and Its Infinite Dimensional Extension](#)”. A few other approaches to stochastic shape analysis are outlined in section “[Other Stochastic Shape Models](#)”, and some applications in statistics of shapes are described in section “[Stochastic Models in Shape Statistics](#)”. We end with a discussion section in “[Conclusion and Outlook](#)”.

Key Concepts from Shape Analysis

This section briefly outlines the non-stochastic shape space theory, as a basis on which stochastic extensions will be constructed in the rest of the chapter. We refer the reader to texts such as Younes (2010) and Marsland and Sommer (2020) for further details or Holm (2011) for the geometric mechanics foundations for the theory. The constructions we describe below fall into the category of outer metrics, specifically the *large deformation diffeomorphic metric mapping* (LDDMM, Christensen et al. 1996; Grenander and Miller 1998; Younes 1998; Trouvé 1998) framework.

Large Deformation Diffeomorphic Metric Mapping

The starting point of LDDMM is the action of the diffeomorphism group $\text{Diff}(\Omega)$ of a domain $\Omega \subseteq \mathbb{R}^d$ on shapes being landmarks, curves, surfaces, images or tensor fields. We here define the shape space and action in two of those cases, landmarks and curves:

Landmarks: A set of n distinct landmarks with position $\mathbf{q} = (q_1, \dots, q_n)$ in $\Omega \subseteq \mathbb{R}^d$ is denoted a landmark configuration. The shape space $\mathcal{S} = \{\mathbf{q} | q_i \in \mathbb{R}^d, q_i \neq$

$q_j, i \neq j$ of all such configurations is denoted the landmark shape space. It inherits a differentiable structure from its natural embedding as an open subset of \mathbb{R}^{nd} . Notice in particular that the manifold is non-compact as pairs of landmarks can come arbitrarily close but cannot overlap. Finally, the action of $\phi \in \text{Diff}(\Omega)$ on \mathcal{S} is by composition $\phi.\mathbf{q} = (\phi(q_1), \dots, \phi(q_n))$.

Curves: Consider the space of closed curves defined by maps $\gamma : \mathbf{S}^1 \rightarrow \Omega$ from the unit circle \mathbf{S}^1 to the shape domain Ω . Often the space is restricted to embeddings, i.e. requiring γ to be without self-intersections and with nowhere-vanishing derivative. The action of $\phi \in \text{Diff}(\Omega)$ on the space \mathcal{S} of embedded curves is again by composition: $\phi.\gamma = \phi \circ \gamma$.

While we have denoted a landmark shape by \mathbf{q} and a curve by γ , we will in the following use \mathbf{q} for all shapes since the shapes will appear as the state variable in Hamilton's equations. Notice we use bold font only for the entire configuration \mathbf{q} , not the indexed landmarks q_i in the configuration, similarly for the momentum \mathbf{p} defined below.

In addition to the above examples, spaces of surfaces, images and tensor fields admit actions on $\text{Diff}(\Omega)$. Consequently, these can be formally treated in the same mathematical framework as landmarks and surfaces. The landmarks constitute the simplest example of a finite dimensional shape space, while curves are a simple example of an infinite dimensional shape space.

Metric and Variational Formulations

One aim of shape analysis is to define a good notion of distance between shapes. The LDDMM approach starts with the problem of shape matching through the optimisation problem

$$\min_v E(v), \quad E(v) = \int_0^1 \|v(t)\|_V^2 + \frac{1}{2\lambda^2} S(\phi(1).\mathbf{q}^0, \mathbf{q}^1), \quad (1)$$

where $\mathbf{q}^0, \mathbf{q}^1 \in \mathcal{S}$ are generically two shapes, $v(t), t \in [0, 1]$ is a t -dependent family of vector fields on the same domain Ω , $\|\cdot\|_V^2$ is a norm and $S(\phi(1).\mathbf{q}^0, \mathbf{q}^1)$ is a measure of the dissimilarity between \mathbf{q}^1 and the deformed shape $\phi(1).\mathbf{q}^0$. The diffeomorphism $\phi(1) \in \text{Diff}(\Omega)$ encoding the deformation of \mathbf{q}^0 is the endpoint of the ODE:

$$\partial_t \phi(t) = v(t) \circ \phi(t), \quad (2)$$

integrated from $t = 0$ to $t = 1$ with $\phi(0) = \text{Id}_\Omega$. The norm $\|\cdot\|_V^2$ on a subset V of the vector fields $\mathfrak{X}(\Omega)$ used for the first term in (1) is often defined from an operator $L : \mathfrak{X}(\Omega) \rightarrow \mathfrak{X}^*(\Omega)$, so that $\|v\|_V^2 = \langle Lv, v \rangle$ using the L^2 -pairing $\langle \cdot, \cdot \rangle : V^* \times V \rightarrow \mathbb{R}$. L , often denoted the momentum operator, has an inverse in the kernel mapping:

$$K : V^*(\Omega) \rightarrow V(\Omega), \tag{3}$$

which makes V a *reproducing kernel Hilbert space* (see, e.g. Younes 2010).

The variational formulation has a number of important consequences:

Riemannian structure. For v minimising (1), the corresponding diffeomorphism flow ϕ is a geodesic with respect to a right-invariant Riemannian metric on $\text{Diff}(\Omega)$ defined by the norm $\|v \circ \phi\|_\phi^2 = \|v\|_V^2$. This Riemannian metric descends to a Riemannian metric on the shape space \mathcal{S} , and the shape curve $t \mapsto \phi(t) \cdot \mathbf{q}^0$ is a geodesic on \mathcal{S} for this metric when v minimises (1).

Hamiltonian dynamics. The norm $\|v\|_V^2$ also defines a Hamiltonian

$$H(\phi, m) = \frac{1}{2} \|Km\|_V^2, \tag{4}$$

for a momentum field $m(t) := Lv(t) \in V^*$, and the co-metric, or kernel K . In this case, the pair $(\phi(t), m(t))$ satisfies Hamilton’s equations; see below. The Hamiltonian is kinetic energy for the Riemannian metric on $\text{Diff}(\Omega)$. As for the Riemannian metric, the Hamiltonian descends to a Hamiltonian on the shape space \mathcal{S} , and this Hamiltonian $H(\mathbf{q}, \mathbf{p})$ ($\mathbf{q}, \mathbf{p} \in T_{\mathbf{q}}^*\mathcal{S}$) is as well kinetic energy for the Riemannian metric on \mathcal{S} . This Hamiltonian is the main object we will work with below.

Lagrangian dynamics. We can equivalently work with a Lagrangian

$$\ell(\phi, v) = \frac{1}{2} \|v \circ \phi\|_\phi^2 = \frac{1}{2} \|v\|_V^2, \tag{5}$$

and derive equivalent dynamics via the Euler-Lagrange equations.

In all three cases, extremal flows for the variational principle are determined uniquely by their initial conditions. On the diffeomorphism side, this is the velocity field at time $t = 0$, i.e. $v(0)$. On the shape side, this is the starting configuration $\mathbf{q}(0)$ and the momentum (covector) $\mathbf{p}(0)$ (velocity vector in Lagrangian dynamics). See illustration in Fig. 2.

Hamiltonian Systems and Noise

Hamiltonian Systems and Landmark Dynamics

Let’s focus on the deterministic Hamiltonian dynamics for a moment. Given a Hamiltonian, for example, for a massive particle in a potential such as $H(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^2}{2m} + U(\mathbf{q})$, Hamilton’s canonical equations for the phase space variables (\mathbf{q}, \mathbf{p}) in this example are

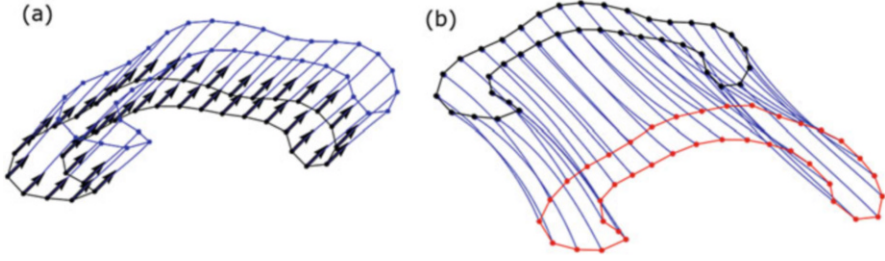


Fig. 2 (a) Human corpus callosum shape represented by landmarks (black curve and points) and geodesic flow (blue curves) specified by an initial vector field (vectors). (b) Geodesic matching between two corpus callosum shapes (black and red). Compare these deterministic evolutions to the stochastic trajectories shown later in the chapter

$$\begin{aligned}\frac{d}{dt}\mathbf{q} &= \frac{\partial}{\partial \mathbf{p}} H(\mathbf{q}, \mathbf{p}) = \frac{\mathbf{p}}{m}, \\ \frac{d}{dt}\mathbf{p} &= -\frac{\partial}{\partial \mathbf{q}} H(\mathbf{q}, \mathbf{p}) = -\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}}.\end{aligned}\tag{6}$$

In the LDDMM setting, the Hamiltonians define the kinetic energy for a Riemannian metric. They are in quadratic form, and the potential energy is absent. The dynamics of this class of Hamiltonians describes geodesic motion, since no force derived from a potential is present. This is the case for landmark dynamics, where the Hamiltonian is given by

$$H_0(\mathbf{q}, \mathbf{p}) = \sum_{i,j=1}^n p_i^T K(q_i, q_j) p_j,\tag{7}$$

when evaluated on a covector $(\mathbf{q}, \mathbf{p}) \in T_q^*S$. The co-metric K in (3) replaces the mass m of the particle. However, it depends nonlinearly on the landmark configuration \mathbf{q} . Thus, Hamilton's equations for the Hamiltonian (7) involve terms for both the position and momentum equation:

$$\begin{aligned}\frac{d}{dt}q_i &= \frac{\partial}{\partial p_i} H_0(\mathbf{q}, \mathbf{p}) = \sum_{j=1}^n K(q_i, q_j) p_j, \\ \frac{d}{dt}p_i &= -\frac{\partial}{\partial q_i} H_0(\mathbf{q}, \mathbf{p}) = -\sum_{j=1}^n p_i^T \partial_{q_i} K(q_i, q_j) p_j.\end{aligned}\tag{8}$$

In practice, one often selects a Gaussian kernel of the form

$$K(q_i, q_j) = K(\|q_i - q_j\|) = \exp\left(-\frac{\|q_i - q_j\|^2}{2\sigma^2}\right), \quad \forall q_i, q_j \in \Omega, \quad (9)$$

with standard deviation σ .

Noise from a Statistical Physics Perspective

In statistical physics, for a given Hamiltonian, noise can be introduced in a natural way with the canonical ensemble, or heat bath at fixed temperature, and with conservation of mass. This system is fully described by the partition function

$$Z = \int e^{-\beta H_0(\mathbf{q}, \mathbf{p})} d\mathbf{q} d\mathbf{p}, \quad (10)$$

where $\beta = 1/T$ is the inverse temperature. To better understand what this system represents, one may consider the following stochastic differential equation:

$$\begin{aligned} \frac{d}{dt} \mathbf{q} &= \frac{\partial}{\partial \mathbf{p}} H_0(\mathbf{q}, \mathbf{p}) \\ d\mathbf{p} &= -\frac{\partial}{\partial \mathbf{q}} H_0(\mathbf{q}, \mathbf{p}) dt - \sigma d\mathbf{W} - \theta \frac{\partial}{\partial \mathbf{p}} H_0(\mathbf{q}, \mathbf{p}) dt, \end{aligned} \quad (11)$$

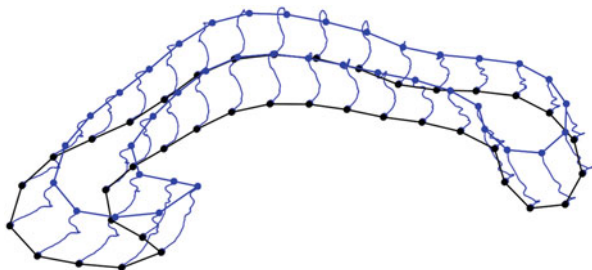
where $d\mathbf{W}$ is the n -dimensional Wiener increment and $\sigma, \theta \in \mathbb{R}$. In this case, the invariant measure of the stochastic dynamics is the well-known Gibbs distribution

$$: \mathbb{P}_\infty(\mathbf{q}, \mathbf{p}) = \frac{1}{Z} e^{-\beta H_0(\mathbf{q}, \mathbf{p})}, \quad (12)$$

where the inverse temperature β is defined by the so-called Einstein relation $\frac{\theta}{2\sigma^2} = \beta$. This relation characterises the balance between the noise and the damping parametrised by, respectively, σ and θ . In the heat bath analogy, the noise represents localised random perturbations, and the damping accounts for friction between the motion and the ambient space.

We refer to Marsland and Shardlow (2017) for more details on this approach for landmark dynamics and some related Bayesian inference problems. Figure 3 illustrates the corresponding dynamics. Although this is interesting from a stationary state perspective, the boundary value problem of finding geodesics between two fixed shapes does not really fit this point of view. In fact, it can be seen as the opposite, where the notion of initial and final conditions is lost at stationarity. The dissipation term is therefore not relevant, as for short times, away from equilibrium, its effect becomes negligible. As we will see in the next section, the dissipation terms break the original Hamiltonian structure, while the noise does not; thus, only the noise can be considered in the original geometrical framework of shape analysis.

Fig. 3 Example evolution of the Langevin equation (11) introduced by Marsland and Shardlow (2017). Initial conditions as in Fig. 2



Non-dissipative Stochastic Shape Models

Let us rewrite the SDE (11) in matrix form by using the function $H_1(\mathbf{p}, \mathbf{q}) = \sigma \cdot \mathbf{p}$, as

$$d\mathbf{x} = \mathcal{J}\nabla_{\mathbf{x}}H_0(\mathbf{x})dt + \sigma\mathcal{J}\nabla_{\mathbf{x}}H_1(\mathbf{x})dW + \theta\mathcal{K}\nabla_{\mathbf{x}}H_0(\mathbf{x})dt, \tag{13}$$

where we use the notation $\mathbf{x} = (\mathbf{q}, \mathbf{p})$ for which ∇ is the corresponding derivative. We also defined two matrices $\mathcal{J} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $\mathcal{K} = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$, one anti-symmetric called a Hamiltonian structure and the other symmetric. In the deterministic case (with $\sigma = 0$) and without dissipation ($\theta = 0$), the anti-symmetry of \mathcal{J} provides conservation of energy H_0 , while here we would only have conservation of ‘stochastically perturbed energy’ $H_0dt + \sigma H_1dW$; see, for example, Holm (2015) and Arnaudon et al. (2018a) for more on that.

Equation (13) is often called a Langevin equation, and it has particularly interesting regimes, such as the overdamped situation. For our previous example of a particle with mass m , that is, with $H_0 = \frac{p^2}{2m} + U(q)$ and $H_1 = -q$, the overdamping regime is equivalent to the limit $m \rightarrow 0$, and Eq. (13) directly reduces to an equation for the velocity $\dot{\mathbf{q}}$ only, of the form

$$\theta\dot{\mathbf{q}} = \sigma\dot{\mathbf{W}} - \nabla_{\mathbf{q}}U(\mathbf{q}). \tag{14}$$

This equation represents a particle undergoing Brownian motion in the potential U . Without the potential U , standard Brownian motion is recovered.

We will now study in more detail three different cases of including noise in shape analysis, based on some generalisations of these equations but with neither damping nor potential. First, in section “[Riemannian Brownian Motion](#)” we will consider Brownian motion but on the shape space. Then we return to the particle analogy in section “[Lagrangian Noise](#)” but with the landmark Hamiltonian and finally extend the additive noise to an Eulerian noise in section “[Eulerian Noise](#)”.

Riemannian Brownian Motion

We have so far discussed the noise from a Hamiltonian perspective with the Hamiltonian being the kinetic energy coming from a Riemannian metric on the landmark space. We now use the Riemannian metric on \mathcal{S} directly to define infinitesimal stochastic perturbations which are identically distributed and have isotropic variance, thereby generating the Riemannian Brownian motion. In infinite-dimensional models, noise with equal variance in all dimensions has infinite magnitude. Hence, Brownian motion in its direct form is defined only on finite dimensional manifolds. In this case, the resulting process is well defined up to a possible explosion time.

Let the shape space \mathcal{S} be a finite-dimensional Riemannian manifold with dimension m and let g denote the Riemannian metric. The Laplace-Beltrami operator Δ on \mathcal{S} is given by $\Delta f = \nabla \cdot \nabla f$ where the divergence is defined as $\nabla \cdot X = \frac{1}{\sqrt{g}} \frac{\partial(\sqrt{g}a^i)}{\partial q^i}$ applied to a vector field $X = a^i \frac{\partial}{\partial q^i}$ and ∇f is the Riemannian gradient of the function $f : \mathcal{S} \rightarrow \mathbb{R}$. Riemannian Brownian motion is a diffusion process on \mathcal{S} with generator $\Delta/2$. There are various characterisations of the Brownian motion and various ways to construct the process; see, e.g. Emery (1989) and Hsu (2002). If $\mathbf{Q}(t)$ is a Brownian motion, its density at time $t > 0$ with respect to the Riemannian volume form satisfies the heat equation:

$$\partial_t p(t, \mathbf{q}) = \frac{1}{2} \Delta p(t, \mathbf{q}), \quad \mathbf{q} \in \mathcal{S}. \tag{15}$$

Therefore, also on a Riemannian manifold, the heat flow is inherently connected to a generalisation of the Brownian motion (14) with coordinate expression:

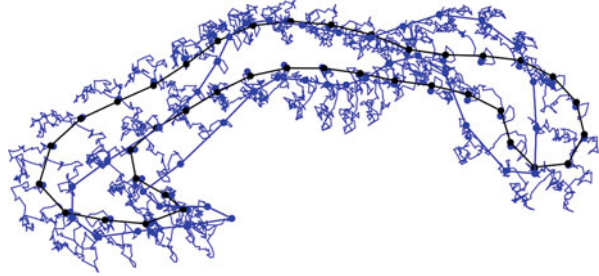
$$d\mathbf{Q}(t)^i = -\frac{1}{2} g(\mathbf{Q}(t))^{kl} \Gamma(\mathbf{Q}(t))_{kl}^i dt + \sqrt{g(\mathbf{Q}(t))^{-1}}^i dW(t), \tag{16}$$

where W is a m -dimensional (Euclidean) Wiener process and the diffusion field $\sqrt{g(\mathbf{Q}(t))^{-1}}$ is a square root of the co-metric tensor $g(\mathbf{Q}(t))^{ij}$. The drift term is a contraction between the metric and the Christoffel symbols Γ_{kl}^i . We now turn to the landmark space, which, as we have seen, is a Riemannian manifold with $m = nd$. Since the co-metric is the kernel K , we obtain in this case the coordinate expression

$$d\mathbf{Q}(t) = -\frac{1}{2} K(\mathbf{Q}_{t_k}, \mathbf{Q}_{t_k})^{kl} \Gamma(\mathbf{Q}(t))_{kl} dt + \sqrt{K(\mathbf{Q}_{t_k}, \mathbf{Q}_{t_k})} dW(t). \tag{17}$$

Figure 4 shows a sample path from the landmark Riemannian Brownian motion. This process has been used in Staneva and Younes (2017) for analysis of stochastic landmark trajectories with continuous observations and in Sommer et al. (2017) with a Brownian bridge simulation to perform statistical estimation on landmark spaces with discrete time observations.

Fig. 4 Example evolution of the landmark Brownian motion (17). Initial shape as in Fig. 2 (no initial velocity)



Contrary to the Euclidean case, a Riemannian Brownian motion is not guaranteed to exist for infinite time. It can explode, meaning that with positive probability, it will leave any compact set in finite time. Sufficient conditions for non-explosion includes compactness of \mathcal{S} or the Ricci curvature being bounded from below. Interestingly, for the landmark manifold which is not compact, it is currently not known whether finite-time explosion can occur. Finite-time explosion would imply that either the landmarks escape to infinity in \mathbb{R}^d in finite time or that two or more landmarks collide in finite time. The investigation of these properties is currently an active area of research.

Lagrangian Noise

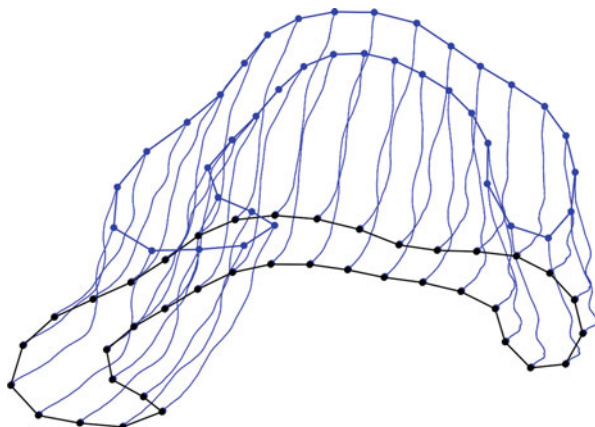
Upon using this formulation without dissipation ($\theta = 0$) for n landmarks, with the Hamiltonian H_0 in (7) and the generalisation of H_1 in n ‘stochastic Hamiltonians’ $H_i = \sigma_i \cdot q_i$, we arrive at the model of Trounev and Vialard (2012) and Vialard (2013), written explicitly as

$$dq_i = \sum_{j=1}^n K(q_i, q_j) p_j dt \tag{18}$$

$$dp_i = - \sum_{j=1}^n p_i^T \partial_{q_i} K(q_i, q_j) p_j dt - \sigma_i dW_i,$$

where we considered a different Wiener process W_i for each H_i . See Fig. 5 for an illustration of this noise. This noise has a Lagrangian flavour to it, because each noise W_i is associated with a single landmark (q_i, p_i) . This Lagrangian formulation of stochastic landmark dynamics has ‘smooth’ trajectories in space, as the noise only appears in the momentum equation. Thus, the paths have the regularity of the integral of the Wiener processes W_i . In addition, landmarks can cross each other under the influence of the noise, which violates one of the properties of the deterministic equations. See Holm and Tyranowski (2016) for a numerical study of these landmark crossings. Finally, it is interesting to note that in the limit of infinitely

Fig. 5 Example evolution of the Lagrangian noise (18) and introduced by Trounev and Vialard (2012) and Vialard (2013). Initial conditions as in Fig. 2



many particles, that is, for infinite dimensional shapes, this noise persists under the form of cylindrical Brownian motion. See Vialard (2013) for more detail.

Eulerian Noise

Within the same Hamiltonian formulation, it is possible to design another type of noise, which can be interpreted as an Eulerian noise, where each Wiener process is not associated with each landmark anymore but to a different field on the image space Ω ; see Arnaudon et al. (2019a). To do so, we select k ‘stochastic Hamiltonians’ H_l of the form:

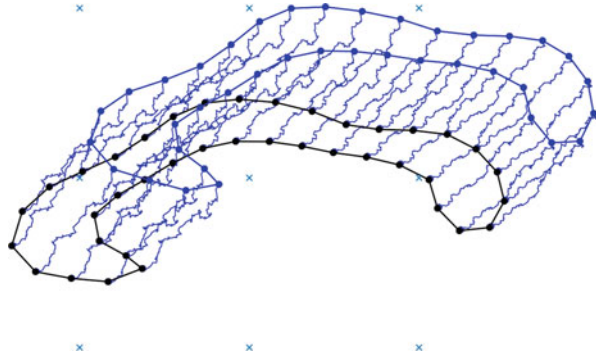
$$H_l(q_i, p_i) = \sigma_l(q_i) \cdot p_i, \tag{19}$$

where the functions σ_l are the fields on Ω , and we can have a number of Wiener processes different from the number of landmarks. With (19), the corresponding stochastic Hamiltonian equation is more complex as it is multiplicative in both equations. It explicitly reads

$$\begin{aligned} \frac{d}{dt}q_i &= \sum_{j=1}^n K(q_i, q_j)p_j dt + \sum_l \sigma_l(q_i) \circ dW(t)^l \\ \frac{d}{dt}p_i &= - \sum_{j=1}^n p_i^T \partial_{q_i} K(q_i, q_j)p_j dt - \sum_l \frac{\partial}{\partial q_i} (p_i \cdot \sigma_l(q_i)) \circ dW(t)^l, \end{aligned} \tag{20}$$

where we use Stratonovich integrals, denoted with \circ , to simplify notation by absorbing the Itô correction term. Because of the choice of functions $\sigma_l(\mathbf{q})$ instead of a coefficient for the Lagrangian noise, this model allows more control of the location, direction and amplitude of the noise throughout the domain Ω . In

Fig. 6 Example evolution of the Eulerian noise (20) introduced in Arnaudon et al. (2017, 2019a). Noise field positions indicated by crosses. Initial conditions as in Fig. 2. Notice that the paths are more noisy due to noise in the position equation, but the large spatial correlation for the noise preserves more the shape than the other noises presented here



particular, the spatial correlation of the noise can be fully characterised by the set of functions σ_i , independently of the number of landmarks. As we will see later, this property is crucial for any inverse problem of learning the noise or estimating the uncertainty of a matching problem from data. Notice that if the noise fields σ_i are taken constant, this model simplifies to landmarks with exactly the same constant additive noise in the position equation, corresponding to a global random displacement of the domain, but not to the Lagrangian model. The noise fields and a sample from the stochastic evolution are visualised in Fig. 6.

These two models are therefore different in nature and by their specific properties may be used for different applications. For example, this model will result in noisy trajectories of landmarks, which requires more care in the numerical integration with Stratonovich noise and some hypotheses about the form of the noise fields σ_i . At the contrary, the Lagrangian noise is simpler to integrate and, in its simplest form, may only require a single constant to parametrise the noise amplitude of all landmarks.

Stochastic Euler-Poincaré Reduction and Its Infinite Dimensional Extension

As mentioned in the previous section, only the Eulerian model for stochastic landmark dynamics can be generalised to other shape spaces. This is because this form of the stochastic Hamiltonian is compatible with reduction by symmetry, which provides the tools to compute the geodesic equations for general shape spaces, and in particular landmarks, as in (8). Here, we outline Euler-Poincaré reduction for diffeomorphisms (Holm and Marsden 2003) leading to the EPDiff equations which provide the basis for the stochastic Euler-Poincaré theory used in section “[Stochastic Euler-Poincaré Reduction and Its Infinite Dimensional Extension](#)”. The main point is that because the Riemannian metric defined above is invariant under right translations, the geodesic equations can be solved on $V \subset \mathfrak{X}(\Omega)$ and then subsequently reconstructed to give a path on $\phi(t)$, by using the flow equation (2). This is an example reduction by symmetry.

Reduction by Symmetry

From Eq. (1), without the dissimilarity term, the matching problem corresponds to the variational principle:

$$\delta E(\phi(t)) = \delta \frac{1}{2} \int_0^1 \ell(v(t)) dt = 0, \tag{21}$$

using the Lagrangian ℓ and with variations $\delta\phi(t)$ of the curve $\phi(t)$ that vanish at the endpoints ϕ_0 and ϕ_1 .

Regarding $\text{Diff}(\Omega)$ as a group, we let R_ϕ be the right translation $R_\phi(\psi) = \psi \circ \phi$. The derivative with respect to ψ is the pushforward $(R_\phi)_*$. Particularly, $(R_{\phi^{-1}})_*$ is a mapping from the tangent space $T_\phi\text{Diff}(\Omega)$ to $T_{\text{id}}\text{Diff}(\Omega)$. The latter may be identified with the Lie algebra of smooth vector fields, $\mathfrak{X}(\Omega)$. Let $w(t) := (R_{\phi(t)^{-1}})_*\delta\phi(t)$ be the right translations of the variation $\delta\phi(t)$ for each t . The variation $\delta v(t)$ of $v(t)$ is related to $w(t)$ by the equality

$$\delta v(t) - \frac{d}{dt}w(t) = [v, w] = -\text{ad}_v w, \tag{22}$$

in terms of the Lie algebra adjoint map ad , where the bracket $[v, w]$ is the Jacobi–Lie bracket between vector fields.

Let $\delta\ell/\delta v$ denote the variational derivative of the Lagrangian ℓ with respect to v , and let $(\frac{\delta\ell}{\delta v}|\delta v) := \delta\ell(v)$ denote the pairing of this with a variation of v . Since $\delta E(\phi)$ vanishes for all such variations from (21), one obtains

$$\frac{d}{dt} \frac{\delta l}{\delta v} + \text{ad}_v^* \frac{\delta l}{\delta v} = 0. \tag{23}$$

These are called the Euler–Poincaré equations when derived for general Lie groups. They are called the EPDiff equations in the present case, when the group is $\text{Diff}(\Omega)$. Here, the Lagrangian only contains a kinetic energy and reads

$$l(v) = \int_\Omega v \cdot Lv dx, \tag{24}$$

where $L = K^{-1}$ is the operator associated with the kernel K discussed earlier. For vector fields, the commutator is given by $[u, v] = u\nabla_x v - v\nabla_x u$. Thus, in terms of the momentum variable $m = Lv$, the EPdiff equation reads:

$$\dot{m} = (u \cdot \nabla_x)m + m(\nabla_x u)^T + (\nabla_x \cdot u)m. \tag{25}$$

This equation is also a Hamiltonian equation, but with a noncanonical Lie–Poisson bracket structure, given by the ad^* operation,

$$\dot{m} = -\text{ad}_{\frac{\delta h}{\delta m}}^* m, \quad (26)$$

in which the reduced Hamiltonian is simply

$$h(v) = \int_{\Omega} m \cdot K m \, dx. \quad (27)$$

Stochastic EPDiff

From this Hamiltonian formulation, it is straightforward to implement the Eulerian noise of section “[Eulerian Noise](#)” whose stochastic Hamiltonians is linear in the momenta:

$$h_l(m) = \int_{\Omega} \sigma_l(x) \cdot m(x) \, dx, \quad (28)$$

where σ_l are, as before, fields on the image domain Ω . The corresponding stochastic EPDiff is then the stochastic differential equation:

$$dm = \text{ad}_{\frac{\delta h}{\delta m}}^* m \, dt + \sum_l \text{ad}_{\frac{\delta h_l}{\delta m}}^* m \circ dW_l. \quad (29)$$

More explicitly, this is

$$\begin{aligned} dm = & \left((u \cdot \nabla_x) m + m (\nabla_x u)^T + (\nabla_x \cdot u) m \right) \\ & + \sum_l \left((\sigma_l \cdot \nabla_x) m + m \cdot (\nabla \sigma)^T + (\nabla \cdot \sigma_l) m \right) \circ dW_l. \end{aligned} \quad (30)$$

In order to see that this equation is a generalisation of the Eulerian model of stochastic landmark dynamics, one simply considers singular solutions of the form:

$$m(x, t) = \sum_i p_i(t) \delta(x - q_i(t)). \quad (31)$$

Once substituted in (30), this singular representation yields the stochastic landmark equations (20). We refer the interested reader to Arnaudon et al. (2018b) and Kühnel et al. (2018) for more detailed treatments of the stochastic EPDiff equation in the context of shape analysis.

Other Stochastic Shape Models

Stochastic extensions of landmark and image dynamics are also considered in the context of Brownian flows in the sense of Kunita Kunita (1997). Here, infinite dimensional stochastic noise is added to the flow equation (2) resulting in the SDE:

$$d\phi(t)(x) = v(\phi(t)(x), t)dt + \sum_{i=1}^{\infty} f_i(\phi(t)(x), t)dW^i(t). \quad (32)$$

This SDE is on the diffeomorphism side: $\phi(t)$ is the evolving diffeomorphism in $\text{Diff}(\Omega)$, and $v(t)$ a t -dependent family of vector fields in $V \subset \mathfrak{X}(\Omega)$. $W(t)^i$, $i \in \mathbb{N}$ is an infinite sequence of Wiener processes, and $f_i : \Omega \times [0, 1] \rightarrow \mathbb{R}^d$ a sequence of suitable vector fields. The added stochastic terms distinguish the SDE from the deterministic equivalent (2). Through the action on shapes, (32) gives a corresponding stochastic shape evolution $\phi(t) \cdot \mathbf{q}$ for a fixed shape \mathbf{q} . The resulting process is a.s. nowhere differentiable, and the energy (1) is therefore infinite if evaluated on ϕ directly. However, Markussen (2004, 2007) establishes a renormalisation procedure defining the energy on finite time partitions and showing that in the limit with infinitely fine partitions, a maximum a posteriori flow exists and that it coincides with solutions to the LDDMM variational problem. It thus gives a probabilistic interpretation of the LDDMM energy.

In Budhiraja et al. (2010), a large-deviation principle is established for flows of the type (32), and it is shown how the LDDMM variational problem appears in the small noise limit, thereby giving a different probabilistic characterisation of the LDDMM energy.

In Wassermann et al. (2014), a locally linear approximation of this SDE gives a Gaussian-process approximation to the stochastic diffeomorphism flow and produces differential systems for the evolution of the mean flow and its pointwise covariance. This in turn allows uncertainty quantification in image matching.

Stochastic Models in Shape Statistics

The stochastic shape models described in the previous section appear in applications when estimating the noise structure along a shape trajectory observed at multiple time points, along ensembles of observed shape trajectories, and for modelling probability distributions on the nonlinear shape space. Here, we describe examples of such applications.

Random Orbits with Time-Continuous Noise

As described in section “Metric and Variational Formulations”, a shape trajectory $t \mapsto \mathbf{q}(t)$ that is extremal for a variational principle is described by its initial

condition $(\mathbf{q}(0), \mathbf{p}(0))$. If we fix a reference shape $\bar{\mathbf{q}}$ (often called a template), one can use this to parametrise a dataset $\mathbf{q}^1, \dots, \mathbf{q}^N$ of N shapes by solving the matching problem (1) for each \mathbf{q}^i starting at $\bar{\mathbf{q}}$ and letting \mathbf{p}^i be the initial momentum parametrising the flow bringing $\bar{\mathbf{q}}$ into correspondence with \mathbf{q}^i . Let \mathbf{v}^i be the corresponding initial velocities corresponding to \mathbf{p}^i .

This represents the dataset in the linear space $T_{\bar{\mathbf{q}}}\mathcal{S}$. The velocities \mathbf{v}^i can also be lifted to $V \subset \mathcal{X}(\Omega)$. In both cases, the data is mapped from the nonlinear shape space to a vector space, and statistical analysis can be performed in the vector space using techniques for analysis of multivariate data. The data can, for example, be visualised by applying PCA to $\mathbf{v}^1, \dots, \mathbf{v}^N$ and plotting the first components of the data.

Because the velocities \mathbf{v}^i parametrise $\text{Diff}(\Omega)$ geodesics ϕ^i that act on $\bar{\mathbf{q}}$ to produce shapes close to \mathbf{q}^i , this model is often denoted the *random orbit model* (Miller et al. 1997). Considering generative models in the random orbit sense, a stochastic variable on V generates a stochastic variable on $\text{Diff}(\Omega)$ that through the action gives a stochastic variable on \mathcal{S} . Thus, the randomness appears in the initial condition of the flow that generate \mathbf{q}^i through $\phi^i(1)$.

The stochastic models described in the previous sections allow one to complement the initial randomness in V with time-continuous randomness. An important example is to model the time evolution of organ shapes in a population of healthy and diseased patients, where the variation between healthy and diseased is placed in the initial velocity, while the organ shape evolution for each subject is allowed to exhibit stochastic variation throughout time. This model can potentially include the stochasticity that is not related to the disease in the time-continuous noise allowing a cleaner disease vs. healthy signal in the initial velocity.

Noise Inference from Evolution of Moments

From the Eulerian stochastic model, the noise fields σ_l are to be determined to obtain relevant stochastic dynamics. One possibility is to infer them from some distributions of shapes, assumed to be samples from a single choice of the noise fields. To solve this inverse problem, one first simplifies the search space by parametrising a tractable number of noise fields, with, for example, Gaussian kernels, and tries to infer these parameters.

In Arnaudon et al. (2019a), the evolution of the first moments of the probability distributions of landmarks on position and momenta was first computed, and the mean and variance of the moments associated with positions were used to match the observed distributions of shapes. This algorithm requires one to derive and solve a set of couple ordinary equations approximating the Fokker-Planck equation for the probability distributions and implement a shooting algorithm on the initial momenta for the mean, variance and the noise parameters. As demonstrated in Arnaudon et al. (2019a), this method gives accurate results when landmarks do not interact much on noisy regions of the image or, equivalently, when the moment approximation is most accurate.

The same method has been applied on the EPDiff equation (30), after the application of an appropriate spatial discretisation in low-frequency Fourier modes (Kühnel et al. 2018), following Zhang and Fletcher (2015). This algorithm works because of the control one has on the spatial correlation of the noise. Indeed, by choosing noise fields as Fourier modes, discarding the highest frequency components of the dynamics to reduce the dimensionality of the problem only restricts the number of noise fields that can be inferred.

Likelihood-Based Inference and Bridge Sampling

Generally, the stochastic models, potentially coupled with randomness in the initial conditions as introduced in the random orbit model, result in time-evolving probability distribution p_t . If data are assumed to be observed at a fixed time T , e.g. $T = 1$, and the observations independent and identically distributed, one can define the likelihood of the model by

$$\mathcal{L}(\theta; \mathbf{q}^1, \dots, \mathbf{q}^N) = \prod_{i=1}^N p_T(\mathbf{q}^i). \quad (33)$$

Here θ contains parameters of the model, including the starting configuration $\bar{\mathbf{q}}$ of the process, the noise fields and width of the kernel K . These parameters can then be estimated by maximising the likelihood, i.e. searching for $\arg \max_{\theta} \mathcal{L}(\theta; \mathbf{q}^1, \dots, \mathbf{q}^N)$. Alternatively, with a Bayesian view, a prior on θ allows to sample from the posterior distribution of θ given $\mathbf{q}^1, \dots, \mathbf{q}^N$. This approach gives a general way to estimate parameters and compare models (Sommer 2020).

The transition density p_t used in the likelihood (33) is a solution of the Fokker-Planck equation; however, it is generally intractable to forward simulate the resulting PDE on high-dimensional (or infinite dimensional) shape spaces. Instead, numerical approximation can be approached with bridge sampling as pursued in Arnaudon et al. (2017, 2019a, 2020) and Sommer et al. (2017); see also Sommer (2020). We briefly outline the approach below, assuming the shape manifold can be represented in Euclidean coordinates such as for the landmark manifold.

Let $\mathbf{Q}(t)$ be the solution of an Itô SDE:

$$d\mathbf{Q}(t) = b(\mathbf{Q}(t), t)dt + \sigma(\mathbf{Q}(t), t)dW(t), \quad (34)$$

where $W(t)$ is a Euclidean Brownian motion, e.g. (17). We are now interested in conditioning $\mathbf{Q}(t)$ on hitting a point \mathbf{v} at time T , thinking of \mathbf{v} as a data point that could be a sample from $\mathbf{Q}(T)$. The conditioned process $\mathbf{Q}^* = \mathbf{Q}|\mathbf{Q}(T) = \mathbf{v}$ has the SDE

$$d\mathbf{Q}^*(t) = b(\mathbf{Q}^*(t), t)dt + \sigma(t, \mathbf{Q}^*(t))\sigma(t, \mathbf{Q}^*(t))^T \nabla \log p_{T-t}(\mathbf{v}; \mathbf{Q}^*(t)) + \sigma(\mathbf{Q}^*(t), t)dW(t), \quad (35)$$

where $p_{T-t}(\mathbf{v}; \mathbf{Q}^*(t))$ is the transition density of the process started at $\mathbf{Q}^*(t)$, running for time $T - t$ and evaluated at \mathbf{v} . However, this SDE can generally not be used for numerical simulations since the gradient of the log-transition density $\nabla \log p_{T-t}(\mathbf{v}; \mathbf{Q}^*(t))$ in the added drift term is generally not available. To circumvent this, Delyon and Hu (2006) introduced the idea of guided proposals approximating the bridge SDE (35) by

$$d\mathbf{Y}(t) = b(\mathbf{Y}(t), t)dt - \frac{\mathbf{Y}(t) - \mathbf{v}}{T - t}dt + \sigma(\mathbf{Y}(t), t)dW(t). \quad (36)$$

Under condition of invertibility of σ and boundedness of the b , σ and σ^{-1} , the process \mathbf{Y} will hit \mathbf{v} a.s. at time T . While the process has a different law than \mathbf{Q}^* , the likelihood ratio can be computed giving the relation

$$\mathbb{E}_{\mathbf{Q}|\mathbf{Q}(T)=\mathbf{v}}[f(\mathbf{Q}(t))] = \frac{\mathbb{E}_{\mathbf{Y}}[f(\mathbf{Y}(t))\varphi(\mathbf{Y}(t))]}{\mathbb{E}_{\mathbf{Y}}[\varphi(\mathbf{Y}(t))]}, \quad (37)$$

for a function φ that can be computed numerically. Furthermore, the transition density is written in terms of φ by

$$p_T(\mathbf{v}; \mathbf{Q}(0)) = \sqrt{\frac{|A(T, \mathbf{v})|}{(2\pi T)^d}} \exp\left(-\frac{\|a(0, \mathbf{Q}(0))^{-1}(\mathbf{Q}(0) - \mathbf{v})\|^2}{2T}\right) \mathbb{E}_{\mathbf{Y}}[\varphi(\mathbf{Y}(t))]. \quad (38)$$

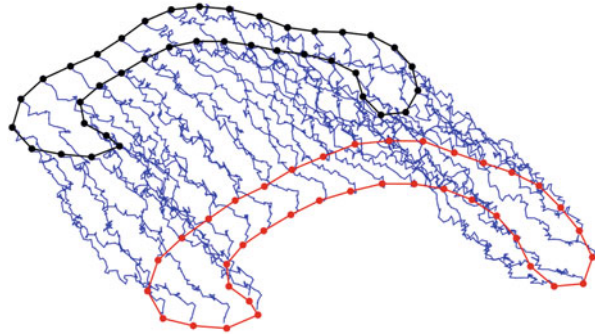
Similar scheme that gives improved approximations of the bridge has subsequently been introduced; see, e.g. Schauer et al. (2017).

In the landmark case, this Euclidean approach to estimating the density p_T can be used because of the vector space representation of $\mathbf{q} \in \mathbb{R}^{nd}$. This has been pursued for the Riemannian Brownian motion in Sommer et al. (2017), for the stochastic EPDiff model in Arnaudon et al. (2017, 2019a), and, recently, using the approach of Schauer et al. (2017), in Arnaudon et al. (2020) that also covers the Lagrangian models. See Fig. 7 for an example of a bridge sample between two corpus callosum shapes using the process (36).

Likelihood Maximisation and Automatic Differentiation

Bridge sampling allows to approximate the likelihood (33) using (38). It remains to maximise $\mathcal{L}(\theta; \mathbf{q}^1, \dots, \mathbf{q}^N)$ with respect to the parameters θ . In Sommer et al. (2017), a simple stochastic gradient optimisation scheme is used for maximising the likelihood on the landmark manifold with the Riemannian Brownian motion. The parameters are here the starting point $\mathbf{q}(0)$ of the process and parameters of the

Fig. 7 Sample from the guided bridge process (36). Initial and final shapes from Fig. 2



kernel matrix determining the metric of the manifold. Such gradient-based schemes need approximations of the gradient $\nabla_{\theta_l} \mathcal{L}(\theta_l; \mathbf{q}^1, \dots, \mathbf{q}^N)$. In the deterministic setting, the gradient with respect to the initial conditions of the energy (1) can be computed using the adjoint equations of the Hamiltonian system. It is often not in practice feasible to derive similar systems for the gradient of (33) or (38). Instead, modern automatic differentiation can be used to compute gradients of the entire numerical simulation scheme used for the stochastic integration of (\mathbf{q}, \mathbf{p}) and φ .

This idea of using automatic differentiation of stochastic geometric systems was first pursued in Arnaudon et al. (2017) using the framework Theano Geometry (Kühnel et al. 2019) (<http://bitbucket.org/stefansommer/theanogeometry>). Recently, it has been extended to general stochastic Hamiltonian system including the ones discussed in this chapter using the automatic differentiation features of Julia (Arnaudon et al. 2020). The use of automatic differentiation for shape and general geometric computations has furthermore been treated in Kühnel and Sommer (2017) and Kühnel et al. (2019) using the Theano framework, in the Geomstats library (<http://geomstats.ai>), in KeOps (<https://www.kernel-operations.io/>) and Deformetrica (<http://www.deformetrica.org/>).

Applications and Extensions

In addition to the presented models of stochastic shape analysis, we briefly describe related extensions and data analysis applications.

In Arnaudon et al. (2018b), the stochastic EPDiff model (30) was applied to images and landmarks in the context of string methods (Weinan et al. 2005; Vandenberg and Venturoli 2009). The string method defines a gradient flow to minimise the energy of the path between two shapes, while the updated string is perturbed by noise. As such, it can be seen as an alternative way to retrieve stochastic paths between shapes in comparison with the bridge sampling methods described above.

The paper additionally links the momentum map representation of images (Bruveris et al. 2009) with the stochastic EPDiff model. The stochastic EPDiff models have been used for medical imaging and computational anatomy in Arnaudon et al. (2017), for example, for modelling variations in the human corpus callosum.

The metamorphosis framework (Trounev and Vialard 2012) combines variations in shapes arising from deformations with variations in the data itself, e.g. pixel intensity variation in images. The stochastic EPDiff models have been extended to include metamorphosis in Holm (2017) and Arnaudon et al. (2019b).

In Holm (2020), stochasticity in the Lagrangian and Eulerian reference frames is coupled via a momentum map. This results in a multi-scale flow with two interpenetrating degrees of freedom coupled by two different forms of stochasticity. This interpenetration approach allows perturbations which are not simply attached to the flow. Instead, they can propagate relative to the flow. The model has been used as a framework for investigating wave-current interaction in the dynamics of ocean-atmosphere coupling, and we expect it to also be relevant in the context of stochastic shape analysis.

Finally, one needs to mention rough path theory, or rough flow theory, a powerful method of dealing with the dynamics of highly oscillatory nonlinear systems. Rough flow theory transcends Itô and Stratonovich stochastic calculus, by providing an almost sure pathwise definition of the solution of a stochastic partial differential equation. The landmark trajectories in the stochastic framework are described by stochastic integrals which do not have a pathwise interpretation. The rough flow treatment restores this property. Moreover, a rough flow solution is well posed in the sense of convergence to a sequence of smooth flows in the p -variation metric, as described, e.g. in Friz and Victoir (2010). In contrast, solutions in Itô and Stratonovich stochastic calculus converge only weakly; namely, they converge in the sense of the L^2 norm. Thus, rough flow theory transcends Itô and Stratonovich stochastic calculus on semimartingale flows by admitting partial differential equations driven by non-semimartingale flows, such as Gaussian processes and Markov processes defined on Banach spaces which are neither differentiable nor of bounded variation (Friz and Victoir 2010). Moreover, rough flows comprise a natural basis for functions on data streams that can be used for machine learning (Lyons 2014).

By using the theory of controlled rough paths (Gubinelli 2004), one may derive a class of rough EPDiff equations for shape analysis as critical points of a rough action functional. The rough variational approach to EPDiff considerably enhances the stochastic variational approach. For example, the rough flow driven variational approach admits non-Markovian perturbations. Memory effects can also be introduced into this approach through a judicious choice of the driving rough flows. In particular, one may choose these models to characterise landmark trajectories in shape analysis as time-dependent geometric rough paths (GRP) on the manifold of diffeomorphic maps. For a parallel derivation of Euler–Poincaré equations on GRP for applications in fluid dynamics, see Crisan et al. (2020).

Conclusion and Outlook

We have surveyed stochastic shape models from a Hamiltonian viewpoint. In the deterministic outer metric LDDMM setup of the diffeomorphism group acting on shape spaces, we have shown that perturbations of either Hamilton's equations or the Hamiltonian can lead directly to stochastic shape models. The analysis lifts Lie group symmetry reduction from the deterministic EPDiff model to its stochastic EPDiff counterpart.

We have discussed several important applications of the stochastic models in shape statistics and described extensions beyond the standard methods to include metamorphosis, string sampling, the relation to fluid dynamics and the coupling of noise in different frames of references.

Shape modelling has been a very active and successful research area from both the theoretical and applied viewpoints, with important applications in biology and medical imaging. Stochastic shape modelling is currently an actively investigated area which we expect will evolve with many fascinating new developments in the coming years.

To close this chapter, we shall not ask, 'What is next for stochastic shape modelling?' In fact, we expect that applications of stochastic shape modelling will certainly continue to branch into many directions and follow its successful destiny. Rather, let us ask, 'What lies beyond stochastic shape modelling?' For this, let us speculate that the successes made recently in stochastic shape modelling might be transferable into the domain of shape analysis with rough flows. We imagine that the fundamental recent developments in rough flow theory and its clear relevance to machine learning (Lyons 2014) could offer an attractive new mathematical framework for shape modelling.

Acknowledgments Research is never done in a vacuum. We are enormously grateful to our friends in the shape analysis community for their remarkable tradition of openness, inclusiveness and kind encouragement to each other in their many joint endeavours. The work is supported by the Villum Foundation grant 00022924 and the Novo Nordisk Foundation grant NNF18OC0052000.

References

- Arnaudon, A., Holm, D.D., Pai, A., Sommer, S.: A Stochastic large deformation model for computational anatomy. In: Information Processing in Medical Imaging. Lecture Notes in Computer Science, pp. 571–582. Springer (2017). https://doi.org/10.1007/978-3-319-59050-9_45
- Arnaudon, A., De Castro, A.L., Holm, D.D.: Noise and dissipation on coadjoint orbits. *J. Nonlinear Sci.* **28**(1), 91–145 (2018a)
- Arnaudon, A., Holm, D., Sommer, S.: String methods for stochastic image and shape matching. *J. Math. Imaging Vis.* **60**(6), 953–967 (2018b). <https://doi.org/10.1007/s10851-018-0823-z>
- Arnaudon, A., Holm, D.D., Sommer, S.: A geometric framework for stochastic shape analysis. *Found. Comput. Math.* **19**(3), 653–701 (2019a). <https://doi.org/10.1007/s10208-018-9394-z>
- Arnaudon, A., Holm, D.D., Sommer, S.: Stochastic metamorphosis with template uncertainties. *Math. Shapes Appl.* **37**, 75 (2019b)

- Arnaudon, A., van der Meulen, F., Schauer, M., Sommer, S.: Diffusion Bridges for Stochastic Hamiltonian Systems with Applications to Shape Analysis. arXiv:2002.00885 [physics] (2020)
- Bauer, M., Bruveris, M., Michor, P.W.: Overview of the geometries of shape spaces and diffeomorphism groups. *J. Math. Imaging Vis.* **50**(1–2), 60–97 (2014). <https://doi.org/10.1007/s10851-013-0490-z>
- Bruveris, M., Gay-Balmaz, F., Holm, D.D., Ratiu, T.S.: The Momentum Map Representation of Images. 0912.2990 (2009)
- Budhiraja, A., Dupuis, P., Maroulas, V.: Large deviations for stochastic flows of diffeomorphisms. *Bernoulli* **16**(1), 234–257 (2010). <https://doi.org/10.3150/09-BEJ203>
- Christensen, G., Rabbitt, R., Miller, M.: Deformable templates using large deformation kinematics. *Image Process. IEEE Trans.* **5**(10), 1435–1447 (1996)
- Crisan, D., Holm, D.D., Leahy, J.M., Nilssen, T.: A Variational Principle for Fluid Dynamics on Geometric Rough Paths. arXiv preprint arXiv:2005.09348 (2020)
- Delyon, B., Hu, Y.: Simulation of conditioned diffusion and application to parameter estimation. *Stoch. Process. Appl.* **116**(11), 1660–1675 (2006). <https://doi.org/10.1016/j.spa.2006.04.004>
- Weinan, E., Ren, W., Vanden-Eijnden, E.: Finite temperature string method for the study of rare events. *J. Phys. Chem. B* **109**(14), 6688–6693 (2005). <https://doi.org/10.1021/jp0455430>
- Emery, M.: *Stochastic Calculus in Manifolds*. Universitext. Springer, Berlin/Heidelberg (1989)
- Friz, P.K., Victoir, N.B.: *Multidimensional Stochastic Processes as Rough Paths: Theory and Applications*, vol. 120. Cambridge University Press, Cambridge/New York (2010)
- Grenander, U.: *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford University Press, Oxford, UK (1994)
- Grenander, U., Miller, M.I.: Computational anatomy: an emerging discipline. *Q. Appl. Math.* **LVI**(4), 617–694 (1998)
- Gubinelli, M.: Controlling rough paths. *J. Funct. Anal.* **216**(1), 86–140 (2004)
- Holm, D.D.: *Geometric Mechanics – Part II: Rotating, Translating and Rolling*, 2nd edn. Imperial College Press, London/Hackensack (2011)
- Holm, D.D.: Variational principles for stochastic fluid dynamics. *Proc. Math. Phys. Eng. Sci./R. Soc.* **471**(2176) (2015). <https://doi.org/10.1098/rspa.2014.0963>
- Holm, D.D.: Stochastic Metamorphosis in Imaging Science. arXiv:1705.10149 [math-ph] (2017)
- Holm, D.D.: Variational Formulation of Stochastic Wave-Current Interaction (SWCI). arXiv:2002.04291 [math-ph, physics:physics] (2020)
- Holm, D.D., Marsden, J.E.: Momentum Maps and Measure-Valued Solutions (Peakons, Filaments and Sheets) for the EPDiff Equation. *nlin/0312048* (2003)
- Holm, D.D., Tyrantowski, T.M.: Variational principles for stochastic soliton dynamics. *Proc. R. Soc. A* **472**(2187), 20150827 (2016). <https://doi.org/10.1098/rspa.2015.0827>
- Holm, D.D., Ratnanather, J.T., Trounev, A., Younes, L.: Soliton dynamics in computational anatomy. *NeuroImage* **23**, S170–S178 (2004). <https://doi.org/10.1016/j.neuroimage.2004.07.017>
- Hsu, E.P.: *Stochastic Analysis on Manifolds*. American Mathematical Society, Boston, MA (2002)
- Kühnel, L., Sommer, S.: Computational anatomy in theano. In: *Mathematical Foundations of Computational Anatomy (MFCA)* (2017)
- Kühnel, L., Arnaudon, A., Fletcher, T., Sommer, S.: Stochastic Image Deformation in Frequency Domain and Parameter Estimation Using Moment Evolutions. arXiv:1812.05537 [cs, math, stat] (2018)
- Kühnel, L., Sommer, S., Arnaudon, A.: Differential geometry and stochastic dynamics with deep learning numerics. *Appl. Math. Comput.* **356**, 411–437 (2019). <https://doi.org/10.1016/j.amc.2019.03.044>
- Kunita, H.: *Stochastic Flows and Stochastic Differential Equations*. Cambridge University Press, Cambridge (1997)
- Lyons, T.: Rough paths, signatures and the modelling of functions on streams. arXiv preprint arXiv:1405.4537 (2014)

- Markussen, B.: A statistical approach to large deformation diffeomorphisms. In: Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04, pp. 181–181 (2004). <https://doi.org/10.1109/CVPR.2004.16>
- Markussen, B.: Large deformation diffeomorphisms with application to optic flow. *Comput. Vis. Image Underst.* **106**(1), 97–105 (2007). <https://doi.org/10.1016/j.cviu.2005.09.006>
- Marsland, S., Shardlow, T.: Langevin equations for landmark image registration with uncertainty. *SIAM J. Imaging Sci.* **10**(2), 782–807 (2017). <https://doi.org/10.1137/16M1079282>
- Marsland, S., Sommer, S.: Riemannian geometry on shapes and diffeomorphisms: Statistics via actions of the diffeomorphism group. In: Pennec, X., Sommer, S., Fletcher, T. (eds.) *Riemannian Geometric Statistics in Medical Image Analysis*, pp. 135–167. Academic Press (2020). <https://doi.org/10.1016/B978-0-12-814725-2.00011-X>
- Miller, M., Banerjee, A., Christensen, G., Joshi, S., Khaneja, N., Grenander, U., Matejic, L.: Statistical methods in computational anatomy. *Stat. Methods Med. Res.* **6**(3), 267–299 (1997)
- Schauer, M., van der Meulen, F., van Zanten, H.: Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli* **23**(4A), 2917–2950 (2017). <https://doi.org/10.3150/16-BEJ833>
- Sommer, S.: Probabilistic approaches to geometric statistics: stochastic processes, transition distributions, and fiber bundle geometry. In: Pennec, X., Sommer, S., Fletcher, T. (eds.) *Riemannian Geometric Statistics in Medical Image Analysis*, pp. 377–416. Academic Press (2020). <https://doi.org/10.1016/B978-0-12-814725-2.00018-2>
- Sommer, S., Arnaudon, A., Kuhnel, L., Joshi, S.: Bridge simulation and metric estimation on landmark manifolds. In: *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*. Lecture Notes in Computer Science, pp. 79–91. Springer (2017). https://doi.org/10.1007/978-3-319-67675-3_8
- Staneva, V., Younes, L.: Learning shape trends: parameter estimation in diffusions on shape manifolds. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 717–725 (2017). <https://doi.org/10.1109/CVPRW.2017.101>
- Trouvé, A.: Diffeomorphisms groups and pattern matching in image analysis. *Int. J. Comput. Vis.* **28**(3), 213–221 (1998). <https://doi.org/10.1023/A:1008001603737>
- Trouve, A., Vialard, F.X.: Shape splines and stochastic shape evolutions: a second order point of view. *Q. Appl. Math.* **70**(2), 219–251 (2012). <https://doi.org/10.1090/S0033-569X-2012-01250-4>
- Vanden-Eijnden, E., Venturoli, M.: Revisiting the finite temperature string method for the calculation of reaction tubes and free energies. *J. Chem. Phys.* **130**(19), 194103 (2009). <https://doi.org/10.1063/1.3130083>
- Vialard, F.X.: Extension to infinite dimensions of a stochastic second-order model associated with shape splines. *Stoch. Process. Appl.* **123**(6), 2110–2157 (2013). <https://doi.org/10.1016/j.spa.2013.01.012>
- Wassermann, D., Toews, M., Niethammer, M., Wells, W.: Probabilistic diffeomorphic registration: representing uncertainty. In: Ourselin, S., Modat, M. (eds.) *Biomedical Image Registration*. Lecture Notes in Computer Science, pp. 72–82. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-08554-8_8
- Younes, L.: Computable elastic distances between shapes. *SIAM J. Appl. Math.* **58**, 565–586 (1998). <https://doi.org/10.1.1.45.503>
- Younes, L.: *Shapes and Diffeomorphisms*. Springer, Berlin/Heidelberg (2010)
- Zhang, M., Fletcher, P.T.: Finite-dimensional lie algebras for fast diffeomorphic image registration. *Inf. Process. Med. Imaging Proc. Conf.* **24**, 249–259 (2015). https://doi.org/10.1007/978-3-319-19992-4_19



Intrinsic Riemannian Metrics on Spaces of Curves: Theory and Computation

39

Martin Bauer, Nicolas Charon, Eric Klassen, and Alice Le Brigant

Contents

Introduction	1350
Matching of Geometric Curves Based on Reparametrization-Invariant	
Riemannian Metrics	1351
General Framework	1351
The SRV Framework	1359
Implementation	1371
The Geodesic Boundary Value Problem on Parametrized Curves	1371
Normalization by Isometries	1372
Minimization over the Reparametrization Group	1372
Open-Source Implementations	1379
Conclusion	1379
References	1380

Abstract

This chapter reviews some past and recent developments in shape comparison and analysis of curves based on the computation of intrinsic Riemannian metrics on the space of curve modulo shape-preserving transformations. We summarize

M. Bauer · E. Klassen

Department of Mathematics, Florida State University, Tallahassee, FL, USA

e-mail: bauer@math.fsu.edu; klassen@math.fsu.edu

N. Charon (✉)

Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA

e-mail: charon@cis.jhu.edu

A. Le Brigant

Department of Applied Mathematics, University Paris 1, Paris, France

e-mail: alice.le-brigant@univ-paris1.fr

the general construction and theoretical properties of quotient elastic metrics for Euclidean as well as non-Euclidean curves before considering the special case of the square root velocity metric for which the expression of the resulting distance simplifies through a particular transformation. We then examine the different numerical approaches that have been proposed to estimate such distances in practice and in particular to quotient out curve reparametrization in the resulting minimization problems.

Keywords

Elastic shape analysis · Curves in Riemannian manifolds · Sobolev metrics · Reparametrization invariance · Square root velocity transform.

Introduction

Many applications that involve quantitative comparison and statistics over sets of geometric objects like curves often rely on a certain notion of metric on the corresponding shape space. Some of them, such as medical imaging or computer vision, are concerned with the outline of an object, represented by a closed curve, while others, such as trajectory analysis or speech recognition, consider open curves drawing the evolution of a given time process in a certain space, say a manifold. In both cases, it is often interesting when studying these curves to factor out certain transformations (e.g., rotations, translations, reparametrizations), so as to study the shape of the considered object, or to deal with the considered time process regardless of speed or pace.

Beyond computing distances between shapes, a desirable goal in these applications is to perform statistical analysis on a set of shapes, e.g., to compute the mean and perform classification or principal component analysis. For this purpose, considering shapes as elements of a *shape manifold* that we equip with a Riemannian structure provides a convenient framework. In this infinite-dimensional shape manifold, points represent shapes, and the distance between two shapes is given by the length of the shortest path linking them – the geodesic. This approach allows us to do more than simply compute distances: it enables us to define the notion of an optimal deformation between two shapes, and to locally linearize the shape manifold using its tangent space. For instance, given a set of shapes, one can perform methods of standard statistical analysis in the flat representation space given by the tangent space at the barycenter.

The idea of a shape space as a Riemannian manifold was first developed by Kendall (1984), who defines shapes as “what is left” of a curve after the effects of translation and rotation and changes of scale are filtered out. Mathematically, this means defining the shape space as a quotient space, where the choice of which transformations to quotient out depends on the application. The shapes considered by Kendall are represented by labeled points in Euclidean space, and the shape spaces are finite-dimensional. More recent works deal with continuous curves with

values in a Euclidean space or a nonlinear manifold (Fig. 1), and thus with infinite-dimensional shape spaces.

There exist two main complementary approaches to define the shape space and its metric. One possibility is to deform shapes by diffeomorphisms of the entire ambient space. In this setting, metrics are defined on the space of spatial deformations, and are called *extrinsic* (or *outer*) metrics as developed in the works of Grenander (1993), Trouvé (1998), and Beg et al. (2005) among other references. Another approach consists in defining metrics directly on the space of curves itself, which are thus called *intrinsic* (or *inner*) metrics. This chapter focuses on the second approach, and studies inner metrics with certain invariance properties. We are specifically interested in the invariance to shape-preserving transformations, in particular to the action of temporal deformations, also called *reparametrizations*, which we represent by diffeomorphisms of the parameter space ($[0, 1]$ for open curves, S^1 for closed curves). In the following sections, we will introduce a class of invariant Sobolev metrics we call *elastic* on the space of immersed curves which in turn descend to metrics on the space of shapes. These were initially studied in Michor and Mumford (2005, 2007) and Mennucci et al. (2008) and in subsequent works. We will then discuss in detail the particular case of the so-called “square root velocity” (SRV) metric (Srivastava et al. 2011), a first-order invariant metric which allows for particularly simple computations not only for curves in Euclidean spaces but also curves with values in homogeneous spaces or even Riemannian manifolds. Finally, we review different methods to factor out the action of the reparametrization group, which, because of its infinite dimensionality, presents an important challenge in the computation of distances and geodesics in this framework.

Matching of Geometric Curves Based on Reparametrization-Invariant Riemannian Metrics

General Framework

Let D be either the interval $I = [0, 1]$ or the circle S^1 and $(M, \langle \cdot, \cdot \rangle)$ a finite-dimensional Riemannian manifold with TM denoting its tangent bundle. In the following we introduce the central object of interest in this book chapter, the infinite-dimensional manifold of open (respectively, closed) curves.

Lemma 1 (Michor 1980). *The space of smooth, regular curves:*

$$\text{Imm}(D, M) = \{c \in C^\infty(D, M) : \langle c'(u), c'(u) \rangle_{c(u)} \neq 0, \forall u \in D\} \quad (1)$$

is a smooth Fréchet manifold with tangent space at c the set of C^∞ vector fields along c , i.e.,

$$T_c \text{Imm}(D, M) = \{h \in C^\infty(D, TM) : h \circ \pi = c\}, \quad (2)$$

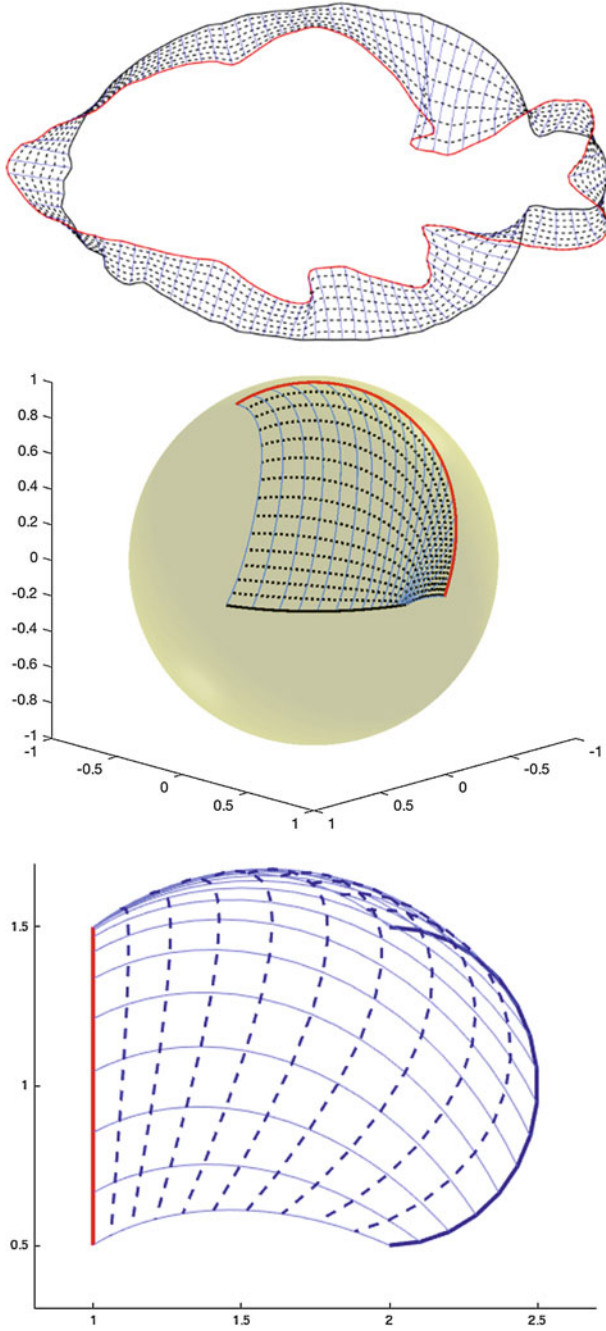


Fig. 1 (continued)

where $\pi : TM \rightarrow M$ denotes the foot point projection.

The main difficulties for understanding this result stem from the manifold structure of the ambient space M . For the convenience of the reader, we note that for $M = \mathbb{R}^d$, the situation simplifies significantly: in that case $\text{Imm}(D, \mathbb{R}^d)$ is an open subset of the infinite-dimensional vector space $C^\infty(D, \mathbb{R}^d)$, and thus tangent vectors to $\text{Imm}(D, \mathbb{R}^d)$ can be identified with smooth functions with values in \mathbb{R}^d as well. See Fig. 2 for a schematic explanation of the involved objects.

In most applications in shape analysis, one is not interested in the parametrized curve itself, but only in its features after quotienting out the action of shape-preserving transformations. Therefore, we introduce the reparametrization group of the domain D :

$$\text{Diff}_+(D) = \{\gamma \in C^\infty(D, D) : \gamma \text{ is an orientation preserving diffeomorphism.}\}. \quad (3)$$

Similarly to the space of immersions, this space carries the structure of an infinite-dimensional manifold. In fact it has even more structure, namely, it is an infinite-dimensional Lie group (Hamilton 1982, Section 4). This group acts on the space of immersed curves by composition from the right, and this action merely changes the parametrization of the curve but not its actual shape. See Fig. 2 for an example of different parametrizations of the same geometric curve.¹

Similarly, we can consider the left action of the group $\text{Isom}(M)$ of isometries of M on $\text{Imm}(D, M)$. Note that the isometry group is always a finite-dimensional group; e.g., for $M = \mathbb{R}^d$, the group $\text{Isom}(M)$ is generated by the set of translations and linear isometries (In some applications one is also interested in modding out the action of the scaling group, which requires a slight modification of the family of elastic metrics. We will not discuss these details here, but refer the interested reader to the literature, e.g., Bruveris and Møller-Andersen 2017.). Thus, the action of the



Fig. 1 Examples of geodesics on spaces of unparametrized curves w.r.t elastic metrics (target curve in red). Some intermediate curves $c(t, \cdot)$ are shown in dashed line and the trajectory of a few specific points in blue. Left figure: second-order Sobolev metric, estimated with the approach of Bauer et al. (2019a), cf. section “Relaxation of the Exact Matching Problem”. Middle figure: SRV metric for curves with values on homogeneous spaces as implemented in Su et al. (2018), where the optimal reparametrization is estimated using dynamic programming; cf. sections “Curves in Lie Groups” and “Dynamic Programming Approach”. Right figure: SRV metric for manifold-valued curves in the hyperbolic plane, as implemented in Le Brigant (2019) with successive horizontalizations; cf. section “Curves in Riemannian Manifolds”, method 1 and section “Iterative ‘Horizontalization’ Method”

¹To be mathematically exact, one should limit oneself to the slightly smaller set of free immersions in this definition, as the quotient space has some mild singularities without this restriction. We will, however, ignore this subtlety for the purpose of this book chapter.

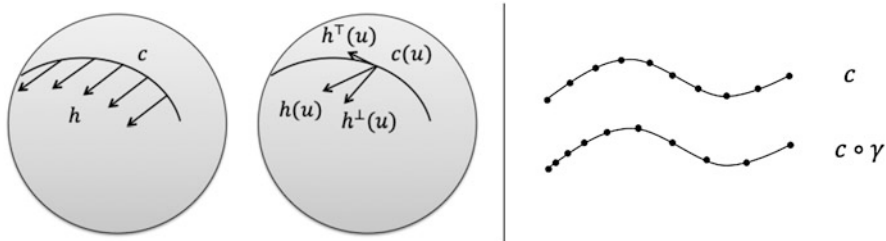


Fig. 2 Left panel: Tangent vector field to a curve $c(u)$ on the two-dimensional sphere $M = S^2$ (left) and its tangential and normal parts (right). Right panel: Two different parametrizations of the same geometric curve

infinite-dimensional group $\text{Diff}_+(D)$ is the most difficult to deal with, both from a theoretical and an algorithmic viewpoint. This allows us now to introduce the shape space of curves (To be mathematically exact, one should limit oneself to the slightly smaller set of free immersions in this definition, as the quotient space has some mild singularities without this restriction. We will, however, ignore this subtlety for the purpose of this book chapter.)

$$\mathcal{S}(D, M) := \text{Imm}(D, M) / (\text{Diff}_+(M) \times \text{Isom}(M)) \tag{4}$$

Note that sometimes we use the phrase “unparametrized shape” to refer to an element of the shape space $\mathcal{S}(D, M)$ and we shall write $[c] \in \mathcal{S}(D, M)$ the equivalence class of a parametrized curve c .

Lemma 2 (Cervera et al. 1991 and Binz and Fischer 1981). *The shape space $\mathcal{S}(D, M)$ is a smooth Frechet manifold, and the projection $p : \text{Imm}(D, M) \rightarrow \mathcal{S}(D, M)$ is a smooth submersion.*

This means specifically that the mapping p is Frechet-differentiable and that for any $c \in \text{Imm}(D, M)$, $dp(c)$ is onto from $T_c \text{Imm}(D, M)$ to $T_{[c]} \mathcal{S}(D, M)$. The so-called vertical space at c associated with the submersion is defined as $\text{Ver}_c = \{h \in T_c \text{Imm}(D, M) \mid dp(c) \cdot h = 0\}$.

We aim to introduce Riemannian metrics on the shape space $\mathcal{S}(D, M)$ by defining metrics on the space of parametrized curves that satisfy certain invariance properties. In the literature these metrics are also referred to as elastic metrics, as they account for both bending and stretching of the curve.

A Riemannian metric on $\text{Imm}(D, M)$ is a smooth family of inner products $G_c(\cdot, \cdot)$ on each tangent space $T_c \text{Imm}(D, M)$, and we call such a metric G *reparametrization-invariant* if it satisfies the relation:

$$G_c(h, k) = G_{c \circ \gamma}(h \circ \gamma, k \circ \gamma) \tag{5}$$

for all $c \in \text{Imm}(D, M)$, $h, k \in T_c \text{Imm}(D, M)$, and $\gamma \in \text{Diff}_+(D)$.

In the following we will introduce the class of Sobolev-type metrics. For the convenience of the reader, we will first discuss the special case of a first-order metric and $M = \mathbb{R}^d$. We will then generalize this to the more complicated situation of curves with values in general manifolds and more general metrics. For a curve $c \in \text{Imm}(D, \mathbb{R}^d)$ and tangent vectors $h, k \in C^\infty(D, \mathbb{R}^d)$, we let

$$G_c(h, k) = \int_D \left(\langle h, k \rangle + \left\langle \frac{h'}{|c'|}, \frac{k'}{|c'|} \right\rangle \right) |c'| du = \int_D (\langle h, k \rangle + \langle D_s h, D_s k \rangle) ds, \tag{6}$$

where the desired invariance follows directly by integration using substitution. Here $D_s = \frac{\partial_u}{|c'|}$ and $ds = |c'| du$ denote differentiation and integration with respect to arclength. These definitions naturally generalize to curves with values in abstract manifolds by replacing the partial derivative ∂_u in D_s by the covariant derivative with respect to the curve velocity $\nabla_{c'(u)}$. We will denote the induced differential operator as $\nabla_s = \frac{\nabla_{c'}}{|c'|}$.

Using this notation, a reparametrization-invariant Sobolev metric of order n on the space of manifold valued curves can be defined via

$$G_c(h, k) = \sum_{i=0}^n \int_D \langle \nabla_s^i h, \nabla_s^i k \rangle_c ds. \tag{7}$$

More generally we can consider metrics that are defined by an abstract, positive, pseudo-differential operator L_c , which satisfies the equivariance property $L_c(h) \circ \gamma = L_{c \circ \gamma}(h \circ \gamma)$ for all reparametrizations γ , immersions c , and tangent vectors h . The corresponding metric can then be written via

$$G_c(h, k) = \int_D \langle L_c(h), L_c(k) \rangle_c ds. \tag{8}$$

A particularly important example of such metrics is given by the family of elastic $G^{a,b}$ metrics – first introduced by Mio et al. (2007) for the case of planar curves:

$$G_c^{a,b}(h, k) = \int_D a^2 \langle (\nabla_s h)^\top, (\nabla_s k)^\top \rangle + b^2 \langle (\nabla_s h)^\perp, (\nabla_s k)^\perp \rangle ds, \tag{9}$$

where $a, b > 0$ are constants and \perp and \top denote the projection on the normal (respectively, tangential) part of the tangent vector. Here normal and tangential are calculated with respect to the foot-point curve c , as illustrated in Fig. 2.

As a next step, we will show that the invariance of the metric G will allow us to define an induced metric on the shape space of unparametrized curves. Before we are able to formulate this result, we review some basic facts on Riemannian submersions. Therefore let (\mathcal{M}, g_1) and (\mathcal{N}, g_2) be two (possibly infinite

dimensional) Riemannian manifolds. A Riemannian submersion is a submersion $p : (\mathcal{M}, g_1) \rightarrow (\mathcal{N}, g_2)$ such that in addition $dp : \text{Hor} \rightarrow T\mathcal{N}$ is an isometry. Here $\text{Hor} \subset T\mathcal{M}$ is the horizontal bundle, which is defined as the g_1 -orthogonal complement of the vertical bundle $\text{Ver} := \ker(dp) \subset T\mathcal{M}$. Classical results in Riemannian geometry allow us now to connect the geometry of the two Riemannian manifolds. Most importantly, for our purposes, is the fact that geodesics on (\mathcal{N}, g_2) correspond to horizontal geodesics on (\mathcal{M}, g_1) . Thus Riemannian submersions are a convenient construction in our quotient space situation, as it allows, by restricting the calculations to horizontal curves, to perform most of the analysis on the top space, i.e., the space of parametrized curves.

We are now able to describe the Riemannian submersion picture for the shape space of unparametrized curves. Consequently, this gives rise to the following result.

Theorem 1. *The reparametrization-invariant metrics (7), (8), and (9) descend to smooth Riemannian metrics on the quotient space $\mathcal{S}(D, M)$ such that the projection p becomes a Riemannian submersion.*

We want to emphasize here that this theorem is nontrivial in our setting: in finite dimensions, the invariance of the Riemannian metric would always imply the existence of a Riemannian metric on the quotient space, such that the projection is a Riemannian submersion. In our infinite-dimensional situation, the proof is slightly more delicate, as one has to show the existence of the horizontal bundle by hand. This can be done by adapting a variant of Moser's trick to the present setting. For the reparametrization-invariant metrics studied in this chapter, the horizontality condition requires one essentially to solve a differential equation of order $2n$ with n being the order of the metric. In the case where one is only interested in factoring out the reparametrization group, these two subspaces are given by

$$\text{Ver}_c = \{h = a.c' \in T_c \text{Imm}(D, M) : a \in C^\infty(D, \mathbb{R})\}, \quad (10)$$

$$\text{Hor}_c = \{k \in T_c \text{Imm}(D, M) : G_c(k, ac') = 0 \text{ for all } a \in C^\infty(D, \mathbb{R})\}; \quad (11)$$

see, e.g., Michor and Mumford (2007) and Bauer et al. (2011). If one wants to factor out in addition the group of isometries of M , one has to change the definition of the vertical and thus horizontal bundle accordingly. The exact formulas will depend on the manifold M .

The above theorem allows us to develop algorithms on the quotient space $\mathcal{S}(D, M)$ while performing most of the operations on the space of parametrized curves. In the following, we discuss how to express the geodesic distance resulting from the above Riemannian metric, which will serve as our similarity measure on the space of shapes. We will first do this for parametrized curves and then in a second step describe the induced distance on the space of geometric curves. For parametrized curves $c_0, c_1 \in \text{Imm}(D, M)$, we have

$$\text{dist}(c_0, c_1) = \inf \int_0^1 \sqrt{G_c(\partial_t c, \partial_t c)} dt, \quad (12)$$

where the infimum has to be calculated over all paths $c : [0, 1] \rightarrow \text{Imm}(D, M)$ such that $c(0) = c_0$ and $c(1) = c_1$. In the following we will usually view paths of curves as functions of two variables $c(t, u)$ where $t \in [0, 1]$ is the time variable along the path and $u \in D$ the curve parameter.

The induced geodesic distance on the quotient shape space $\mathcal{S}(D, M)$ can now be calculated via

$$\text{dist}^{\mathcal{S}}([c_0], [c_1]) = \inf_{\substack{\gamma \in \text{Diff}_+(D) \\ g \in \text{Isom}(M)}} \text{dist}(c_0, g \circ c_1 \circ \gamma) = \inf_{\substack{\gamma \in \text{Diff}_+(D) \\ g \in \text{Isom}(M)}} \text{dist}(g \circ c_0 \circ \gamma, c_1). \quad (13)$$

Note that this can be formulated as a joint optimization problem over the path of curves c , the reparametrization function γ , and the isometry $g \in \text{Isom}(M)$.

In finite dimensions, geodesic distance always gives rise to a true distance function, i.e., it is symmetric, is positive, and satisfies the triangle inequality. On the contrary, this can fail quite spectacularly in this infinite-dimensional situation, as the geodesic distance can vanish identically on the space. This phenomenon has been found first by Eliashberg and Polterovich for the $W^{-1,p}$ -metric on the symplectomorphism group (Eliashberg and Polterovich 1993). In the context of reparametrization-invariant metrics on space of immersions, this surprising result has been proven by Michor and Mumford (2005). In the following theorem, we summarize results on the geodesic distance for the class of Sobolev metrics. See Michor and Mumford (2007), Bauer et al. (2012, 2020b), and Jerrard and Maor (2019) and the references therein for further information on this topic.

Theorem 2. *The geodesic distance of the reparametrization-invariant L^2 -metric – as defined in equation (7) with $n = 0$ – vanishes on both the space of regular parametrized curves $\text{Imm}(D, M)$ and on the shape space $\mathcal{S}(D, M)$. On the other hand, the geodesic distance is positive on both of these spaces if the order of the Sobolev metric is at least one.*

This result suggests that metrics of order at least one are potentially well-suited for applications in shape analysis. For such applications, one is usually interested in computing numerically the geodesic distance as well as the corresponding optimal path between two given curves. In Riemannian geometry, these optimal paths are called minimizing geodesics, and they are locally described by the so-called geodesic equation, which is simply the first-order optimality condition for the length functional as defined in (12). In our context these equations become rather difficult; they are nonlinear PDEs of order $2n$ (where n is the order of the metric). Nevertheless there exist powerful results on existence of solutions.

In order to formulate these results, we need to introduce the space of all immersions of finite Sobolev regularity, i.e., for $s > \frac{3}{2}$, we consider the space

$$\text{Imm}^s(D, M) := \{c \in H^s(D, M) : |c'| \neq 0\}, \quad (14)$$

which is a smooth Banach manifold. Here $H^s(D, M)$ denotes the Sobolev space of order s ; see, e.g., Bauer et al. (2020c) for the exact definition in a similar notation. Note that the condition $|c'| \neq 0$ is well defined as all functions in $H^s(D, \mathbb{R}^d)$ are C^1 for $s > \frac{3}{2}$. We are now able to state the main result on geodesic and metric completeness, which is of relevance to our applications. In order to keep the presentation as concise as possible, we will formulate this result for closed curves and will only comment on the open curve case below.

Theorem 3 (Bruveris et al. 2014; Bruveris 2015 and Bauer et al. 2020c). *Let dist be the geodesic distance of the Sobolev metric G , as defined in (7), of order $n \geq 2$ on the space $\text{Imm}(S^1, M)$ of smooth regular curves. The following statements hold:*

1. *The metric G and its corresponding geodesic distance function extend smoothly to the space of Sobolev immersions $\text{Imm}^s(S^1, M)$ for all $s \geq n$.*
2. *The space $\text{Imm}^n(S^1, M)$ equipped with the geodesic distance function dist (of the Sobolev metric of order n) is a complete metric space.*
3. *For any two curves in the same connected component of $\text{Imm}^n(S^1, M)$, there exists a minimizing geodesic connecting them.*

For open curves it has been shown that the constant coefficient metric as defined in (7) is in fact not metrically complete (Bauer et al. 2019a). The reason for this is that one can always shrink down a straight line (open geodesic in the manifold M resp.) to a point using finite energy. One can, however, regain the analogue of the above completeness result for open curves by considering a length-weighted version of the Riemannian metric; see Bauer et al. (2020c).

As a direct consequence of the completeness results, we obtain the existence of optimal reparametrizations, i.e., the well-posedness of the matching problem on the space of unparametrized curves. To state our main result on existence of optimal reparametrizations, we introduce the quotient space of Sobolev immersion modulo Sobolev diffeomorphisms:

$$S^s(D, M) := \text{Imm}^s(D, M) / \text{Diff}_+^s(D) / \text{Isom}(M). \quad (15)$$

We have not determined whether this space carries the structure of a manifold. Nevertheless, we can consider the induced geodesic distance on this space and obtain the following completeness result, which we will formulate again for closed curves only.

Theorem 4 (Bruveris 2015). *Let $n \geq 2$ and let dist be the geodesic distance of the Sobolev metric of order n on $\text{Imm}^n(S^1, M)$. Then $\mathcal{S}^n(S^1, M)$ equipped with the quotient distance $\text{dist}^{\mathcal{S}}$ is a complete metric space. Furthermore, given two unparametrized curves $[c_0], [c_1] \in \mathcal{S}^n(S^1, M)$, there exists an optimal reparametrization γ and isometry g , i.e., the infimum*

$$\text{dist}^{\mathcal{S}}([c_0], [c_1]) = \inf_{\substack{\gamma \in \text{Diff}_+^n(S^1) \\ g \in \text{Isom}(M)}} \text{dist}(c_0, g \circ c_1 \circ \gamma) \quad (16)$$

is attained. Here $c_0, c_1 \in \text{Imm}(S^1, M)$ can be taken as arbitrary representatives of the geometric curves $[c_0]$ and $[c_1]$.

In the article Bruveris (2015), this result is formulated for the action of the infinite-dimensional group $\text{Diff}_+(S^1)$ only and for $M = \mathbb{R}^d$ only. The proof can however be easily adapted to incorporate the action of the compact group $\text{Isom}(M)$, and, using the results of Bauer et al. (2020c), it directly translates to the case of manifold-valued curves. Similar as in Theorem 3, this results continue to hold for open curves after changing the Riemannian metric to a length-weighted version.

For further results on general Sobolev metrics on spaces of curves, we refer to the vast literature on the topic, including Sundaramoorthi et al. (2007), Bauer et al. (2014b, 2020a), Klassen et al. (2004), Michor and Mumford (2007), Younes (1998), and Tumpach and Preston (2017). An example of a geodesic between two planar closed curves for a second-order Sobolev metric is shown in Fig. 1 (left), which was computed with the approach described later in section “Relaxation of the Exact Matching Problem”. In the following section, we will study one particular metric of order one that will lead to explicit formulas for geodesics and geodesic distance on open, parametrized curves. This will in turn allow us to recover the results on existence of geodesics and optimal reparametrizations. These optimal objects will however fail to have the regularity properties that the optimizers in this section were guaranteed to have.

The SRV Framework

Curves in \mathbb{R}^d

The reparametrization-invariant Riemannian metrics discussed above are designed to induce Riemannian metrics on the space of shapes. In general, calculating geodesics and distances with respect to these metrics requires numerical optimization, and is often computation-intensive. However, for the case of open curves in \mathbb{R}^d , one of these metrics provides geodesics and distances that are especially easy to compute. This method is known as the “square root velocity” (SRV) framework.

The main tool in this framework is the map $Q : \text{Imm}(D, \mathbb{R}^d) \rightarrow C^\infty(D, \mathbb{R}^d)$, often referred to in the literature as the SRV transform or function, defined by

$$Q(c)(u) = \frac{c'(u)}{\sqrt{|c'(u)|}}. \quad (17)$$

The importance of this map becomes evident in the following theorem by Srivastava et al. (2011), which connects it to the $G^{a,b}$ -metric (9) for a particular choice of constants a and b .

Theorem 5. *The mapping Q as defined above is an isometric immersion from the space of immersions modulo translations $\text{Imm}(D, \mathbb{R}^d)/\text{Tra}$ with the elastic $G^{1,1/2}$ -metric to $C^\infty(D, \mathbb{R}^d)$ with the flat L^2 -metric.*

Remark 1. This theorem essentially allows us to transform the computations from a complicated nonlinear manifold to a vector space equipped with a flat metric. In particular, we will see that it leads to explicit formulas for both geodesics and geodesic distance in the case of open curves. For planar curves ($d = 2$), an analogous transformation for the elastic $G^{a,b}$ -metric with $a = b = 1$ was found earlier by Younes (1998) and Younes et al. (2008). These transformations have been generalized to all parameters satisfying $a^2 - 4b^2 \geq 0$ (curves in \mathbb{R}^d) by Bauer et al. in (2014a) and more recently to arbitrary parameters (planar curves) by Needham and Kurtek (2020). We will focus in this book chapter solely on the SRV transform, but many of the results are also true for these other transformations and metrics.

In the following we will describe the SRV framework in the case of open curves, and we will only comment briefly on applications of the SRV transform to closed curves at the end of the section.

Open Curves The reason for treating the case of open curves separately is the fact that the mapping Q becomes a bijection, which will allow us to completely transform all calculations to the image of Q – a vector space. While we could perform all of these operations in the smooth category, it turns out to be beneficial to consider this method on a much larger space, which will then turn out to be the metric completion of the space of smooth immersions with respect to the SRV metric.

Henceforth, for $I = [0, 1]$, let $AC(I, \mathbb{R}^d)$ denote the set of absolutely continuous functions $I \rightarrow \mathbb{R}^d$. Since the considered metric will be invariant under translation, we standardize all curves to begin at the origin; therefore, let $AC_0(I, \mathbb{R}^d)$ denote the set of all $c \in AC(I, \mathbb{R}^d)$ such that $c(0) = 0$. We can extend the mapping Q as defined in (17) to a mapping on this larger space via $Q : AC_0(I, \mathbb{R}^d) \rightarrow L^2(I, \mathbb{R}^d)$ as follows:

$$Q(c)(u) = \begin{cases} \frac{c'(u)}{\sqrt{|c'(u)|}} & \text{if } c'(u) \neq 0; \\ 0 & \text{if } c'(u) = 0. \end{cases} \quad (18)$$

A straightforward calculation shows that Q has an explicit inverse given by

$$c(u) = Q^{-1}(q)(u) = \int_0^u |q(y)|q(y)dy, \tag{19}$$

and, thus, that Q is a bijection. $\text{Diff}_+(I)$ acts on $AC_0(I, \mathbb{R}^d)$ from the right by composition; hence, there is a unique right action of $\text{Diff}_+(I)$ on $L^2(I, \mathbb{R}^d)$ that makes Q equivariant. The explicit formula for this action is

$$(q * \gamma)(u) = \sqrt{\gamma'(u)}q(\gamma(u)), \tag{20}$$

where $q \in L^2(I, \mathbb{R}^d)$ and $\gamma \in \text{Diff}_+(I)$. Furthermore, the action of $\text{Diff}_+(I)$ on $L^2(I, \mathbb{R}^d)$ defined by (20) is by linear isometries; this follows directly by an application of integration by substitution. Finally, because Q is a bijection, we can use it to induce a Hilbert manifold structure (i.e., a smooth structure and a Riemannian metric) on $AC_0(I, \mathbb{R}^d)$. Note that this Riemannian metric is exactly the extension of the $G^{1,1/2}$ -metric to the space of absolutely continuous curves, cf. Theorem 5.

The central theme of the SRV framework is that the isometry Q enables us to transform many questions involving the geometry of $AC_0(I, \mathbb{R}^d)$ to questions involving the well-understood geometry of $L^2(I, \mathbb{R}^d)$. In particular we obtain the following theorem concerning completeness, geodesics, and geodesic distance.

Theorem 6 (Lahiri et al. 2015 and Bruveris 2016). *The space of absolutely continuous curves equipped with the SRV metric is a geodesically and metrically complete space. Furthermore, given any curves $c_0, c_1 \in AC_0(I, \mathbb{R}^d)$, the unique minimizing geodesic connecting them is given by*

$$c(t, u) = Q^{-1}((1 - t)Q(c_0)(u) + tQ(c_1)(u)), \tag{21}$$

and thus the geodesic distance between c_0 and c_1 can be calculated via

$$\text{dist}(c_0, c_1) = \sqrt{\int_0^1 |Q(c_0)(u) - Q(c_1)(u)|^2 du}. \tag{22}$$

Optimal Reparametrizations At this point, it remains to discuss the existence of optimal matchings in the definition of the quotient metric, namely, given two curves c_0, c_1 , does there exist a reparametrization $\gamma \in \text{Diff}_+(I)$ that attains the infimum in (13)? The first result in this direction was obtained by Trouvé and Younes in (2000b) (we also refer to the discussion in Younes 2019, Section 12.7.4). In this work they analyze the existence of minimizers for a general class of optimization problems on the group of diffeomorphisms of $[0, 1]$. In the case of the elastic $G^{a,b}$ -metrics (9) for open planar curves and when $a > b$, it implies that the existence of an optimal reparametrization γ always holds for piecewise C^1 curves. When $a = b$, one needs to assume in addition that there does not exist a flat region of one curve together with a point on the other curve for which the tangent vectors are pointing in

opposite directions (and with parameters within a certain distance of one another). However, for $a < b$, the conditions become much more restrictive, as one needs to exclude the situation in which there is an open interval in the parameter domain of one curve where the angle between the tangents and the tangent at a point of nearby parameter in the other curve exceeds $a\pi/b$. In particular, for the SRV metric, this basically constrains angles between tangent vectors of the two curves to be smaller than $\pi/2$, which is an impractical assumption in typical applications. As we discuss next, it turns out that by allowing instead of a single diffeomorphism a pair of “generalized” reparametrization functions, one can recover an existence result for fairly general classes of curves.

In the following we aim to describe this construction, which will require us to consider the closure of the $\text{Diff}_+(I)$ orbits on $AC_0(I, \mathbb{R}^d)$. Hence, we define an equivalence relation on $AC_0(I, \mathbb{R}^d)$ by $c_1 \sim c_2$ if and only if the $\text{Diff}_+(I)$ orbits of $Q(c_1)$ and $Q(c_2)$ have the same closure in $L^2(I, \mathbb{R}^d)$. We then define the *shape space* of open curves in \mathbb{R}^d as

$$S(I, \mathbb{R}^d) = AC_0(I, \mathbb{R}^d) / \sim,$$

and for $c \in AC_0(I, \mathbb{R}^d)$, we let $[c]$ denote the equivalence class of c under \sim .

In order to better understand these equivalence classes, we need an expanded version of $\text{Diff}_+(I)$. To be precise, define $\overline{\text{Diff}}_+(I)$ to be the set of all absolutely continuous functions $\gamma : I \rightarrow I$ such that $\gamma(0) = 0$, $\gamma(1) = 1$, and $\gamma'(u) \geq 0$ almost everywhere. Note that $\overline{\text{Diff}}_+(I)$ is only a monoid, not a group, since the only elements of $\overline{\text{Diff}}_+(I)$ that have inverses are those γ such that $\gamma'(u) \neq 0$ almost everywhere. We then have the following description of a general equivalence class of $AC_0(I, \mathbb{R}^d)$ under the relation \sim .

Lemma 3 (Lahiri et al. 2015). *Let $c \in AC_0(I, \mathbb{R}^d)$, and assume that $c'(u) \neq 0$ almost everywhere. Then the equivalence class of c under \sim is equal to*

$$\{c \circ \gamma : \gamma \in \overline{\text{Diff}}_+(I)\}.$$

Note that if $c'(u) = 0$ on a set of nonzero measure, then we cannot directly use Lemma 3 to characterize $[c]$; however, we can reparametrize c by arclength to obtain another element \tilde{c} in the same equivalence class as c , and then use Lemma 3 to characterize $[c] = [\tilde{c}]$.

We can now define a distance function on the shape space as follows: if $[c_1]$ and $[c_2]$ are elements of $S(I, \mathbb{R}^d)$, then we let

$$\text{dist}^S([c_0], [c_1]) = \inf_{w_0 \in [c_0], w_1 \in [c_1]} \|Q(w_0) - Q(w_1)\|_{L^2}.$$

Note that it seems at first that we need to consider reparametrizations of both c_0 and c_1 , because $\overline{\text{Diff}}_+(I)$ is not a group but only a monoid. However, it can be shown that the infimum will be the same if we only consider reparametrizations of one of the

curves. See Lahiri et al. (2015) and Bruveris (2016). The optimal reparametrization problem for curves in $AC_0(I, \mathbb{R}^d)$ can now be formulated as follows: suppose c_0 and c_1 are elements of $AC_0(I, \mathbb{R}^d)$, and that both have nonvanishing derivatives almost everywhere. Do there exist γ_0 and γ_1 in $\text{Diff}_+(I)$ such that

$$\|Q(c_0 \circ \gamma_0) - Q(c_1 \circ \gamma_1)\|_{L^2} = \text{dist}^S([c_0], [c_1])?$$

The following theorem gives the known results about this problem.

Theorem 7 (Lahiri et al. 2015 and Bruveris 2016). *Let c_0 and c_1 be elements of $AC_0(I, \mathbb{R}^d)$ with both having nonvanishing derivatives almost everywhere. We have:*

1. *if at least one of these curves is piecewise linear, then a pair γ_0, γ_1 of optimal reparametrizations exists;*
2. *if c_0 and c_1 are both of class C^1 , then a pair γ_0, γ_1 of optimal reparametrizations exists;*
3. *there exists a pair $c_0, c_1 \in AC_0(I, \mathbb{R}^d)$, both Lipschitz, for which no pair of optimal reparametrizations exists.*

Remark 2. Later in this chapter numerical techniques for approximating optimal reparametrizations are discussed. However, we note here that in Lahiri et al. (2015), an algorithm is developed for determining precise optimal reparametrizations for the case in which both c_0 and c_1 are piecewise linear curves. Nevertheless, since this algorithm is computationally rather expensive, usually the numerical methods described in section “[Implementation](#)” are used to solve the matching problem in practice. Furthermore, all of the algorithms that we discuss in section “[Implementation](#)” solve only for one reparametrization function (as opposed to a pair of optimal reparametrization functions as required by the above theorem). Thus the existence of minimizers for these algorithms is only guaranteed for metrics of order two or higher (by the results of Theorem 4). For lower-order metrics, such as the SRV metric, the computed distances can approximate the true geodesic distances of arbitrary precision by the density of $\text{Diff}_+(I)$ in $\overline{\text{Diff}_+(I)}$.

Closed curves For applications in which curves correspond to boundaries of planar regions, the SRV framework can be adapted to the space of closed curves. A priori, it is natural to describe a closed curve as an immersion of the circle S^1 into \mathbb{R}^d ; then the natural group of reparametrizations is $\text{Diff}_+(S^1)$. However, in order to apply the SRV methods already outlined, we will work again in the absolutely continuous category and describe a closed curve by an open curve whose initial and end points happen to coincide. Hence, we define the set of absolutely continuous, closed curves by

$$AC_0(I, \mathbb{R}^d)_{cl} = \{c \in AC_0(I, \mathbb{R}^d) : c(0) = c(1)\},$$

which is a codimension d submanifold of $AC_0(I, \mathbb{R}^d)$. In order to endow $AC_0(I, \mathbb{R}^d)_{cl}$ with a Riemannian structure, we simply restrict the SRV metric on $AC_0(I, \mathbb{R}^d)$ to this submanifold. Unfortunately, $AC_0(I, \mathbb{R}^d)_{cl}$ is not a geodesically convex submanifold, so computing geodesics and geodesic distances is not as straightforward as it is in $AC_0(I, \mathbb{R}^d)$.

Fortunately, the necessary analytical tools have been developed to solve this problem. To find a geodesic between two curves c_0 and c_1 in $AC_0(I, \mathbb{R}^d)_{cl}$, one can use the following procedure:

1. Calculate a geodesic $\{c_t\}$ between c_0 and c_1 in $AC_0(I, \mathbb{R}^d)$ using Theorem 6.
2. For each $t \in [0, 1]$, project c_t to a nearby point \tilde{c}_t in $AC_0(I, \mathbb{R}^d)_{cl}$. This requires a gradient algorithm as described in Srivastava et al. (2011) and Srivastava and Klassen (2016).
3. Deform $\{\tilde{c}_t\}$ to a geodesic in $AC_0(I, \mathbb{R}^d)_{cl}$ using a path-straightening procedure, as described in Srivastava et al. (2011) and Srivastava and Klassen (2016).

In practice, Step 3 is often omitted to save computation, because the path produced by Step 2 is generally very close to a geodesic. In order to find optimal reparametrizations for a pair of closed curves, it is not enough to consider the methods developed for open curves, because of the freedom to choose any point on a closed curve to be its starting and ending point (i.e., the point $c(0) = c(1)$). To remedy this, the algorithms discussed for open curves need to be implemented along a densely spaced set of points on one of the curves in order to choose the matching that leads to the shortest geodesic between the curves. For details, see Srivastava and Klassen (2016).

Curves in Lie Groups

In the following sections, we will discuss the methods for extending the SRV framework to curves in Lie groups, homogeneous spaces, and manifolds. We start by the simplest generalization: curves with values in Lie groups, for which the existence of a designated tangent space, the Lie algebra, makes the generalization of the SRV framework straightforward, cf. Celledoni et al. (2016b) and Su et al. (2018).

Consider a finite-dimensional Lie group \mathfrak{G} with Lie algebra $\mathfrak{g} = T_e\mathfrak{G}$, where $e \in \mathfrak{G}$ denotes the neutral element. We will assume that \mathfrak{g} has been equipped with an inner product and that this inner product has been extended to a left-invariant Riemannian metric on \mathfrak{G} . Following the square root velocity framework (SRVF) described above for curves in \mathbb{R}^d , we define the map:

$$\begin{cases} Q : AC(I, \mathfrak{G}) \rightarrow \mathfrak{G} \times L^2(I, \mathfrak{g}) \\ Q(c) = (c(0), q), \end{cases} \tag{23}$$

where

$$q(u) = \begin{cases} \frac{dL_{c(u)^{-1}}c'(u)}{\sqrt{\|c'(u)\|}} & c'(u) \neq 0 \\ 0 & c'(u) = 0 \end{cases} \tag{24}$$

Note that $L_{c(u)^{-1}}$ denotes left translation on \mathfrak{G} by $c(u)^{-1}$, which is added to transport the whole curve to the same tangent space \mathfrak{g} . Note also that the second part of this transformation is simply the generalization of the SRV transform for curves in a Euclidean space to curves with values in a Lie group and the first factor is added to keep track of the starting point. In Su et al. (2018), it is shown that the map Q is a bijection.

We put a product metric on $\mathfrak{G} \times L^2(I, \mathfrak{g})$ coming from the left-invariant metric on \mathfrak{G} and the L^2 -metric on $L^2(I, \mathfrak{g})$. Then the smooth structure and Riemannian metric on $\mathfrak{G} \times L^2(I, \mathfrak{g})$ are pulled back to $AC(I, \mathfrak{G})$ leading to the following explicit formula for the corresponding geodesic distance:

$$\text{dist}(c_0, c_1)^2 = \text{dist}^{\mathfrak{G}}(c_0(0), c_1(0))^2 + \int_0^1 \|q_1(u) - q_0(u)\|^2 du, \tag{25}$$

with $\text{dist}^{\mathfrak{G}}$ being the geodesic distance on the finite-dimensional group \mathfrak{G} and $q_i(u)$ being the q -map, as defined in equation (24), of the curve c_i . Note that the smooth structure and metric are invariant under the action of $\text{Diff}_+(I)$ and also under the left action of \mathfrak{G} . For the relation of the corresponding Riemannian metric to the class of elastic metrics as defined in equation (7), we refer to the articles Su et al. (2018) and Celledoni et al. (2016b).

Example 1. To make the above more explicit on a simple example, consider \mathfrak{G} the Lie group $\text{SO}(n, \mathbb{R})$ of real $n \times n$ orthogonal matrices with determinant one, the group operation being the standard matrix product. On the corresponding Lie algebra, which is the space of antisymmetric $n \times n$ matrices, we consider the inner product:

$$\langle A, B \rangle = \text{tr}(A^T B) = -\text{tr}(AB). \tag{26}$$

For a curve c in $\text{SO}(n, \mathbb{R})$, its q -map then simply writes:

$$q(u) = \frac{c(u)^{-1}c'(u)}{\sqrt{\text{tr}\left((c(u)^{-1}c'(u))^T(c(u)^{-1}c'(u))\right)}} = \frac{c(u)^T c'(u)}{\sqrt{\text{tr}(c'(u)^T c'(u))}}. \tag{27}$$

For the last equality, we used that $c(u)^T = c(u)^{-1}$. Moreover the geodesic distance on \mathfrak{G} is given explicitly by $\text{dist}^{\mathfrak{G}}(c_0, c_1) = \|\log(c_0^T c_1)\|_F^2$, where \log denotes the standard matrix logarithm and $\|\cdot\|_F$ the Frobenius norm. This leads to the following specific expression of the SRV distance (25) for parametrized curves in $\text{SO}(n, \mathbb{R})$:

$$\text{dist}(c_0, c_1)^2 = \|\log(c_0(0)^T c_1(0))\|_F^2 + \int_0^1 \|q_1(u) - q_0(u)\|_F^2 du. \tag{28}$$

Curves in Homogenous Spaces

For homogenous spaces the situation becomes slightly more complicated and will require an additional minimization over a finite-dimensional group.

We first recall the definition of a homogenous space. A *homogeneous space* $M = \mathfrak{G}/\mathfrak{K}$ is a quotient of a Lie group \mathfrak{G} by a closed Lie subgroup \mathfrak{K} . Note that this quotient is interpreted only as a set of left cosets; it cannot be thought of as a quotient group, since there is no assumption that \mathfrak{K} is a normal subgroup. For purposes of this chapter, we will assume that the subgroup \mathfrak{K} is compact. Examples of homogeneous spaces include spheres, Grassmannians, hyperbolic spaces, and spaces of symmetric positive definite matrices which occur in many applications.

There is a natural left action of \mathfrak{G} on $M = \mathfrak{G}/\mathfrak{K}$, and we endow M with a Riemannian metric that is invariant under this \mathfrak{G} -action as follows. First, we put a Riemannian metric on \mathfrak{G} that is left-invariant under the action of \mathfrak{G} and bi-invariant under the action of \mathfrak{K} . This is always possible using an averaging argument and the compactness of \mathfrak{K} . This metric then descends to a metric on M that is invariant under the left action of \mathfrak{G} . In order to study the shape space of curves with values in the homogeneous space M , we wish to put a Riemannian metric on the space $AC(I, M)$ that is invariant under the action of $\text{Diff}_+(I)$ and the natural left action of \mathfrak{G} . We now summarize how this is accomplished using a natural adaptation of the SRV approach for Lie groups from the previous section; see Celledoni et al. (2016a) and Su et al. (2018) for more details.

The main idea is to lift curves in M to curves in \mathfrak{G} that are horizontal (i.e., orthogonal to each \mathfrak{K} -coset that they meet). This allows us then to use the ideas for curves in Lie groups, as described in the previous section. Therefore let $\mathfrak{k} \subset \mathfrak{g}$ be the Lie algebra of \mathfrak{K} , and let \mathfrak{k}^\perp be the orthogonal complement of \mathfrak{k} in \mathfrak{g} . Let $\pi : \mathfrak{G} \rightarrow M$ denote the natural surjection. If we restrict Q^{-1} (the inverse of the map defined in equation (23)) to $\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp)$, and then compose with π , we obtain a surjection:

$$\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp) \rightarrow AC(I, M).$$

This surjection is not a bijection, because a curve c in $AC(I, M)$ does not have a unique horizontal lift to \mathfrak{G} . Rather, it has a unique horizontal lift starting at each point of $\pi^{-1}(c(0))$. To fix this, we define a right action of \mathfrak{K} on $\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp)$ by

$$(c_0, q) * y = (c_0 y, y^{-1} q y),$$

where $y \in \mathfrak{K}$, $c_0 \in \mathfrak{G}$, and $q \in L^2(I, \mathfrak{k}^\perp)$. Taking the quotient under this action precisely remedies the lack of injectivity, yielding a bijection

$$(\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp))/\mathfrak{K} \rightarrow AC(I, M).$$

This bijection, which is equivariant with respect to the left action of \mathfrak{G} , is the key tool that we use to define a Riemannian metric on $AC(I, M)$. To see this, note first that we can endow $\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp)$ with the natural product metric in the same way that we did in the case of Lie groups. Then, note that this metric is invariant under the right action of \mathfrak{R} , so it induces a metric on the quotient space $(\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp))/\mathfrak{R}$ and, hence, on $AC(I, M)$. Furthermore, this Riemannian metric is invariant under the left action of \mathfrak{G} .

Geodesics Geodesics in $L^2(I, \mathfrak{k}^\perp)$ are simply straight lines. Let us assume that we can compute geodesics in \mathfrak{G} , as well. Then geodesics in $\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp)$ are products of geodesics in these two spaces. To compute geodesics and geodesic distance in $AC(I, M)$, we need to compute geodesics in $(\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp))/\mathfrak{R}$. This is accomplished as follows. Suppose we are given two elements of $(\mathfrak{G} \times L^2(I, \mathfrak{k}^\perp))/\mathfrak{R}$, $[(c_1, q_1)]$ and $[(c_2, q_2)]$. In order to calculate a geodesic between them, we must find $y \in \mathfrak{R}$ that minimizes $d((c_1, q_1), (c_2y, y^{-1}q_2y))$. Note that this is a minimization problem over the compact Lie group \mathfrak{R} . In fact, the gradient of this function on \mathfrak{R} can be explicitly calculated (see Lemma 5 of Su et al. (2018) for the computation), reducing the computation of geodesics to an optimization problem on a compact Lie group with an explicit gradient. This technique yields efficiently computable formulas for geodesics and geodesic distances; see Celledoni et al. (2016a) and Su et al. (2018). See Fig. 4 for an example of geodesics between curves on the sphere. Furthermore, analogues of the optimal reparametrization results, cf. Theorem 7, have been proven; see Su et al. (2018).

Finally, we note that under the framework just described, the Lie group \mathfrak{G} acts on $AC(I, M)$ by isometries. Hence, for some applications, one may wish to mod out by this action (in addition to the reparametrization group) when defining the shape space of open curves in M . We observe that the current framework extends very naturally to performing the additional optimization implied by this quotient operation. We refer the reader to Su et al. (2017, 2018) for more details.

Curves in Riemannian Manifolds

Let us focus again on open curves, i.e., when D is the interval $I = [0, 1]$. For manifold-valued curves, the generalization of the SRV framework is no longer straightforward. Here we discuss three different generalizations. The first method builds on the elastic $G^{1,1/2}$ -metric, replacing ordinary derivatives by covariant derivatives with respect to the connection ∇ of the base manifold M . The two other methods, while not implementing the precise elastic method, are less computationally expensive, and often yield useful comparisons between curves. Both of these methods replace each curve in the Riemannian manifold M by a curve in a single tangent space of M , thus moving the computations to that tangent space, while in the first one, computations are done directly in the base manifold.

Method 1 In the case of curves with values in a general manifold, the elastic $G^{1,1/2}$ -metric is no longer a flat metric. However it can still be obtained as a pullback by the SRV transform of a natural metric on the tangent bundle $T \text{Imm}(I, M)$,

namely, a pointwise version of the Sasaki metric on TM . Recall that the Sasaki metric is a natural choice of metric on the tangent bundle TM that depends on the horizontal and vertical projections of each tangent vector. Intuitively, the horizontal projection of a tangent vector of $T_{(p,w)}TM$ for any $(p, w) \in TM$ corresponds to the way it moves the base point p , and its vertical projection, to the way it linearly moves w . More precisely, define just as in the Euclidean case the SRV transform to be $Q : \text{Imm}(I, M) \rightarrow T \text{Imm}(I, M)$:

$$Q(c)(u) = c'(u) / \sqrt{|c'(u)|}.$$

Consider the following metric on the tangent bundle $T \text{Imm}(I, M)$: for any pair $(c, h) \in T \text{Imm}(I, M)$, and any infinitesimal deformations $\xi_1, \xi_2 \in T_{(c,h)}T \text{Imm}(I, M)$ of the pair (c, h) , define

$$\hat{G}_{(c,h)}(\xi_1, \xi_2) = \langle \xi_1(0)^{\text{hor}}, \xi_2(0)^{\text{hor}} \rangle + \int_I \langle \xi_1(u)^{\text{ver}}, \xi_2(u)^{\text{ver}} \rangle du, \tag{29}$$

where $\xi_1(u)^{\text{hor}} \in TM$ and $\xi_1(u)^{\text{ver}} \in TM$ are the horizontal and vertical projections of the tangent vector $\xi_1(u) \in T_{(c(u),h(u))}TM$ for all $u \in I$. Then, the elastic $G^{1,1/2}$ -metric is the pullback of \hat{G} with respect to the SRV transform Q , i.e.:

$$G_c^{1,1/2}(h, k) = \hat{G}_{Q(c)}(T_c Q(h), T_c Q(k)) = \langle h(0), k(0) \rangle + \int_I \langle \nabla_{h(u)} Q(c), \nabla_{k(u)} Q(c) \rangle du, \tag{30}$$

for any curve $c \in \text{Imm}(I, M)$ and $h, k \in T_c \text{Imm}(I, M)$, where $\nabla_{h(u)} Q(c)$ denotes the covariant derivative in M of the vector field $Q(c)$ in the direction of the vector field h . Notice that here we add a position term to the integral definition (9) of the $G^{1,1/2}$ -metric in order to take into account translations. Accordingly, the energy of a path of curves $[0, 1] \ni t \mapsto c(t)$ which SRV transform we write $q(t, \cdot) = Q(c(t))$ for the $G^{1,1/2}$ -metric is given by

$$E(c) = \int_0^1 \left(|\partial_t c(t, 0)|^2 + \int_I |\nabla_t q(t, u)|^2 du \right) dt. \tag{31}$$

Here, $\nabla_t h$ denotes the covariant derivative in M of a vector field $t \mapsto h(t, u)$ along a curve $t \mapsto c(t, u)$, i.e., $\nabla_t h = \nabla_{\partial_t} h$. A variational approach yields the following conditions for such a path to be geodesic.

Proposition 1 (Le Brigant 2017). *A path of curves $[0, 1] \ni t \mapsto c(t)$ is a geodesic for the $G^{1,1/2}$ -metric if and only if its SRV representation $q(t) = Q(c(t))$ verifies the following equations:*

$$\nabla_t \partial_t c(t, 0) + r(t, 0) = 0, \quad \forall t \in [0, 1],$$

$$\nabla_t^2 q(t, u) + |q(t, u)| \left(r(t, u) + r(t, u)^T \right) = 0, \quad \forall (t, u) \in [0, 1] \times I,$$

where the vector field r depends on the curvature tensor \mathcal{R} of the base manifold M and on the parallel transport $\partial_t c(t, v)^{v,u}$ of the vector field $\partial_t c(t, \cdot)$ along $c(t, \cdot)$ from $c(t, v)$ to $c(t, u)$:

$$r(t, u) = \int_u^1 \mathcal{R}(q, \nabla_t q) \partial_t c(t, v)^{v,u} dv.$$

In the flat case $M = \mathbb{R}^d$, the curvature term r in the geodesic equation vanishes, and we obtain $\nabla_t \partial_t c(t, 0) = \partial_t^2 c(t, 0) = 0$, $\nabla_t^2 q(t, u) = \partial_t^2 q(t, u) = 0$ for all $(t, u) \in [0, 1] \times I$. We then recover the fact that the geodesic for the SRV metric between two curves in \mathbb{R}^d links their starting points with a straight line and linearly interpolates between their SRV representations. In the general case, the initial value problem for geodesics can be solved by finite differences, and the boundary value problem by geodesic shooting. In the case where the base manifold M has constant sectional curvature, e.g., the sphere or the hyperbolic plane, a comprehensive discrete framework was proposed in Le Brigant (2019) that correctly approximates the continuous setting and makes numerical computations easier.

Method 2 An important complication linked to curves taking their values in a nonlinear manifold is that tangent vectors $h \in T_c \text{Imm}(I, M)$, which are smooth vector fields along the curve c , are functions taking their values in different linear spaces. In order to bypass this difficulty, another way to go is to parallel transport the SRV transform of each curve to a single tangent space, namely, the tangent space to the curve’s starting point. Consider the vector bundle $\pi : \mathcal{C} \rightarrow M$ in which the fiber over each point $x \in M$ is the set of smooth functions in the tangent space $T_x M$, i.e., $\pi^{-1}(x) = C^\infty(I, T_x M)$. Then, define a map:

$$\begin{cases} Q^\parallel : \text{Imm}(I, M) \rightarrow \mathcal{C} \\ Q^\parallel(c) = q^\parallel \in \pi^{-1}(c(0)), \end{cases}$$

where, for each $u \in I$, $q^\parallel(u)$ is obtained by parallel translating the vector $c'(u)/\sqrt{|c'(u)|}$ along the curve c from $c(u)$ to $c(0)$. The function $q^\parallel = Q^\parallel(c)$ therefore takes its values in $T_{c(0)}M$ and is called the “transported square root velocity” (TSRV) representation of the curve c . The vector bundle \mathcal{C} is endowed with a metric that, just like (29), is a pointwise version of the Sasaki metric, i.e., defined for each $(x, v) \in \mathcal{C}$ and tangent vectors $(w_1, \eta_1), (w_2, \eta_2) \in T_{(x,v)}\mathcal{C}$ by

$$\hat{G}_{(x,v)}((w_1, \eta_1), (w_2, \eta_2)) = \langle w_1, w_2 \rangle + \int_I \langle \eta_1(u), \eta_2(u) \rangle du.$$

It is easily shown that the pullback of this metric to $\text{Imm}(I, M)$ is invariant under reparametrizations and under the group of isometries of M , and therefore yields an alternative to the elastic metric (30). The energy of a path of curves $[0, 1] \ni t \mapsto c(t)$ for this metric is given by an expression similar to (31)

$$E(c) = \int_0^1 \left(|\partial_t c(t, 0)|^2 + \int_I |\nabla_t q^\parallel(t, u)|^2 du \right) dt. \tag{32}$$

One finds that the conditions for such a curve to be a geodesic have been simplified with respect to those of the exact elastic metric framework written in Proposition 1.

Proposition 2 (Zhang et al. 2015). *A path of curves $[0, 1] \ni t \mapsto c(t)$ is a geodesic minimizing the energy (32) if and only if its TSRV representation $q^\parallel(t) = Q^\parallel(c(t))$ verifies the following equations:*

$$\begin{aligned} \nabla_t \partial_t c(t, 0) + \int_I \mathcal{R}(q^\parallel, \nabla_t q^\parallel) \partial_t c(t, u) du &= 0, \quad \forall t \in [0, 1], \\ \nabla_t^2 q^\parallel(t, u) &= 0, \quad \forall (t, u) \in [0, 1] \times I, \end{aligned}$$

where \mathcal{R} denotes the curvature tensor of the base manifold M .

In the context of finding the geodesic c between two curves c_1 and c_2 , the first equation describes the behavior of the baseline curve $t \mapsto c(t, 0)$ linking the starting points $c_1(0)$ and $c_2(0)$, and the second equation expresses the fact that $q^\parallel = Q^\parallel(c)$ is covariant linear, i.e., $q^\parallel(t, u)$ can be obtained as a linear interpolation between the TSRV representations $q_1^\parallel(u)$ and $q_2^\parallel(u)$ of c_1 and c_2 , parallel transported along the baseline curve to $c(t, 0)$. The difficulty of implementing this method depends on the particular manifold M . For curves in the sphere S^2 , the baseline curve linking the starting points is a circular arc, thus yielding simplifications with respect to the general geodesic shooting problem (Zhang et al. 2018a). The case of curves in the space of positive definite symmetric matrices is studied in Zhang et al. (2018b).

Method 3 A third possibility is to parallel transport the SRV representations of the curves to a particular reference point $p \in M$. This is the simplest method of all since the SRV representation of a curve is not only contained in a single linear space, but also this space is the same for all curves. The map of interest is then

$$\begin{cases} Q^{\parallel,p} : \text{Imm}(I, M) \rightarrow C^\infty(I, T_p M) \\ Q^{\parallel,p}(c) = q^{\parallel,p}, \end{cases}$$

where $q^{\parallel,p}(u)$ is obtained by parallel translating the vector $c'(u)/\sqrt{|c'(u)|}$ along the shortest geodesic in M from $c(u)$ to p . One then defines the distance between two curves c_0 and c_1 to be the L^2 distance between $q_0^{\parallel,p} = Q^{\parallel,p}(c_0)$ and $q_1^{\parallel,p} = Q^{\parallel,p}(c_1)$, i.e.:

$$d(c_0, c_1) = \left(\int_I |q_0^{\parallel, p}(u) - q_1^{\parallel, p}(u)|^2 du \right)^{1/2}.$$

This distance function is invariant under reparametrizations of the curves, but it is not invariant under isometries of M . The main advantage of this method is computational speed. A disadvantage is that it depends heavily on the choice of the reference point p , and may induce serious distortions for curves that venture far away from p . Finally, there can be problems with the definition of $Q^{\parallel, p}$ itself, since there can be more than one minimizing geodesic between $c(u)$ and p , and parallel translation along these different geodesics can yield different results. In general, if all the curves being compared are not too far from the reference point p , this method can yield useful results at low computational cost; see Su et al. (2014) for applications to curves in S^2 .

Implementation

In this section we will discuss the computation of the geodesic distance. We will first briefly address the case of parametrized curves. In the second part, we will then describe the main difficulty in this context which is the minimization over reparametrizations in the group $\text{Diff}_+(D)$. In particular we will describe several different approaches that have been developed to tackle this highly nontrivial task.

The Geodesic Boundary Value Problem on Parametrized Curves

For open curves with values in Euclidean space, Lie groups or homogenous spaces and the SRV metric, there exist analytic solution formulas for these operations, and thus these computations become trivial. For most of the other situations discussed in this chapter, the absence of such formulas requires one to solve these problems using numerical optimization. Therefore, one first has to choose a discretization for all of the involved objects, i.e., one has to discretize the path of curves $c(t, u)$ for $t \in [0, 1]$ and $u \in D$. A standard approach for this task consists of choosing B-splines in both time and space, i.e.:

$$c(t, u) = \sum_{i,j} c_{i,j} B_i(t) C_j(u) \quad (33)$$

where B_i and C_j are the chosen B-spline basis functions and where $c_{i,j}$ for $i = 0 \dots N_t$ and $j = 0 \dots N_u$ are the coefficients. Note that this includes as a special case the discretization of regular curves as piecewise linear functions. This procedure then reduces the calculation of the geodesic distance (12) to an unconstrained minimization problem of the discretized length functional, where the control points $c_{i,j}$ for $i = 1 \dots N_t - 1$ and $j = 0 \dots N_u$ of the B-splines are

the free variables. Here the control points of the boundary curves c_{0j} and $c_{N,j}$ are chosen as fixed parameters and are not changed in the optimization procedure. After this discretization step, one can use standard methods of numerical optimization, such as the L-BFGS method, to approximate the solution of the finite-dimensional unconstrained minimization problem. For further information, in the notation of this chapter, we refer the reader to the article Bauer et al. (2017). See also Bauer et al. (2019a), Nardi et al. (2016), and Michor and Mumford (2006).

Normalization by Isometries

The shape space $\mathcal{S}(D, M)$ in (4) involves quotienting out isometric transformations of M ; in other words one has to technically minimize in (13) the elastic distance over $g \in \text{Isom}(M)$. This is a finite-dimensional group which, for most manifolds M encountered in practice, usually has a simple parametric representation.

One common approach being used, although not rigorously equivalent to the optimization in (13), is to pre-align the two shapes with respect to isometries of M prior to estimating the elastic distance. When $M = \mathbb{R}^d$, this amounts to finding the optimal rotation and translation that best align them, which is classically addressed by Procrustes analysis, cf., for example, Dryden and Mardia (2016).

Alternatively, one can parametrize the group $\text{Isom}(M)$ and perform the minimization over g within the estimation of the distance itself, i.e., jointly with reparametrizations. For planar curves, this simply amounts to optimizing over a two-dimensional translation vector and the angle of rotation, which is the approach used, in particular, in Bauer et al. (2017, 2019a). Note that for general \mathbb{R}^d , a similar strategy is also possible by representing rotations as the exponential of antisymmetric matrices. In the case of manifold-valued curves however, normalizing with respect to isometries of M may not always be relevant or can be harder to deal with in practice. This typically depends on the availability of convenient representations of the isometry group $\text{Isom}(M)$; we refer the reader to Su et al. (2018) where some simple examples are considered.

Minimization over the Reparametrization Group

In addition to isometries of M , computation of distances and geodesics on the quotient space $\mathcal{S}(D, M)$ also requires to minimize the metric over reparametrizations in the group $\text{Diff}_+(D)$, which is here infinite-dimensional. Several different approaches have been proposed to tackle this specific issue under various situations, which we review in the following paragraphs.

Dynamic Programming Approach

A first method, which was proposed initially in Trouvé and Younes (2000a) and Mio et al. (2007), is to convert this problem into a discrete optimization one. Considering

piecewise linear (i.e., polygonal) curves, one may in turn choose to look for an optimal reparametrization of $\text{Diff}_+(D)$ that is also piecewise linear. For curves in a Euclidean space and the SRV metric, this is in part supported by the recent work of Lahiri et al. (2015) where authors show that such optimal piecewise linear reparametrizations exist. In general, as piecewise linear functions are a dense set in the space of absolutely continuous functions, it is reasonable in practice to restrict the search to reparametrizations of this form.

More specifically, assume that the two curves c_0 and c_1 are both piecewise linear. For simplicity, let's also assume that $D = [0, 1]$ and that both curves are sampled uniformly on D , namely, that c_0 and c_1 are linear on each of the subintervals $D_i = [t_i, t_{i+1}]$ for all $i = 0, \dots, N - 1$ where $t_i = i/N$. One may then approximate positive diffeomorphisms in $\text{Diff}_+(D)$ by piecewise linear homeomorphisms of D with nodes in the set $\{0, t_1, t_2, \dots, t_N\}$. Writing $J = \{t_0, t_1, t_2, \dots, t_N\}$, we can equivalently consider all the polygonal paths defined on the grid $J \times J$ joining $(0, 0)$ to $(1, 1)$ and which are the graph of an increasing piecewise linear function with nodes in J . This set Γ is now finite albeit containing a very large number of possible paths.

Nevertheless, an efficient way to determine an optimal discrete reparametrization is through dynamic programming. This is well-suited to situations where the energy to minimize can be written as an additive function over the different segments of the discrete path, which is made possible by the SRV transform in the case of elastic $G^{1,1/2}$ -metrics (or more generally for the $G^{a,b}$ -metric using the transforms of Younes et al. 2008, Needham and Kurtek 2020, and Bauer et al. 2014a). We want to note here that this method is not well-suited to cases in which one does not have access to an explicitly computable distance function, such as for the higher-order elastic metrics.

Indeed, if $\gamma \in \Gamma$ is piecewise linear on the K consecutive segments of vertices $(t_{i_0}, t_{j_0}) = (0, 0), (t_{i_1}, t_{j_1}), \dots, (t_{i_K}, t_{j_K}) = (1, 1)$ with $t_{i_0} < t_{i_1} < \dots < t_{i_K}$ and $t_{j_0} < t_{j_1} < \dots < t_{j_K}$, then the discrete energy to be minimized is expressed as

$$E(\gamma) = \|Q(c_0) - Q(c_1 \circ \gamma)\|_{L^2}^2 = \sum_{m=0}^{K-1} E(\gamma_{i_m, j_m}^{i_{m+1}, j_{m+1}})$$

where $E(\gamma_{i_m, j_m}^{i_{m+1}, j_{m+1}})$ is the energy of the linear path from vertex (t_{i_m}, t_{j_m}) to $(t_{i_{m+1}}, t_{j_{m+1}})$ and is given by

$$E(\gamma_{i_m, j_m}^{i_{m+1}, j_{m+1}}) = \frac{1}{N} \sum_{k=i_m}^{i_{m+1}-1} \left| Q(c_0)(t_k) - \sqrt{\frac{t_{j_{m+1}} - t_{j_m}}{t_{i_{m+1}} - t_{i_m}}} Q(c_1)(t_k) \right|^2.$$

Now the generic dynamic programming method first computes the minimal energy among all paths in Γ going from $(0, 0)$ to any given vertex (t_i, t_j) , which we write $E^{i,j}$, through the following iterative procedure on i :

1. Set $E^{(0,0)} = 0$.
2. For a given $i \in \{1, \dots, N\}$ and all $j \in \{1, \dots, N\}$, compute $E^{(i,j)}$ and $P^{(i,j)}$ as

$$E^{(i,j)} = \min_{(k,l) \in N_{ij}} E^{(k,l)} + E^{(i,j)}_{(k,l)}, \quad P^{(i,j)} = \operatorname{argmin}_{(k,l) \in N_{ij}} E^{(k,l)} + E^{(i,j)}_{(k,l)} \quad (34)$$

where $E^{(i,j)}_{(k,l)}$ denotes in short the energy of the linear path from vertex (t_k, t_l) to vertex (t_i, t_j) and N_{ij} is a set of admissible vertex indices connecting to (i, j) .

At the end of this process, one obtains the minimal energy $E^{(N,N)}$. A corresponding optimal path $\gamma \in \Gamma$ can be simply recovered by backtracking from the final vertex $(1, 1)$ to $(0, 0)$, the index of the vertices in γ being specifically $(i_q, j_q) = (N, N)$, $(i_{q-1}, j_{q-1}) = P^{(i_q, j_q)}$, \dots , $(i_1, j_1) = P^{(i_2, j_2)}$ and $(i_0, j_0) = P^{(i_1, j_1)} = (0, 0)$.

The choice of search neighborhood N_{ij} in the above procedure has a critical impact on the resulting complexity. To find the true minimum over all possible paths in Γ , one should technically take in (34), $N_{ij} = \{(k, l) : 0 \leq k \leq i - 1, 0 \leq l \leq j - 1\}$ for any $1 \leq i, j \leq N - 1$. This would result however in a high numerical cost of the order $O(N^4)$. It can be significantly reduced by restricting N_{ij} to a smaller set of admissible neighboring vertices. For instance, authors in Mio et al. (2007) propose to limit the search to a small square of size 3×3 with upper right vertex $(i - 1, j - 1)$. While this constrains the possible minimal and maximal slope of the estimated γ , it is generally sufficient in most cases and reduces the numerical complexity to $O(N^2)$, making the whole approach efficient in practice. Note that alternative dynamic programming algorithms have been investigated more recently, in particular in the work of Bernal et al. (2016) which makes use of adaptive strips neighborhoods to further reduce the complexity to $O(N)$.

Discretizing the Diffeomorphism Group and Using Gradient-Based Methods

A second method, which has been proposed in the context of the SRV metric in Huang et al. (2014, 2016) and for higher-order Sobolev metrics in Bauer et al. (2017), is also based on a direct discretization of the diffeomorphism group and the space of curves. However, in contrast with the previous section where diffeomorphisms of D were discretized as piecewise linear functions, this method offers more flexibility. For example, one could choose – similarly to section “The Geodesic Boundary Value Problem on Parametrized Curves” – B-spline representations of reparametrizations. Considering the distance function (16) on the space of unparametrized curves in this discretization leads again to a finite-dimensional minimization problem, which can be tackled by standard methods.

In the case when one has no access to an explicit formula for the geodesic distance – such as for higher-order Sobolev metrics – it is computationally efficient to view this problem as a joint minimization problem over the (discretized) path of curves:

$$c(t, u) = \sum_{i,j} c_{ij} B_i(t) C_j(u)$$

and the reparametrization function

$$\gamma(u) = \sum_k \gamma_k D_k(u).$$

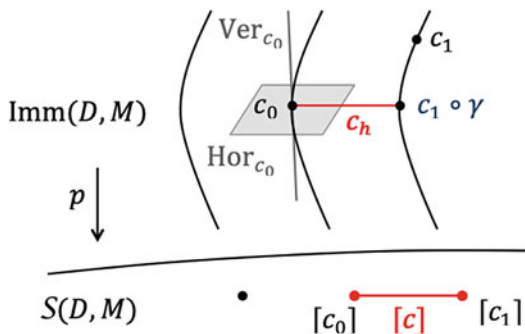
Here B_i, C_j and D_k are the chosen basis functions for the discretization of the path of curves and the reparametrization function, respectively. One difficulty in this context is that the composition of the (discretized) target $c(1, u)$ and the (discretized) reparametrization function $\gamma(u)$ typically leaves the chosen discretization space. Thus one has to consider the corresponding projection operator that projects this reparametrized curve back to the discretization space. This procedure can lead to numerical phenomena such as loss of features in the target curve. For more details we refer to the presentation in Bauer et al. (2017).

Iterative “Horizontalization” Method

Another possibility is to exploit the principal bundle structure formed by the space of parametrized curves and their shapes. The fibers of this bundle are the sets of all the curves that are identical modulo reparametrization, i.e., that project onto the same shape (Fig. 3). Any tangent vector $h \in T_c \text{Imm}(D, M)$ can be decomposed as the sum of a vertical part $h^{\text{ver}} \in \text{Ver}_c$ tangent to the fiber, which has an action of reparametrizing the curve without changing its shape, and a horizontal part $h^{\text{hor}} \in \text{Hor}_c = (\text{Ver}_c)^\perp_G$, G -orthogonal to the fiber. While the horizontal subspace depends on the choice of the reparametrization invariant metric G , the vertical subspace is always the same:

$$\text{Ver}_c = \ker dp(c) = \{mv := mc' / |c'| : m \in C^\infty([0, 1], \mathbb{R}), m(0) = m(1) = 0\}.$$

Fig. 3 Principal bundle structure formed by the space of curves and their shapes. The horizontal geodesic c_h between c_0 and the optimally matched $c_1 \circ \gamma$ projects to a geodesic $[c] = p(c_h)$ between the corresponding shapes



Paths of curves with horizontal velocity vectors are called horizontal, and horizontal geodesics for G project onto geodesics of the shape space for the Riemannian metric induced by the Riemannian submersion $p : \text{Imm}([0, 1], M) \rightarrow \mathcal{S}([0, 1], M)$; see, e.g., Michor (2008, Section 26.12). A natural way to solve the boundary value problem in the shape space is by fixing the parametrization c_0 of one of the curves and computing the horizontal geodesic linking c_0 to the closest reparametrization $c_1 \circ \gamma$ of the second curve c_1 , by iterative “horizontalizations” of geodesics. The idea is to decompose any path of curves $t \mapsto c(t) \in \text{Imm}(D, M)$ as

$$c(t, u) = c^{\text{hor}}(t, \gamma(t, u)) \quad \forall (t, u) \in [0, 1] \times D, \tag{35}$$

where $t \mapsto c^{\text{hor}}(t)$ is a horizontal path and is reparametrized by a path of diffeomorphisms $t \mapsto \gamma(t) \in \text{Diff}^+(D)$. Differentiating with respect to u and t and taking the squared norm with respect to G yields

$$\begin{aligned} |\partial_u c|^2 &= |\partial_u \gamma|^2 |\partial_u c^{\text{hor}} \circ \gamma|^2, \\ |\partial_t c|^2 &= |\partial_t c^{\text{hor}} \circ \gamma|^2 + |\partial_t \gamma|^2 |\partial_u c^{\text{hor}} \circ \gamma|^2, \end{aligned}$$

where in the second expression we have used the fact that $\partial_t c^{\text{hor}} \circ \gamma$ is horizontal by definition of c^{hor} , and $\partial_u c^{\text{hor}}$ is vertical as we can see from the first expression. From this, we immediately see that if the metric G is reparametrization invariant, taking the horizontal part of a path decreases its length:

$$L_G(c^{\text{hor}}) \leq L_G(c).$$

Therefore, by taking the horizontal part of the geodesic linking two curves c_0 and c_1 , we obtain a shorter, horizontal path linking c_0 to the fiber of c_1 , which gives a closer (in terms of G) representative $\tilde{c}_1 = c_1 \circ \gamma(1)$ of the target curve. However it is no longer a geodesic path. By computing the geodesic between c_0 and this new representative \tilde{c}_1 , we are guaranteed to reduce once more the distance to the fiber. The optimal matching algorithm simply iterates these two steps, and converges to a horizontal geodesic. At each step, the horizontal part of the geodesic can be computed using the following result.

Proposition 3 (Le Brigant 2019). *The path of diffeomorphisms $t \mapsto \gamma(t) \in \text{Diff}_+(D)$ that transforms a path $t \mapsto c(t) \in \text{Imm}(D, M)$ into a horizontal path is solution of the PDE:*

$$\partial_t \gamma(t, u) = \frac{m(t, u)}{|\partial_u c(t, u)|} \partial_u \gamma(t, u), \tag{36}$$

with initial condition $\gamma(0) = \text{Id}$, and where $m(t, u) := |\partial_t c^{\text{ver}}(t, u)|$.

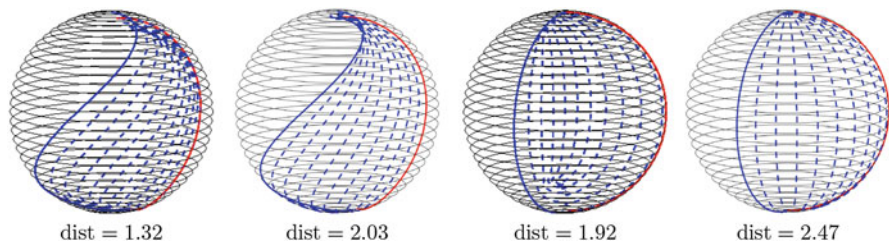


Fig. 4 Numerical comparison of the distance and geodesics between 3D curves lying on the unit sphere modulo reparametrizations: first and third pictures for the SRVF metric in the Euclidean space \mathbb{R}^3 (computed with the relaxed algorithm of Bauer et al. 2019b), second and forth pictures for the SRVF distance on S^2 (estimated with the method of Su et al. 2018). Observe that the geodesics calculated in Euclidean space do not stay on the sphere and thus result in a lower SRVF distance

This method can be applied as long as the horizontal part of a tangent vector (or equivalently, the norm of the vertical component m) can be computed. For the class of $G^{a,b}$ -elastic metrics, and for the SRV metric in particular, m can be found by solving an ODE; see Le Brigant (2019). An example of geodesic between curves in the hyperbolic plane estimated with this approach is shown in Fig. 1 (right).

Relaxation of the Exact Matching Problem

A last possible approach to deal with reparametrization invariance in the computation of geodesics and distances on the quotient space (without directly optimizing over reparametrizations) is to introduce a relaxation term for the end time constraint providing a measure of discrepancy up to reparametrization to the target curve c_1 . This is inspired by similar methods used earlier on in diffeomorphic registration frameworks; see, e.g., Glaunès et al. (2008), Durrleman et al. (2010), Charon and Trounev (2013), Roussillon and Glaunès (2016), and Kaltenmark et al. (2017) among other references. But it can also be applied in the context of elastic metric matching, as recent works such as Bauer et al. (2019a,b) and Sukurdeep et al. (2019) have shown. In this section, we will assume that curves are immersed in the Euclidean space \mathbb{R}^d .

Going back to the original formulation of the geodesic distance given by (12) and (13), the idea is to start by replacing the end time boundary constraint that $c(1) = c_1 \circ \gamma$ for some $\gamma \in \text{Diff}_+(D)$ using a surrogate fidelity (or discrepancy) term $\tilde{d}(c(1), c_1)$. Assuming that $\tilde{d}(c(1), c_1)$ is invariant to the parametrization of both $c(1)$ and c_1 , i.e., that \tilde{d} defines a distance on the quotient space, one gets the equivalence between the above boundary condition and $\tilde{d}(c(1), c_1) = 0$. Then we may choose to relax the constraint and consider the alternative variational problem:

$$\inf \int_0^1 G_c(\partial_t c, \partial_t c) dt + \lambda \tilde{d}(c(1), g \circ c_1)^2 \tag{37}$$

over all paths $c : [0, 1] \rightarrow \text{Imm}(D, \mathbb{R}^d)$ such that $c(0) = c_0$. Note that minimization over $\gamma \in \text{Diff}_+(D)$ is no longer needed here, and a minimizing path c of (37) is by construction a geodesic between c_0 and $c(1) \approx c_1$ in the quotient space $\mathcal{S}(D, \mathbb{R}^d)$. In the above, $\lambda > 0$ denotes a fixed weighting coefficient between the two terms which controls the accuracy of the matching to the target c_1 . Other strategies such as augmented Lagrangian methods can also be used to adapt the choice of this parameter in order to reach a prescribed matching accuracy, cf. Bauer et al. (2019a).

Remark 3. In the specific case of the SRV metric of section “Curves in \mathbb{R}^d ”, the variational problem (37) can be even further simplified to a minimization problem over the end curve $c^1 = c(1) \in \text{Imm}(D, \mathbb{R}^d)$ instead of a full curve path. Indeed, using the properties of the SRV transform, it is easy to see that the problem can be equivalently rewritten as

$$\inf_{c^1} \|Q(c^1) - Q(c_0)\|_{L^2}^2 + \lambda \tilde{d}(c^1, g \circ c_1)^2.$$

and leads, after discretization, to a simple minimization problem over the vertices of the deformed curve. This formulation is for instance implemented in Bauer et al. (2019b). Note that this principle also applies to other simplifying transforms associated with different choices of elastic parameters as proposed and implemented in Sukurdeep et al. (2019).

This entire approach relies on the discrepancy distance \tilde{d} which, in particular, needs to be itself independent of curve parametrization. This may sound redundant as this is also the purpose of the quotient metric construction we have been discussing all along in this chapter. Yet one can construct discrepancy metrics that are both simple and easy to compute in practice, i.e., that do not require solving an extra optimization problem. Even though these discrepancy distances do not fit within the Riemannian metric setting that we are ultimately interested in, they remain ideally suited as auxiliary terms within the elastic matching problem. While different constructions are possible, the key strategy developed in the aforementioned references consists in embedding any unparametrized curve into a certain measure space and thereby recover explicit distances derived from kernel metrics on this measure space. We will however not elaborate on the actual construction of such embeddings and metrics; the interested reader may refer to the recent survey of Charon et al. (2020).

Unlike the methods discussed in the previous sections, this relaxed approach does not necessarily compute the exact distance between the two curves. Yet it can prove particularly useful in situations where one or both curves are corrupted by noise or small topological perturbations that may otherwise considerably affect the estimated value of the distance. In addition to the example of Fig. 1 (left), we show in Fig. 4 additional geodesics for the SRVF metric between curves of \mathbb{R}^3 (lying on the unit sphere) estimated by this approach, which we compare to the geodesics for the SRVF metric on the homogeneous space S^2 .

Open-Source Implementations

Several of the methods and algorithms described above are available in open-source software packages. Here is a (non-exhaustive) list of some of these:

- **Second-order elastic metrics for curves in \mathbb{R}^d :** Implementation of a four-parameter family of metric (including in particular the family of $G^{a,b}$ -metric) is available at <https://github.com/h2metrics/h2metrics>
Both the inexact matching approach of section “[Relaxation of the Exact Matching Problem](#)” and the gradient-based approach of section “[Discretizing the Diffeomorphism Group and Using Gradient-Based Methods](#)” are implemented.
- **SRV framework for curves in \mathbb{R}^d :** several different implementations for this classical method exist. This includes in particular the R-package by J. Tucker <https://cran.r-project.org/web/packages/fdasrvf/> and the Matlab implementation of M. Bruveris as available on GitHub: <https://github.com/martinsbruveris/libsrvf>
In the second one, both the dynamic programming approach of section “[Dynamic Programming Approach](#)” and the explicit solution formula discussed in Remark 2 are implemented.
- **SRV metric for curves in homogenous spaces and Lie groups:** Code for several choices for the target space M can be found at <https://github.com/zhesu1/SRVFhomogeneous>
Optimal reparametrizations are estimated using the dynamic programming approach of section “[Dynamic Programming Approach](#)”.

Conclusion

In this chapter, we reviewed the current state of the art of curve comparison through intrinsic quotient Riemannian metrics for Euclidean as well as non-Euclidean curves. We discussed the theoretical framework, in particular the questions of non-degeneracy of Sobolev metrics and geodesic completeness of the corresponding infinite-dimensional manifolds before analyzing more specifically the case of the SRV metric for which the variational expression of the distance considerably simplifies. We also discussed several numerical approaches that have been proposed for the computation of such metrics in the different settings and for which several open-source implementations are available.

There are many directions in which this framework can be extended. One is the construction and computation of corresponding intrinsic metrics between surfaces modulo reparametrizations. Due to their significantly more complex structure than curves, this is a subject of ongoing and active investigations both from the mathematical and numerical sides: we refer interested readers, e.g., to Jermyn et al. (2017), Kurtek et al. (2011), Su et al. (2020), Tumpach et al. (2015), and Kilian et al. (2007).

Going back to curves, as noted in Remark 1, there have been several extensions and variations of the SRV framework which introduced simplifying transforms for other first-order metrics than the specific one considered in section “The SRV Framework”. We finally mention the recent work of Younes (2018) which explored the possibility to combine intrinsic Sobolev metrics with extrinsic diffeomorphism-based metrics within a hybrid framework.

Acknowledgments M. Bauer was partially supported by NSF-grant 1912037 (collaborative research in connection with NSF-grant 1912030) and NSF-grant 1953244 (collaborative research in connection with NSF-grant 1953267). N. Charon was partially supported by NSF-grant 1945224 and NSF-grant 1953267 (collaborative research in connection with NSF-grant 1953244). Eric Klassen gratefully acknowledges the support of the Simons Foundation-grant 317865.

References

- Bauer, M., Harms, P., Michor, P.W.: Sobolev metrics on shape space of surfaces. *J. Geom. Mech.* **3**(4), 389–438 (2011)
- Bauer, M., Bruveris, M., Harms, P., Michor, P.W.: Vanishing geodesic distance for the Riemannian metric with geodesic equation the KdV-equation. *Ann. Glob. Anal. Geom.* **41**(4), 461–472 (2012)
- Bauer, M., Bruveris, M., Marsland, S., Michor, P.W.: Constructing reparameterization invariant metrics on spaces of plane curves. *Differ. Geom. Appl.* **34**, 139–165 (2014a)
- Bauer, M., Bruveris, M., Michor, P.W.: Overview of the geometries of shape spaces and diffeomorphism groups. *J. Math. Imag. Vis.* **50**(1–2), 60–97 (2014b)
- Bauer, M., Bruveris, M., Harms, P., Møller-Andersen, J.: A numerical framework for Sobolev metrics on the space of curves. *SIAM J. Imag. Sci.* **10**(1), 47–73 (2017)
- Bauer, M., Bruveris, M., Charon, N., Møller-Andersen, J.: A relaxed approach for curve matching with elastic metrics. *ESAIM: Control Optim. Calc. Var.* **25**, 72 (2019a)
- Bauer, M., Charon, N., Harms, P.: Inexact elastic shape matching in the square root normal field framework. In: *Geometric Science of Information*, pp. 13–20. Springer, Cham (2019b)
- Bauer, M., Harms, P., Michor, P.W.: Fractional sobolev metrics on spaces of immersions. *Calc. Var. Partial Differ. Equ.* **59**(2), 1–27 (2020a)
- Bauer, M., Harms, P., Preston, S.C.: Vanishing distance phenomena and the geometric approach to sqg. *Arch. Ration. Mech. Anal.* **235**(3), 1445–1466 (2020b)
- Bauer, M., Maor, C., Michor, P.W.: Sobolev metrics on spaces of manifold valued curves. arXiv preprint arXiv:2007.13315 (2020c)
- Beg, M.F., Miller, M.I., Trounev, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **61**, 139–157 (2005)
- Bernal, J., Dogan, G., Hagwood, C.R.: Fast dynamic programming for elastic registration of curves. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1066–1073 (2016)
- Binz, E., Fischer, H.R.: The manifold of embeddings of a closed manifold. In: *Differential Geometric Methods in Mathematical Physics*, pp. 310–325. Springer, Berlin/Heidelberg/New York (1981)
- Bruveris, M.: Completeness properties of Sobolev metrics on the space of curves. *J. Geom. Mech.* **7**(2), 125–150 (2015)
- Bruveris, M.: Optimal reparametrizations in the square root velocity framework. *SIAM J. Math. Anal.* **48**(6), 4335–4354 (2016)
- Bruveris, M., Møller-Andersen, J.: Completeness of length-weighted Sobolev metrics on the space of curves (2017). arXiv:1705.07976

- Bruveris, M., Michor, P.W., Mumford, D.: Geodesic completeness for Sobolev metrics on the space of immersed plane curves. In: Forum of Mathematics, Sigma, vol. 2. Cambridge University Press, Cambridge (2014)
- Celledoni, E., Eidnes, S., Schmeding, A.: Shape analysis on homogeneous spaces: a generalised srvt framework. In: The Abel Symposium, pp. 187–220. Springer (2016a)
- Celledoni, E., Eslitzbichler, M., Schmeding, A.: Shape analysis on lie groups with applications in computer animation. *J. Geom. Mech.* **8**(3), 273–304 (2016b)
- Cervera, V., Mascaro, F., Michor, P.W.: The action of the diffeomorphism group on the space of immersions. *Differ. Geom. Appl.* **1**(4), 391–401 (1991)
- Charon, N., Trounev, A.: The varifold representation of non-oriented shapes for diffeomorphic registration. *SIAM J. Imag. Sci.* **6**(4), 2547–2580 (2013)
- Charon, N., Charlier, B., Glaunès, J., Gori, P., Roussillon, P.: Fidelity metrics between curves and surfaces: currents, varifolds, and normal cycles. In: Riemannian Geometric Statistics in Medical Image Analysis, pp. 441–477. Academic Press, San Diego (2020)
- Dryden, I.L., Mardia, K.V.: *Statistical Shape Analysis, with Applications in R*, 2nd edn. Wiley, Chichester (2016)
- Durrleman, S., Fillard, P., Pennec, X., Trounev, A., Ayache, N.: Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents. *NeuroImage* **55**(3), 1073–1090 (2010)
- Eliashberg, Y., Polterovich, L.: Bi-invariant metrics on the group of Hamiltonian diffeomorphisms. *Int. J. Math.* **4**(5), 727–738 (1993)
- Glaunès, J., Qiu, A., Miller, M., Younes, L.: Large deformation diffeomorphic metric curve mapping. *Int. J. Comput. Vis.* **80**(3), 317–336 (2008)
- Grenander, U.: *General Pattern Theory: A Mathematical Study of Regular Structures*. Clarendon Press Oxford, Oxford/Clarendon/New York (1993)
- Hamilton, R.S.: The inverse function theorem of Nash and Moser. *Am. Math. Soc.* **7**(1), 65–122 (1982)
- Huang, W., Gallivan, K.A., Srivastava, A., Absil, P.-A.: Riemannian optimization for registration of curves in elastic shape analysis. *J. Math. Imag. Vis.* **54**(3), 320–343 (2016)
- Huang, W., Gallivan, K.A., Srivastava, A., Absil, P.-A., et al.: Riemannian optimization for elastic shape analysis. In: *Mathematical Theory of Networks and Systems*. Springer (2014)
- Jermyn, I.H., Kurtek, S., Laga, H., Srivastava, A.: Elastic shape analysis of three-dimensional objects. *Synth. Lect. Comput. Vis.* **12**(1), 1–185 (2017)
- Jerrard, R.L., Maor, C.: Vanishing geodesic distance for right-invariant sobolev metrics on diffeomorphism groups. *Ann. Glob. Anal. Geom.* **55**(4), 631–656 (2019)
- Kaltenmark, I., Charlier, B., Charon, N.: A general framework for curve and surface comparison and registration with oriented varifolds. In: *Computer Vision and Pattern Recognition (CVPR)* (2017)
- Kendall, D.G.: Shape manifolds, procrustean metrics, and complex projective spaces. *Bull. Lond. Math. Soc.* **16**(2), 81–121 (1984)
- Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. In: *ACM Transactions on Graphics (TOG)*, vol. 26, p. 64. ACM (2007)
- Klassen, E., Srivastava, A., Mio, M., Joshi, S.H.: Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(3), 372–383 (2004)
- Kurtek, S., Klassen, E., Ding, Z., Jacobson, S.W., Jacobson, J.L., Avison, M.J., Srivastava, A.: Parameterization-invariant shape comparisons of anatomical surfaces. *IEEE Trans. Med. Imag.* **30**(3), 849–858 (2011)
- Lahiri, S., Robinson, D., Klassen, E.: Precise matching of PL curves in \mathbb{R}^N in the square root velocity framework. *Geom. Imag. Comput.* **2**(3), 133–186 (2015)
- Le Brigan, A.: Computing distances and geodesics between manifold-valued curves in the SRV framework. *J. Geom. Mech.* **9**(2), 131–156 (2017)
- Le Brigan, A.: A discrete framework to find the optimal matching between manifold-valued curves. *J. Math. Imag. Vis.* **61**(1), 40–70 (2019)

- Mennucci, A.C., Yezzi, A., Sundaramoorthi, G.: Properties of Sobolev-type metrics in the space of curves. *Interfaces Free Bound.* **10**(4), 423–445 (2008)
- Michor, P.W.: *Manifolds of Differentiable Mappings*, vol. 3. Birkhauser and Springer (1980)
- Michor, P.W.: *Topics in Differential Geometry*, vol. 93. American Mathematical Society, Providence (2008)
- Michor, P.W., Mumford, D.: Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. *Doc. Math.* **10**, 217–245 (2005)
- Michor, P.W., Mumford, D.: Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.* **8**, 1–48 (2006)
- Michor, P.W., Mumford, D.: An overview of the riemannian metrics on spaces of curves using the Hamiltonian approach. *Appl. Comput. Harmon. Anal.* **23**(1), 74–113 (2007)
- Mio, W., Srivastava, A., Joshi, S.: On shape of plane elastic curves. *Int. J. Comput. Vis.* **73**(3), 307–324 (2007)
- Nardi, G., Peyré, G., Vialard, F.-X.: Geodesics on shape spaces with bounded variation and Sobolev metrics. *SIAM J. Imag. Sci.* **9**(1), 238–274 (2016)
- Needham, T., Kurtek, S.: Simplifying transforms for general elastic metrics on the space of plane curves. *SIAM J. Imag. Sci.* **13**(1), 445–473 (2020)
- Roussillon, P., Glaunès, J.: Kernel metrics on normal cycles and application to curve matching. *SIAM J. Imag. Sci.* **9**(4), 1991–2038 (2016)
- Srivastava, A., Klassen, E.: *Functional and Shape Data Analysis*. Springer Series in Statistics. Springer, New York (2016)
- Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.H.: Shape analysis of elastic curves in Euclidean spaces. *IEEE T. Pattern Anal.* **33**(7), 1415–1428 (2011)
- Su, J., Kurtek, S., Klassen, E., Srivastava, A.: Statistical analysis of trajectories on Riemannian manifolds: bird migration, hurricane tracking and video surveillance. *Ann. Appl. Stat.* **8**(1), 530–552 (2014)
- Su, Z., Klassen, E., Bauer, M.: The square root velocity framework for curves in a homogeneous space. In: *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 680–689 (2017)
- Su, Z., Klassen, E., Bauer, M.: Comparing curves in homogeneous spaces. *Differ. Geom. Appl.* **60**, 9–32 (2018)
- Su, Z., Bauer, M., Preston, S.C., Laga, H., Klassen, E.: Shape analysis of surfaces using general elastic metrics. *J. Math. Imag. Vis.* **62**, 1087–1106 (2020)
- Sukurdeep, Y., Bauer, M., Charon, N.: An inexact matching approach for the comparison of plane curves with general elastic metrics. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 512–516. IEEE (2019)
- Sundaramoorthi, G., Yezzi, A., Mennucci, A.C.: Sobolev active contours. *Int. J. Comput. Vis.* **73**(3), 345–366 (2007)
- Trouvé, A.: Diffeomorphisms groups and pattern matching in image analysis. *Int. J. Comput. Vis.* **28**(3), 213–221 (1998)
- Trouvé, A., Younes, L.: Diffeomorphic matching problems in one dimension: Designing and minimizing matching functionals. In: *European Conference on Computer Vision*, pp. 573–587. Springer (2000a)
- Trouvé, A., Younes, L.: On a class of diffeomorphic matching problems in one dimension. *SIAM J. Control Optim.* **39**(4), 1112–1135 (2000b)
- Tumpach, A.B., Drira, H., Daoudi, M., Srivastava, A.: Gauge invariant framework for shape analysis of surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 46–59 (2015)
- Tumpach, A.B., Preston, S.C.: Quotient elastic metrics on the manifold of arc-length parameterized plane curves. *J. Geom. Mech.* **9**(2), 227–256 (2017)
- Younes, L.: Computable elastic distances between shapes. *SIAM J. Appl. Math.* **58**(2), 565–586 (1998)
- Younes, L.: Hybrid Riemannian metrics for diffeomorphic shape registration. *Ann. Math. Sci. Appl.* **3**(1), 189–210 (2018)
- Younes, L.: *Shapes and Diffeomorphisms*. Springer (2019)

- Younes, L., Michor, P.W., Shah, J., Mumford, D.: A metric on shape space with explicit geodesics. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur.* **19**(1), 25–57 (2008)
- Zhang, Z., Su, J., Klassen, E., Le, H., Srivastava, A.: Video-based action recognition using rate-invariant analysis of covariance trajectories. *arXiv preprint arXiv:1503.06699* (2015)
- Zhang, Z., Klassen, E., Srivastava, A.: Phase-amplitude separation and modeling of spherical trajectories. *J. Comput. Graph. Stat.* **27**(1), 85–97 (2018a)
- Zhang, Z., Su, J., Klassen, E., Le, H., Srivastava, A.: Rate-invariant analysis of covariance trajectories. *J. Math. Imag. Vis.* **60**(8), 1306–1323 (2018b)



An Overview of SaT Segmentation Methodology and Its Applications in Image Processing

40

Xiaohao Cai, Raymond Chan, and Tiejong Zeng

Contents

Introduction	1386
SaT Methodology	1389
SaT-Based Methods and Applications	1392
T-ROF Method	1392
Two-Stage Method for Poisson or Gamma Noise	1393
SLaT Method for Color Images	1396
Two-Stage Method for Hyperspectral Images	1398
Tight-Frame-Based Method for Images with Vascular Structures	1400
Wavelet-Based Segmentation Method for Spherical Images	1401
Three-Stage Method for Images with Intensity Inhomogeneity	1403
Conclusions	1405
References	1406

Abstract

As a fundamental and challenging task in many subjects such as image processing and computer vision, image segmentation is of great importance but is constantly challenging to deliver, particularly, when the given images or data are

X. Cai (✉)

School of Electronics and Computer Science, University of Southampton, Southampton, UK
e-mail: x.cai@soton.ac.uk

R. Chan (✉)

Department of Mathematics, College of Science, City University of Hong Kong, Kowloon Tong, Hong Kong, China
e-mail: rchan.sci@cityu.edu.hk

T. Zeng (✉)

Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China
e-mail: zeng@math.cuhk.edu.hk

corrupted by different types of degradations like noise, information loss, and/or blur. In this article, we introduce a segmentation methodology – smoothing and thresholding (SaT) – which can provide a flexible way of producing superior segmentation results with fast and reliable numerical implementations. A bunch of methods based on this methodology are to be presented, including many applications with different types of degraded images in image processing.

Keywords

Image segmentation · Image processing · Mumford-shah model · Variational model; Inverse problem

Introduction

Image segmentation aims to group objects in an image with similar characteristics together. It is one of the fundamental tasks in image processing and computer vision, having numerous engineering, medical, and commercial applications. It also serves as a preliminary step for higher level computer vision tasks like object recognition and interpretation. Most of the methods in literature face the following dilemmas: (i) lack of flexibility, applicability, and interpretability and (ii) difficult to trade off the efficiency and effectiveness. It is therefore not an easy task for users to know which method could fulfill their needs. In this regard, the users are required to make modifications here and there on existing methods accordingly, which is however frustrating if the users are not familiar with segmentation technologies. It is important to have a segmentation methodology which is simple to understand and apply and, at the same time, fast and reliable. In this article, we introduce a segmentation methodology – smoothing and thresholding (SaT) – which is able to meet these challenges (Cai et al. 2013b, 2017, 2019; Cai and Steidl 2013; Chan et al. 2014).

The piecewise constant Mumford-Shah (PCMS) model (nonconvex, a special case of the Mumford-Shah model (Mumford and Shah 1989)) and the Rudin-Osher-Fatemi (ROF) model (convex, Rudin et al. 1992) are two of the most famous variational models in the research areas of image segmentation and restoration, respectively. Note that image restoration intends to remove image degradations such as noise, blur, or occlusions.

Let $\Omega \subset \mathbb{R}^2$ be a bounded, open set with Lipschitz boundary and $f : \Omega \rightarrow [0, 1]$ be a given (degraded) image. In 1989 Mumford and Shah proposed solving segmentation problems by minimizing over $\Gamma \subset \Omega$ and $u \in H^1(\Omega \setminus \Gamma)$ the energy functional

$$E_{\text{MS}}(u, \Gamma; \Omega) = \mathcal{H}^1(\Gamma) + \lambda' \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx + \lambda \int_{\Omega} (u - f)^2 dx, \quad \lambda', \lambda > 0, \quad (1)$$

where \mathcal{H}^1 denotes the one-dimensional Hausdorff measure in \mathbb{R}^2 . The functional E_{MS} contains three terms: the penalty term on the length of Γ , the H^1 semi-norm that enforces the smoothness of u in $\Omega \setminus \Gamma$, and the data fidelity term controlling the

distance of u to the given image f . Related approaches in a spatially discrete setting were proposed in Blake and Zisserman (1987) and Geman and Geman (1984). An early attempt to solve the challenging task of finding a minimizer of the nonconvex and non-smooth Mumford-Shah functional (1) was done by approximating it using a sequence of simpler elliptic problems; see Ambrosio and Tortorelli (1990) for the so-called Ambrosio-Tortorelli approximation. Many approaches to simplify model (1) were meanwhile proposed in the literature, for example, in Pock et al. (2009b), a convex relaxation of the model was suggested. Another important simplification is to restrict its solution to be piecewise constant, which leads to the so-called PCMS model.

The PCMS model is based on the restriction $\nabla u = 0$ on $\Omega \setminus \Gamma$, which results in

$$E_{\text{PCMS}}(u, \Gamma; \Omega) = \mathcal{H}^1(\Gamma) + \lambda \int_{\Omega} (u - f)^2 dx. \quad (2)$$

Assuming that $\Omega = \bigcup_{i=0}^{K-1} \Omega_i$ with pairwise disjoint sets Ω_i and constant functions $u(x) \equiv m_i$ on Ω_i , $i = 0, \dots, K - 1$, model (2) can be rewritten as

$$E_{\text{PCMS}}(\Omega, m) = \frac{1}{2} \sum_{i=0}^{K-1} \text{Per}(\Omega_i; \Omega) + \lambda \sum_{i=0}^{K-1} \int_{\Omega_i} (m_i - f)^2 dx, \quad (3)$$

where $\Omega := \{\Omega_i\}_{i=0}^{K-1}$, $m := \{m_i\}_{i=0}^{K-1}$, and $\text{Per}(\Omega_i; \Omega)$ denotes the perimeter of Ω_i in Ω . If the number of phases is two, i.e., $K = 2$, the PCMS model is the model of the active contours without edges (Chan-Vese model) (Chan and Vese 2001),

$$E_{\text{CV}}(\Omega_1, m_0, m_1) = \text{Per}(\Omega_1; \Omega) + \lambda \left(\int_{\Omega_1} (m_1 - f)^2 dx + \int_{\Omega \setminus \Omega_1} (m_0 - f)^2 dx \right). \quad (4)$$

In Chan and Vese (2001), the authors proposed to solve (4), where it can easily get stuck in local minima. To overcome this drawback, a convex relaxation approach was proposed in Chan et al. (2006a). More precisely, it was shown that a global minimizer of $E_{\text{CV}}(\cdot, m_0, m_1)$ for fixed m_0, m_1 can be found by solving

$$\bar{u} = \arg \min_{u \in BV(\Omega)} \left\{ TV(u) + \lambda \int_{\Omega} ((m_0 - f)^2 - (m_1 - f)^2) u dx \right\}, \quad (5)$$

and setting $\Omega_1 := \{x \in \Omega : \bar{u}(x) > \rho\}$ for any choice of $\rho \in [0, 1)$; see also Bellettini et al. (1991) and Bresson et al. (2007). Note that the first term of (5) is known as the total variation (TV) and the space BV is the space of functions of bounded variation; see Section 2 for the definition. In other words, (5) is a tight relaxation of the Chan-Vese model with fixed m_0 and m_1 . For the convex formulation of the full model (4), see Brown et al. (2012).

There are many other approaches for two-phase image segmentation based on the Chan-Vese model and its convex version; see, e.g., Zhang et al. (2008), Bresson et al. (2007), Dong et al. (2010), and Bauer et al. (2017). In particular, a hybrid level

set method was proposed in Zhang et al. (2008), which replaces the first term of (4) by a boundary feature map and the data fidelity terms in (4) by the difference between the given image f and a fixed threshold chosen by a user or a specialist. Method Zhang et al. (2008) was used in medical image segmentation. However, since every time it needs the user to choose a proper threshold for its model, it is not automatic and thus its applications are restricted. In Bresson et al. (2007), the TV term of (5) was replaced by a weighted TV term which helps the new model to capture much more important geometric properties. In Dong et al. (2010), the TV term of (5) was replaced by a wavelet frame decomposition operator which, similar to the model in Bresson et al. (2007), can also capture important geometric properties. Nevertheless, for its solution u , no similar conclusions as the ones in Chan et al. (2006a) can be addressed; that is, there is no theory to support that its segmentation result $\Omega_1 = \{x : u(x) > \rho\}$ for $\rho \in [0, 1)$ is a solution as to some kind of objective functional. In Bauer et al. (2017), the Chan-Vese model was extended for 3D biopore segmentation in tomographic images.

In Vese and Chan (2002), Chan and Vese proposed a multiphase segmentation model based on the PCMS model using level sets. However, this method can also get stuck easily in local minima. Convex (non-tight) relaxation approaches for the PCMS model were proposed, which are basically focusing on solving

$$\min_{m_i, u_i \in [0, 1]} \left\{ \sum_{i=0}^{K-1} \int_{\Omega} |\nabla u_i| dx + \lambda \sum_{i=0}^{K-1} \int_{\Omega} (m_i - f)^2 u_i dx \right\}, \quad \text{s.t.} \quad \sum_{i=0}^{K-1} u_i = 1. \tag{6}$$

For more detail along this line, refer, e.g., to Bar et al. (2011), Cai (2015), Cai et al. (2015), Lellmann and Schnörr (2011), Li et al. (2010), Pock et al. (2009a), Yuan et al. (2010b), Zach et al. (2008) and the references therein.

In 1992, Rudin, Osher, and Fatemi (Rudin et al. 1992) proposed the variational model

$$\min_{u \in BV(\Omega)} \left\{ TV(u) + \frac{\mu}{2} \int_{\Omega} (u - f)^2 dx \right\}, \quad \mu > 0. \tag{7}$$

which has been studied extensively in the literature; see, e.g., Chambolle (2005), Chambolle et al. (2010), Chan et al. (2006b) and references therein.

A subtle connection between image segmentation and image restoration has been raised in Cai et al. (2013b). In detail, a two-stage image segmentation method is proposed – SaT method – which finds the solution of a convex variant of the Mumford-Shah model in the first stage, followed by a thresholding step in the second one. The convex minimization functional in the first stage (the smoothing stage) is the ROF functional (7) plus an additional smoothing term $\int_{\Omega} |\nabla u|^2 dx$. In Cai et al. (2019), a linkage between the PCMS and ROF models was shown, which gives rise to a new image segmentation paradigm: manipulating image segmentation through image restoration plus thresholding. This is also the essence of the SaT segmentation methodology.

The remainder of this article is organized as follows. Firstly, the SaT segmentation and its advantages are introduced. After that, more SaT-based methods and applications are presented and demonstrated, followed by a brief conclusion.

SaT Methodology

The main procedures of the SaT segmentation methodology are first smoothing and then thresholding, where the smoothing step is executed by solving pertinent convex objective functions (note that most of segmentation models in literature are nonconvex and therefore much harder to handle compared to convex models) and the thresholding step is just completed by thresholding the result from the smoothing step using proper thresholds; see an instance given below.

The smoothing process in Cai et al. (2013b) is to solve the convex minimization problem (cf. the non-smooth Mumford-Shah functional (1)):

$$\inf_{g \in W^{1,2}(\Omega)} \left\{ \frac{\mu}{2} \int_{\Omega} (f - Ag)^2 dx + \frac{\lambda}{2} \int_{\Omega} |\nabla g|^2 dx + \int_{\Omega} |\nabla g| dx \right\}, \quad (8)$$

where λ and μ are positive parameters and A is the blurring operator if the observed image is blurred by A or the identity operator if there is no blurring. The minimizer of (8) is a smoothed approximation of f . The first term in (8) is the data-fitting term, the second term ensures smoothness of the minimizer, and the third term ensures regularity of the level sets of the minimizer. We emphasize that model (8) can be minimized quickly by using currently available efficient algorithms such as the split-Bregman algorithm (Goldstein and Osher 2009) or the Chambolle-Pock method (Chambolle and Pock 2011). After we have obtained g in (8), assume we are given the thresholds

$$\min\{g\} = \rho_0 < \rho_1 < \dots < \rho_{K-1} < \rho_K = \max\{g\}.$$

Then we threshold g by setting $x \in \Omega$ to be in the sub-domain Ω_i if $\rho_{i-1} \leq g(x) < \rho_i$. The values $\{\rho_i\}_{i=1}^{K-1}$ can be obtained by applying the K-means method, a popular clustering method, on the intensity of g , or they can be obtained by trial and error in order to get a finer segmentation.

Theorem 1. *Let Ω be a bounded connected open subset of \mathbb{R}^2 with a Lipschitz boundary. Let $f \in L^2(\Omega)$ and $\text{Ker}(A) \cap \text{Ker}(\nabla) = \{0\}$, where A is a bounded linear operator from $L^2(\Omega)$ to itself and $\text{Ker}(A)$ is the kernel of A . Then (8) has a unique minimizer $g \in W^{1,2}(\Omega)$.*

Proof. See Cai et al. (2013b) for the detailed proof.

Figures 1, 2, and 3 illustrate the SaT framework using the two-phase segmentation strategy in Cai et al. (2013b).

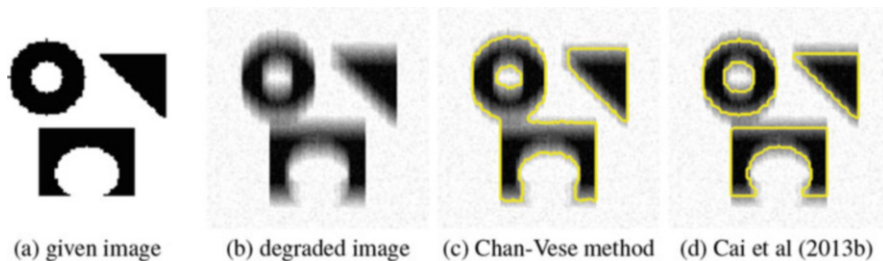


Fig. 1 Segmentations with Gaussian noise and blur. (a) Given binary image; (b) degraded image with motion blur (for the motion blur, the motion is vertical and the filter size is 15) and Gaussian noise (with mean 10^{-3} and variance 2×10^{-3}); (c) Chan-Vese method (Chan and Vese 2001); and (d) SaT segmentation with K-means thresholding (Cai et al. 2013b)

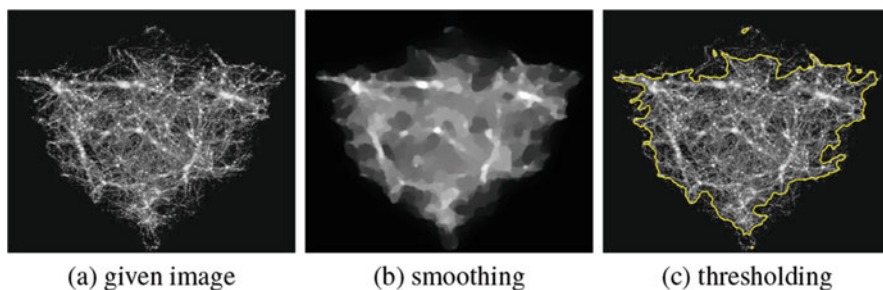


Fig. 2 SaT segmentation framework illustration using a two-phase segmentation example. (a) Given image (size 384×480); (b) obtained smoothed image (i.e., a solution of the convex model in Cai et al. 2013b); (c) segmentation result (boundary highlighted in yellow color) after thresholding (b) using threshold 0.2. Particularly, (b) and (c) correspond to the first and second steps in the SaT segmentation framework, respectively

The good performance of the SaT approach is solidly backed up. If we set the parameter λ in (8) to zero, one can show (see Cai and Steidl 2013 and Cai et al. 2019) that the SaT method is equivalent to the famous Chan-Vese segmentation method (Chan and Vese 2001), which is a simplified Mumford-Shah model. Furthermore, numerical experiments show that a properly selected λ can usually increase segmentation accuracies.

The SaT method is very efficient and flexible. It performs excellently for degraded images (e.g., noisy and blurry images and images with information loss). It also has the following advantages. Firstly, the smoothing model with (8) is strictly convex. This guarantees a unique solution of (8), which can be solved efficiently by many optimization methods. Secondly, the thresholding step is independent of the smoothing step. Therefore, the SaT approach is capable of segmentations with arbitrary phases, and one can easily try different thresholds without recalculating (8). On the contrary, for other segmentation methods, the number of phases K has to be determined before the calculation, and it is usually computationally expensive to regenerate a different segmentation if K changes. Thirdly, the SaT approach is

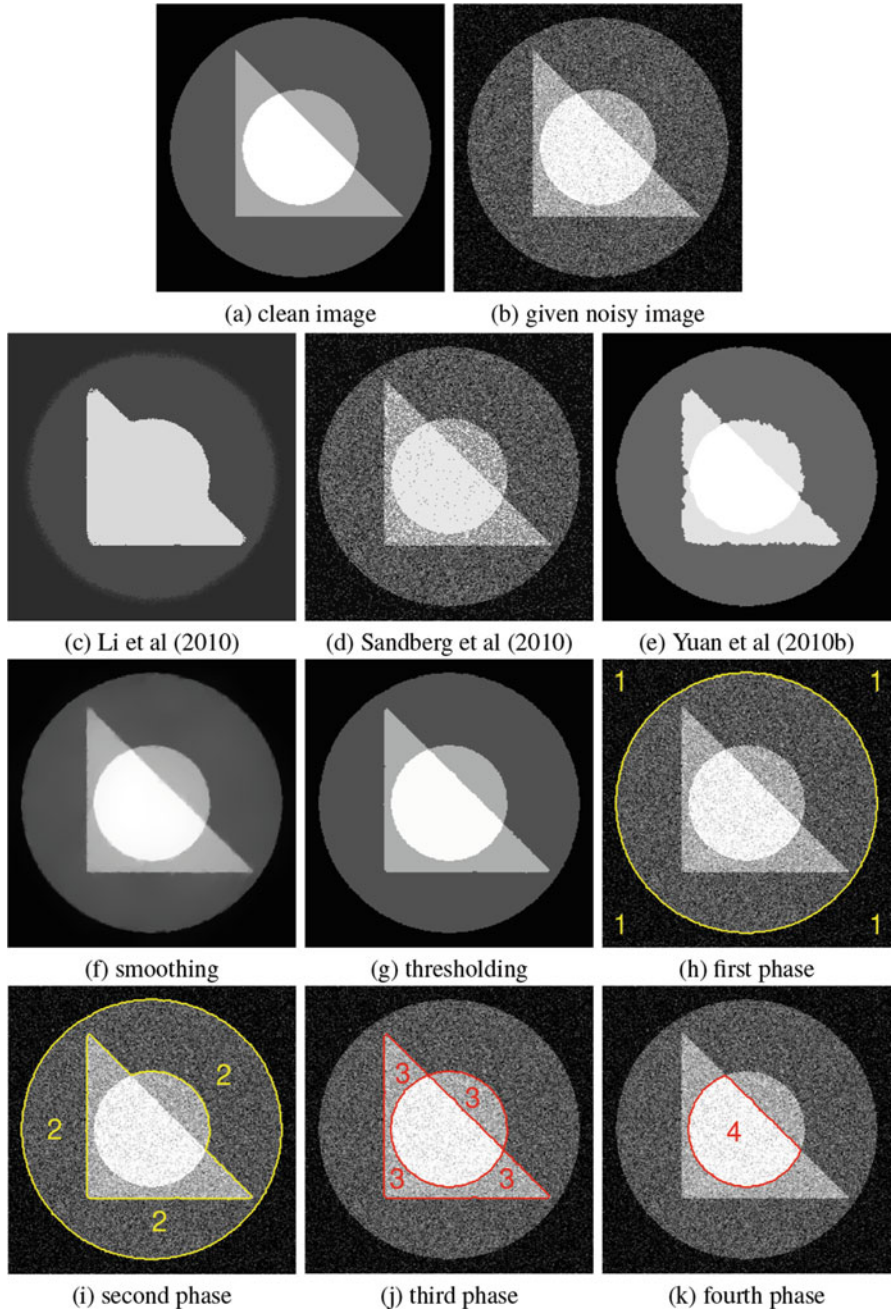


Fig. 3 (continued)

very flexible. One can easily modify the smoothing step to better segment images with specific properties.

The SaT segmentation methodology has been used for images corrupted by Poisson and Gamma noises (Chan et al. 2014), degraded color images (Cai et al. 2017), images with intensity inhomogeneity (Chan et al. 2019), hyperspectral images (Chan et al. 2020), vascular structures (Cai et al. 2011, 2013a), spherical images (Cai et al. 2020), etc.

SaT-Based Methods and Applications

To exemplify the excellent performance of the SaT segmentation methodology, in the following, a few methods related to the SaT segmentation methodology with different applications are introduced.

T-ROF Method

In Cai and Steidl (2013) and Cai et al. (2019), the *thresholded-ROF* (T-ROF) method was proposed. It highlights a relationship between the PCMS model (3) and the ROF model (7), proving that thresholding the minimizer of the ROF model leads to a partial minimizer of the PCMS model when $K = 2$ (Chan-Vese model (4)), which remains true under specific assumptions when $K > 2$.

Theorem 2 (Relation between ROF and PCMS models for $K = 2$). *Let $K = 2$ and $u^* \in BV(\Omega)$ solve the ROF model (7). For given $0 < m_0 < m_1 \leq 1$, let $\tilde{\Sigma} := \{x \in \Omega : u^*(x) > \frac{m_1+m_0}{2}\}$ fulfill $0 < |\tilde{\Sigma}| < |\Omega|$. Then $\tilde{\Sigma}$ is a minimizer of the PCMS model (4) for $\lambda := \frac{\mu}{2(m_1-m_0)}$ and fixed m_0, m_1 . In particular, $(\tilde{\Sigma}, m_0, m_1)$ is a partial minimizer of (4) if $m_0 = \text{mean}_f(\Omega \setminus \tilde{\Sigma})$ and $m_1 = \text{mean}_f(\tilde{\Sigma})$.*

Proof. See Cai et al. (2019) for the detailed proof.

This linkage between the PCMS model and the ROF model validates the effectiveness of the proposed SaT method in Cai et al. (2013b) for image segmentation. Due to the significance of the PCMS model and ROF model, respectively, in image segmentation and image restoration, this linkage bridges to some extent these two research areas and might serve as a motivation to improve and design better



Fig. 3 Four-phase segmentation. (a) Clean 256×256 image; (b) given noisy image (Gaussian noise with zero mean and variance 0.03); (c)–(e) results of methods Li et al. (2010), Sandberg et al. (2010) and Yuan et al. (2010b), respectively; (f) obtained smoothed image (i.e., a solution of the convex model in Cai et al. (2013b)); (g) segmentation result after thresholding (f) using thresholds $\rho_1 = 0.1652$, $\rho_2 = 0.4978$, $\rho_3 = 0.8319$; (h)–(k) boundary of each phase of the result in (g)

methods. A direct benefit is the newly proposed efficient segmentation method – T-ROF method. The T-ROF method exactly follows the paradigm to perform image segmentation through image restoration plus iterative thresholding, where these thresholds are selected automatically following certain rules. This appears to be more sophisticated than the SaT method (Cai et al. 2013b) which is based on K-means. It is worth emphasizing that the ROF model and the T-ROF model both need to be solved once, and the T-ROF method gives optimal segmentation results akin to the PCMS model. The convergence of the T-ROF method regarding threshold automatic selection is also proved.

On the one hand, the T-ROF method can be regarded as a special case of the SaT method. However, it is directly obtained from the linkage between the PCMS model and the ROF model and thus is more theoretically justified. Moreover, the strategy of choosing the thresholds automatically and optimally in the T-ROF method is not covered in the SaT method in Cai et al. (2013b). The strategy makes the T-ROF method more effective particularly for degraded images whose phases have close intensities. On the other hand, the T-ROF method inherits the advantages of the SaT method – fast speed and computational cost independent of the required number of phases K . In contrast, methods solving the PCMS model become computational demanding as the required number of phases increases.

To demonstrate the great performance of the T-ROF method, Fig. 4 gives an example of segmenting a synthetic retina image based on one manually segmented result from the DRIVE dataset (<http://www.isi.uu.nl/Research/Databases/DRIVE/>). Figures 4a and b are the clean manual segmentation image and the noisy image generated by adding Gaussian noise with mean 0 and variance 0.1. Note that in Fig. 4a, the original binary manual segmentation image is changed to three phases by lowering the intensity of those vessels on the right hand side from 1 to 0.3; the intensities of the background and the vessels on the left hand side are, respectively, 0 and 1. Obviously, segmenting the noisy three-phase image in Fig. 4b is extremely challenging due to those thin blood vessels which have a big chance of being smoothed out. Figure 4 shows that the T-ROF method together with the SaT method (Cai et al. 2013b) achieves the best result (with much faster speed compared with others). For more detail of the T-ROF method, please refer to Cai and Steidl (2013) and Cai et al. (2019).

Two-Stage Method for Poisson or Gamma Noise

The Poisson noise and the multiplicative Gamma noise are firstly recalled below. For the Poisson noise, for each pixel $x \in \Omega$, we assume that the intensity $f(x)$ is a random variable following the Poisson distribution with mean $g(x)$, i.e., its probability mass function is

$$p_{f(x)}(n; g(x)) = \frac{(g(x))^n e^{-g(x)}}{n!},$$

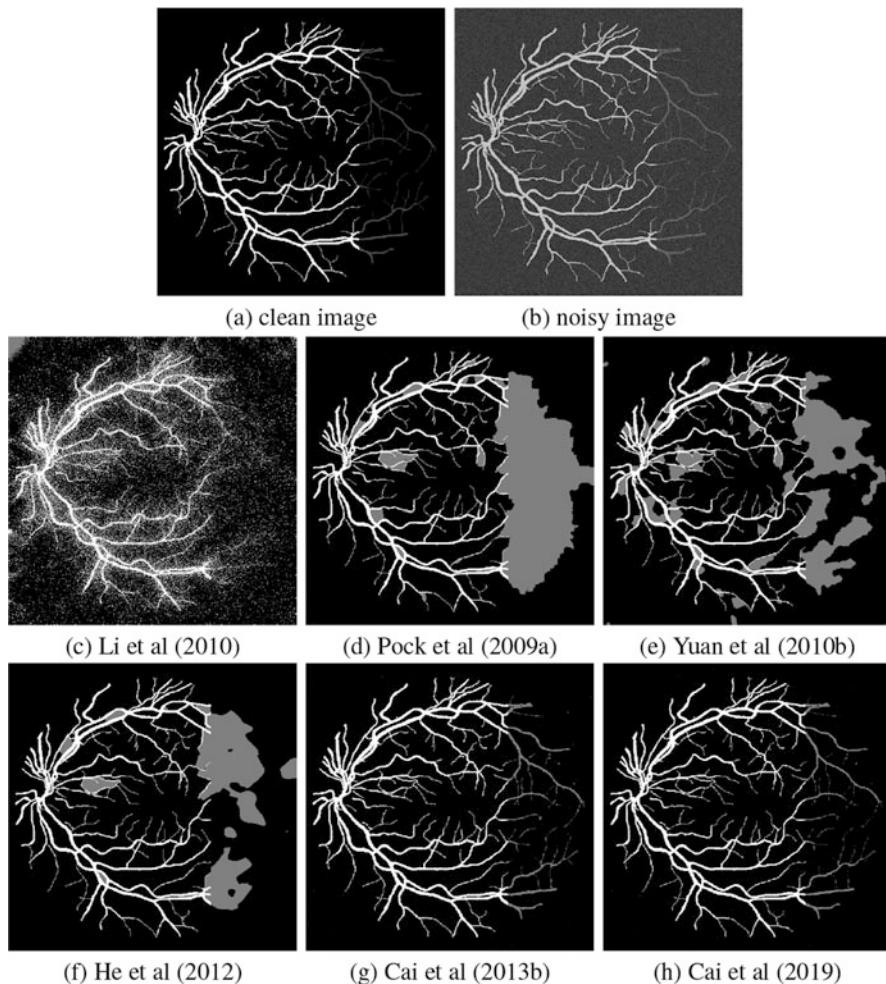


Fig. 4 Retina image segmentation which contains extremely thin vessels (size 584×565). (a) Clean image; (b) noisy image; (c)–(h) results of methods (Li et al. 2010; Pock et al. 2009a; Yuan et al. 2010b; He et al. 2012; Cai et al. 2013b) and the T-ROF method (Cai et al. 2019), respectively

where n is the intensity of f at the pixel x . In this case, we say that f is corrupted by Poisson noise. For the Gamma noise, suppose that for each pixel $x \in \Omega$ the random variable $\eta(x)$ follows the Gamma distribution, i.e., its probability density function is

$$p_{\eta(x)}(y; \theta, K) = \frac{1}{\theta^K \Gamma(K)} y^{K-1} e^{-\frac{y}{\theta}} \text{ for } y \geq 0, \quad (9)$$

where Γ is the usual Gamma-function and θ and K denote the scale and shape parameters in the Gamma distribution, respectively. Notice that the mean of $\eta(x)$ is

$K\theta$ and the variance of $\eta(x)$ is $K\theta^2$. For multiplicative noise, we assume in general that the mean of $\eta(x)$ equals 1; see Aubert and Aujol (2008) and Durand et al. (2010). Then we have $K\theta = 1$ and its variance is $1/K$. We assume the degraded image is $f(x) = g(x) \cdot \eta(x)$ and say that f is corrupted by multiplicative Gamma noise.

The construction of a data fidelity term can be inspired by the following observations. With the abuse of notation, suppose f is the given image with noise following a certain statistical distribution, and let $p(g|f)$ be the conditional probability of g when we have observed f . Then based on maximum a posteriori approach, restoring the image g is equivalent to maximizing the probability $p(g|f)$. Assume the prior distribution of g is given by

$$p(g) \propto \exp(-\beta \int_{\Omega} |\nabla g| dx),$$

where β is a parameter. If the noise follows the Poisson distribution, then maximizing $p(g|f)$ corresponds to minimizing the functional

$$\int_{\Omega} (g - f \log g) dx + \beta \int_{\Omega} |\nabla g| dx \tag{10}$$

(see Le et al. 2007). If the noise is multiplicative following the Gamma distribution, then maximizing $p(u|f)$ corresponds to minimizing the functional

$$\int_{\Omega} \left(\frac{f}{g} + \log g\right) dx + \beta \int_{\Omega} |\nabla g| dx \tag{11}$$

(see Aubert and Aujol 2008). However, it is observed in the numerical examples in Aubert and Aujol (2008) and Shi and Osher (2008) that for the denoising model (11) the noise survives much longer at low image values if we increase the regularization parameter. Therefore, in Shi and Osher (2008), the authors suggested to take $w = \log g$ and change the objective functional (11) to

$$\int_{\Omega} (f e^{-w} + w) dx + \beta \int_{\Omega} |\nabla w| dx. \tag{12}$$

In Chan et al. (2014), a two-stage method for segmenting blurry images in the presence of Poisson or multiplicative Gamma noise is proposed. It was inspired by the SaT segmentation method in Cai et al. (2013b) and the Gamma noise denoising method in Steidl and Teuber (2010). Specifically, the data fidelity term of the model (8) at the first stage of the SaT segmentation method in Cai et al. (2013b) was replaced by the one which is suitable for Gamma noise, i.e.,

$$\inf_{g \in W^{1,2}(\Omega)} \left\{ \mu \int_{\Omega} (Ag - f \log Ag) dx + \frac{\lambda}{2} \int_{\Omega} |\nabla g|^2 dx + \int_{\Omega} |\nabla g| dx \right\}. \tag{13}$$

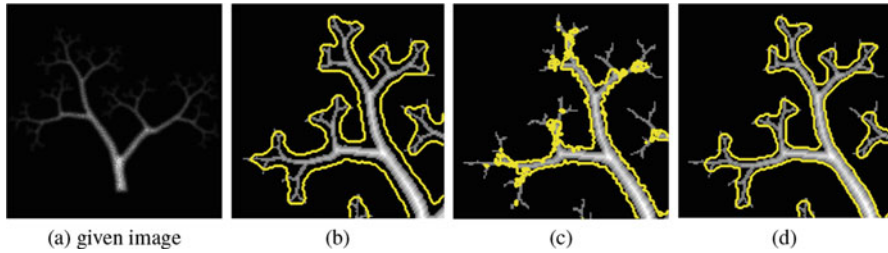


Fig. 5 Segmentations of a fractal image corrupted with Gamma noise and blur. (a) Degraded image; (b)–(d) results of methods (Yuan et al. 2010a; Dong et al. 2011), and SaT with user-provided thresholds (Chan et al. 2014), respectively. For clarity, only the top-left corner of the segmentations is shown. We see that the SaT method produces the best result, with the segmentation line (the yellow line) very close to the real boundary

Then at the second stage the solution g is thresholded to reveal different segmentation features.

The follow Theorems 3 and 4 assure that model (13) has a unique minimizer with identity or blurring operator A .

Theorem 3. *Let Ω be a bounded connected open subset of \mathbb{R}^2 with a Lipschitz boundary. Let $f \in L^\infty(\Omega)$ with $\inf f > 0$ and A be the identity operator. Then (13) has a unique minimizer $u \in W^{1,2}(\Omega)$ satisfying $0 < \inf f \leq u \leq \sup f$.*

Proof. See Chan et al. (2014) for the detailed proof.

Theorem 4. *Let Ω be a bounded connected open subset of \mathbb{R}^2 with a Lipschitz boundary. Let $f \in L^\infty(\Omega)$ with $\inf f > 0$, and let \mathcal{A} be a continuous linear operator from $W^{1,2}(\Omega)$ to itself. Assume $\text{Ker}(\mathcal{A}) \cap \text{Ker}(\nabla) = \{0\}$, and then (13) has a unique minimizer $u \in W^{1,2}(\Omega)$.*

Proof. See Chan et al. (2014) for the detailed proof.

Figure 5 gives an example which shows the great performance of the SaT-based method (Chan et al. 2014) for images with multiplicative Gamma noise.

SLaT Method for Color Images

Extending or conceiving segmentation methods for color images is not a simple task since one needs to discriminate segments with respect to both luminance and chrominance information. The two-phase Chan-Vese model (Chan and Vese 2001) was generalized to deal with vector-valued images in Chan et al. (2000) by combining the information in the different channels using the data fidelity term. Many methods are applied in the usual RGB color space (Cai 2015; Chan et al.

2000; Cremers et al. 2007; Jung et al. 2007; Kay et al. 2009; Martin et al. 2001; Pock et al. 2009a; Storath and Weinmann 2014), among others. It is often mentioned that the RGB color space is not well adapted to segmentation because for real-world images the R, G, and B channels can be highly correlated. In Rotaru et al. (2008), RGB images are transformed into HSI (hue, saturation, and intensity) color space in order to perform segmentation. In Benninghoff and Garcke (2014), a general segmentation approach was developed for gray-value images and further extended to color images in the RGB, the HSV (hue, saturation, and value), and the CB (chromaticity-brightness) color spaces. However, a study on this point in Paschos (2001) has shown that the Lab (perceived lightness, red-green, and yellow-blue) color space defined by the CIE (Commission Internationale de l'Eclairage) is better adapted for color image segmentation than the RGB and the HSI color spaces. In Cardelino et al. (2013), RGB input images were first converted to Lab space. In Wang et al. (2015), color features were described using the Lab color space and texture using histograms in RGB space.

A careful examination of the methods that transform a given RGB image to another color space (HSI, CB, Lab, etc.) before performing the segmentation task has shown that these algorithms are always applied only to noise-free RGB images (though these images unavoidably contain quantization and compression noise). For instance, this is the case of Benninghoff and Garcke (2014), Cardelino et al. (2013), Rotaru et al. (2008) and Wang et al. (2015), among others. One of the main reasons is that if the input RGB image is degraded, the degradation would be hard to control after a transformation to another color space (Paschos 2001).

A color image is usually represented by a vector valued function $f = (f_1, f_2, f_3) : \Omega \rightarrow \mathbb{R}^3$, where the components f_1 , f_2 , and f_3 generally represent red, green, and blue channels, respectively. The difficulty for color image segmentation partly comes from the strong interchannel correlation. A novel extension of the SaT approach is the smoothing, lifting, and thresholding (SLaT) method introduced in Cai et al. (2017), which is able to work on vector-valued (color) images possibly corrupted with noise, blur, and missing data. One first solves (8) for the three components f_1 , f_2 , and f_3 to obtain three smooth functions g_1 , g_2 , and g_3 . Then one transforms (g_1, g_2, g_3) to another color space $(\bar{g}_1, \bar{g}_2, \bar{g}_3)$ which can reduce interchannel correlation. This is the lifting process, and the Lab color space is usually a good choice. In the thresholding step, one performs K-means to threshold the lifted image with 6 channels $(g_1, g_2, g_3, \bar{g}_1, \bar{g}_2, \bar{g}_3)$ to get the phases.

In Cai et al. (2017), model (8) was also extended to tackle information loss and both Gaussian and Poisson noises. In particular, the existence and uniqueness of the extended model with information loss and both Gaussian and Poisson noises was also proved.

This SLaT method is easy to implement with promising results; see Fig. 6 with images chosen from the Berkeley Segmentation Dataset and Benchmark (<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>) Moreover, the SLaT method has the ability to segment color images corrupted by noise, blur, or when some pixel information is lost. More experimental results in Cai et al. (2017) on



Fig. 6 Color image segmentation for degraded images. First row: degraded color images (the first three images are degraded by various noise and blur, and the last two images are degraded by 60% information loss and noise). Second row: Pock et al. (2009a). Third row: SLaT method (Cai et al. 2017)

RGB images coupled with Lab secondary color space demonstrate that the method gives much better segmentation results for images with degradation than some state-of-the-art segmentation models both in terms of quality and CPU time cost.

Two-Stage Method for Hyperspectral Images

Remotely sensed hyperspectral images are images taken from drones, airplanes, or satellites that record a wide range of electromagnetic spectrum, typically more than 100 spectral bands from visible to near-infrared wavelengths. Since different materials reflect different spectral signatures, one can identify the materials at each pixel of the image by examining its spectral signatures. Hyperspectral images are used in many applications, including agriculture (Patel et al. 2001; Datt et al. 2003), disaster relief (Eismann et al. 2009), food safety (Gowen et al. 2007), military (Manolakis and Shaw 2002; Stein et al. 2002), and mineralogy (Hörig et al. 2001).

One of the most important problems in hyperspectral data exploitation is hyperspectral image classification. It has been an active research topic in past decades (Fauvel et al. 2013). The pixels in the hyperspectral image are often labeled manually by experts based on careful review of the spectral signatures and investigation of the scene. Given these ground-truth labels of some pixels (also called “training pixels”), the objective of hyperspectral image classification is to assign labels to part or all of the remaining pixels (the “testing pixels”) based on their spectral signatures and their locations.

In Chan et al. (2020), a two-stage method was proposed based on the SaT method (Cai et al. 2013b) for hyperspectral image classification. Pixel-wise classifiers, such

as the classical support vector machine (SVM), consider spectral information only. As spatial information is not utilized, the classification results are not optimal, and the classified image may appear noisy. Many existing methods, such as morphological profiles, superpixel segmentation, and composite kernels, exploit the spatial information. In Chan et al. (2020), a two-stage approach was proposed. In the first stage, SVMs are used to estimate the class probability for each pixel. In the second stage, the SaT model is applied to each probability map to denoise and segment the image into different classes. The proposed method effectively utilizes both spectral and spatial information of the datasets and is fast as only convex minimization is needed in addition to the SVMs.

We emphasize that the convex model used in Chan et al. (2020) is the model (8) at the first stage of the SaT segmentation method in Cai et al. (2013b), with a constraint, i.e.,

$$\inf_{g_k} \left\{ \frac{\mu}{2} \int_{\Omega} (f_k - Ag_k)^2 dx + \frac{\lambda}{2} \int_{\Omega} |\nabla g_k|^2 dx + \int_{\Omega} |\nabla g_k| dx \right\},$$

$$\text{s.t. } g_k|_{\Omega_{\text{train}}} = f_k|_{\Omega_{\text{train}}}, \quad (14)$$

where f_k represents the probability map of the k th class obtained from stage one using the SVM method, g_k is the improved probability map of the k th class, and Ω_{train} is the set of training pixels. After obtaining g_k , $k = 1 \dots, K$, individual pixels will be labeled to a set which possesses the maximum values among $g_k(x)$, $k = 1 \dots, K$. Note that the above stage two performs like the SaT strategy.

Figure 7 gives an example which shows the great performance of the two-stage method (Chan et al. 2020) for hyperspectral image classification. For more detail, please refer to Chan et al. (2020).

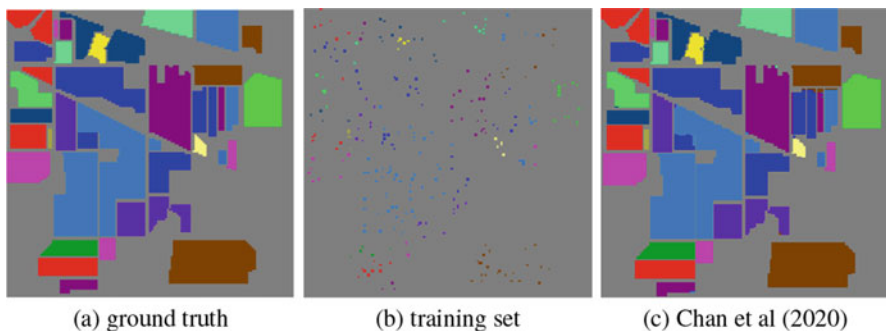


Fig. 7 Hyperspectral image classification of the Indian Pines dataset. (a) Ground truth, (b) training set (10% of total pixels), and (c) classification with SaT (Chan et al. 2020) (98.83% overall accuracy)

Tight-Frame-Based Method for Images with Vascular Structures

The segmentation problem of branching tubular objects in 2D and 3D images arises in many applications, for example, extracting roads in aerial photography, and anatomical surfaces of blood vessels in medical images. Identifying tube-like structures is of great importance in medical imaging, with the primary application of segmenting blood vessels in magnetic resonance angiography (MRA) images. Unlike classical segmentation problems, vessel segmentation is characterized by different aims such as the following: (a) detect correctly branches and complex topologies, (b) detect vessels of very different thickness (from very thin to very thick), (c) repair small occlusions (false disconnections), (d) remove noise incorrectly segmented, and (e) control the minimum thickness of the vessels by a user-given precision. Moreover, when used in a real-time medical environment, automatic, robust, and efficient methods are essential. All these requirements make the vessel segmentation problem very challenging.

Many different approaches for image segmentation and, in particular, vessel segmentation have been proposed in the literature; see, for example, Chapman et al. (2004), Chen and Amini (2004), Dong et al. (2010), Franchini et al. (2010), Gooya et al. (2008), Krissian et al. (2000), Lorigo et al. (2001), Sum and Cheung (2008), Yan and Kassim (2006), Zonoobi et al. (2009) and the extended reviews Cremers et al. (2007) and Kirbas and Quek (2004). Below we give a brief account of some of these methods.

In Cai et al. (2011, 2013a), a tight-frame-based method was proposed to automatically identify tube-like structures in medical imaging, with the primary application of segmenting blood vessels in magnetic resonance angiography images. The method iteratively refines a region that encloses the potential boundary of the vessels. At each iteration, the tight-frame algorithm was applied to denoise and smooth the potential boundary and sharpen the region, in a similar fashion as the SaT strategy. The cost per iteration is proportional to the number of pixels in the image. It is proved that the iteration converges in a finite number of steps to a binary image whereby the segmentation of the vessels can be done straightforwardly.

Let $\mathbf{f} = \text{vec}(f)$ denote the vector obtained by concatenating the columns of f . It is worth mentioning the tight-frame algorithms used in, e.g., Cai et al. (2008) can be presented in the following generic form:

$$\mathbf{f}^{(i+\frac{1}{2})} = \mathcal{U}(\mathbf{f}^{(i)}), \quad (15)$$

$$\mathbf{f}^{(i+1)} = \mathcal{A}^T \mathcal{T}_\lambda(\mathcal{A} \mathbf{f}^{(i+\frac{1}{2})}), \quad i = 1, 2, \dots \quad (16)$$

Here $\mathbf{f}^{(i)}$ is an approximate solution at the i th iteration, \mathcal{U} is a problem-dependent operator, and $\mathcal{T}_\lambda(\cdot)$ is the soft-thresholding operator defined as follows. Given vectors $\mathbf{v} = [v_1, \dots, v_n]^T$ and $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_n]^T$, $\mathcal{T}_\lambda(\mathbf{v}) \equiv [t_{\lambda_1}(v_1), \dots, t_{\lambda_n}(v_n)]^T$, where

$$t_{\lambda_k}(v_k) \equiv \begin{cases} \text{sgn}(v_k)(|v_k| - \lambda_k), & \text{if } |v_k| > \lambda_k, \\ 0, & \text{if } |v_k| \leq \lambda_k. \end{cases} \quad (17)$$

Let $P^{(i+1)}$ be the diagonal matrix where the diagonal entry is 1 if the corresponding index is in $\Lambda^{(i+1)}$ and 0 otherwise. Then

$$\mathbf{f}^{(i+1)} \equiv (I - P^{(i+1)})\mathbf{f}^{(i+\frac{1}{2})} + P^{(i+1)}\mathcal{A}^T \mathcal{F}_{\lambda}(\mathcal{A}\mathbf{f}^{(i+\frac{1}{2})}). \quad (18)$$

By reordering the entries of the vector $\mathbf{f}^{(i+1)}$ into columns, we obtain the image $f^{(i+1)}$. We remark that the effect of (18) is to denoise and smooth the image on $\Lambda^{(i+1)}$.

Figures 8 and 9 give examples which show the great performance of the tight-frame-based method (Cai et al. 2013a) for images with tube-like structures. For more detail, please refer to Cai et al. (2013a).

Wavelet-Based Segmentation Method for Spherical Images

Spherical images are common in nature, for example, in cosmology (McEwen et al. 2007b), astrophysics (Schmitt et al. 2012), planetary science (Audet 2014),

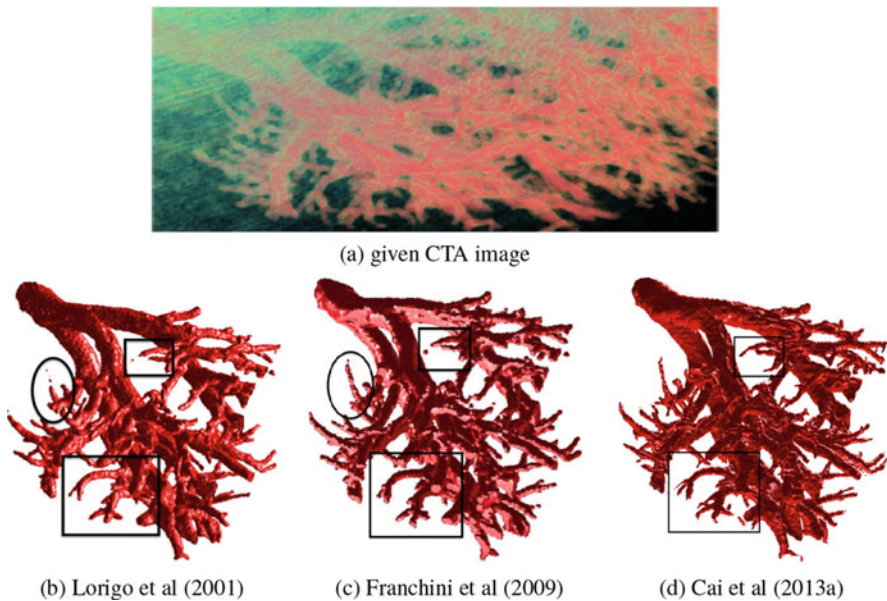


Fig. 8 Segmentation of the kidney volume dataset. (a) Given CTA image; (b) CURVES segmentation (Lorigo et al. 2001); (c) ADA segmentation (Franchini et al. 2009); (d) tight-frame-based method (Cai et al. 2013a)

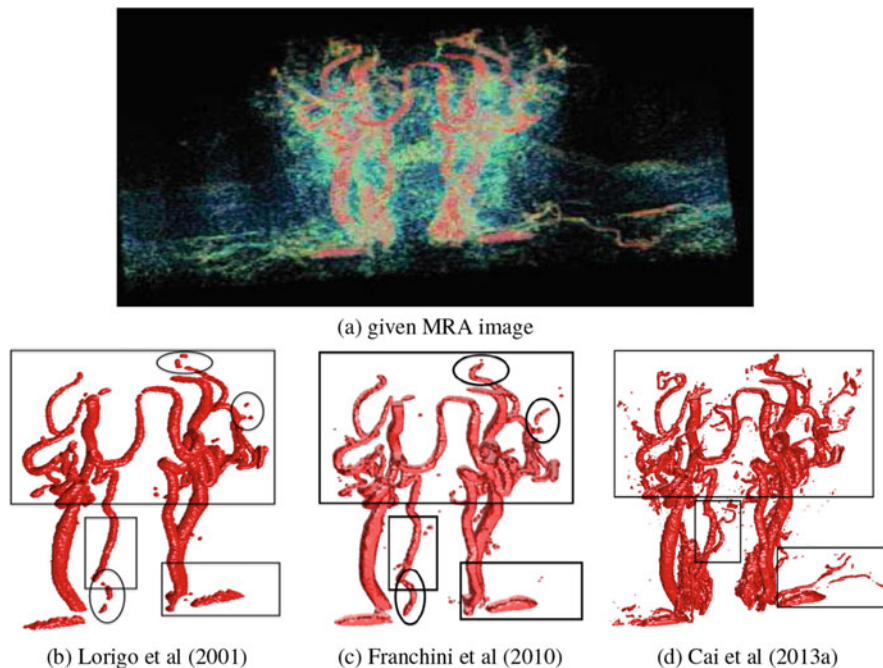


Fig. 9 Segmentation of the brain volume dataset. (a) Given MRA image; (b) CURVES segmentation (Lorigo et al. 2001); (c) ADA segmentation (Franchini et al. 2010); (d) tight-frame-based method (Cai et al. 2013a)

geophysics (Simons et al. 2011), and neuroscience (Rathi et al. 2011), where images are naturally defined on the sphere. Clearly, images defined on the sphere are different to Euclidean images in 2D and 3D in terms of symmetries, coordinate systems, and metrics constructed (see, e.g., Li and Hai 2010).

Wavelets have become a powerful analysis tool for spherical images, due to their ability to simultaneously extract both spectral and spatial information. A variety of wavelet frameworks have been constructed on the sphere in recent years, e.g., Baldi et al. (2009), McEwen et al. (2018), and have led to many insightful scientific studies in the fields mentioned above (see McEwen et al. 2007b, Schmitt et al. 2012, Audet 2014, Simons et al. 2011, Rathi et al. 2011). Different types of wavelets on the sphere have been designed to probe different structures in spherical images, for example, isotropic or directional and geometrical features, such as linear or curvilinear structures, to mention a few. Axisymmetric wavelets (Baldi et al. 2009; Leistedt et al. 2013) are useful for probing spherical images with isotropic structure, directional wavelets (McEwen et al. 2018) for probing directional structure, ridgelets (Michailovich and Rathi 2010; Starck et al. 2006) for analyzing antipodal signals on the sphere, and curvelets (Starck et al. 2006; Chan et al. 2017) for studying highly anisotropic image content such as curve-like features (we refer to Candés and Donoho (2005) for the general definition

of Euclidean ridgelets and curvelets). Fast algorithms have been developed to compute exact forward and inverse wavelet transforms on the sphere for very large spherical images containing millions of pixels (McEwen et al. 2007a). Localization properties of wavelet constructions have also been studied in detail (McEwen et al. 2018), showing important quasi-exponential localization and asymptotic uncorrelation properties for certain wavelet constructions. An investigation into the use of axisymmetric and directional wavelets for sparse image reconstruction was performed recently in Wallis et al. (2017), showing excellent performance.

In Cai et al. (2020), a wavelet-based method was proposed to segment images on the sphere, accounting for the underlying geometry of spherical data. The method is a direct extension of the tight-frame-based segmentation method (Cai et al. 2011, 2013a) used to automatically identify tube-like structures such as blood vessels in medical imaging. It is compatible with any arbitrary type of wavelet frame defined on the sphere, such as axisymmetric wavelets, directional wavelets, curvelets, and hybrid wavelet constructions. Such an approach allows the desirable properties of wavelets to be naturally inherited in the segmentation process. In particular, directional wavelets and curvelets, which were designed to efficiently capture directional signal content, provide additional advantages in segmenting images containing prominent directional and curvilinear features.

Figure 10 gives an example which shows the great performance of the wavelet-based segmentation method for spherical images. For more detail, please refer to Cai et al. (2020).

Three-Stage Method for Images with Intensity Inhomogeneity

The intensity inhomogeneity is a common phenomenon in real-world images and may bring considerable difficulties for image segmentation (Li et al. 2008). The intensity inhomogeneity can be roughly divided into two types: the extrinsic one and the intrinsic one. The extrinsic intensity inhomogeneity is globally revoked by the image acquisition devices or illumination variations which frequently appear in medical images. On the other hand, the intrinsic one is caused by the local discrepancy of the image color, intensity, or texture pattern in objects and backgrounds which usually appear in natural images.

The extrinsic inhomogeneous intensities are usually smoothly varying. Involving the local intensity information in the energy functional is a common way to address the issue of extrinsic inhomogeneity. Li et al. (2008) and Wang et al. (2010) used Gaussian kernel methods to characterize the intensities in local regions. The intrinsic intensity inhomogeneity varies sharply. Some texture segmentation algorithms (e.g., Brox et al. 2010 and Cremers et al. 2007) have been proposed to tackle such kinds of intensity inhomogeneity. New features (e.g., structure tensors (Ge et al. 2015), salient information (Kim and Kim 2013)) were also designed to get the desired segmentation results. Zhi and Shen (2018) proposed a level set-based method by incorporating saliency information and image intensity as region external energy to

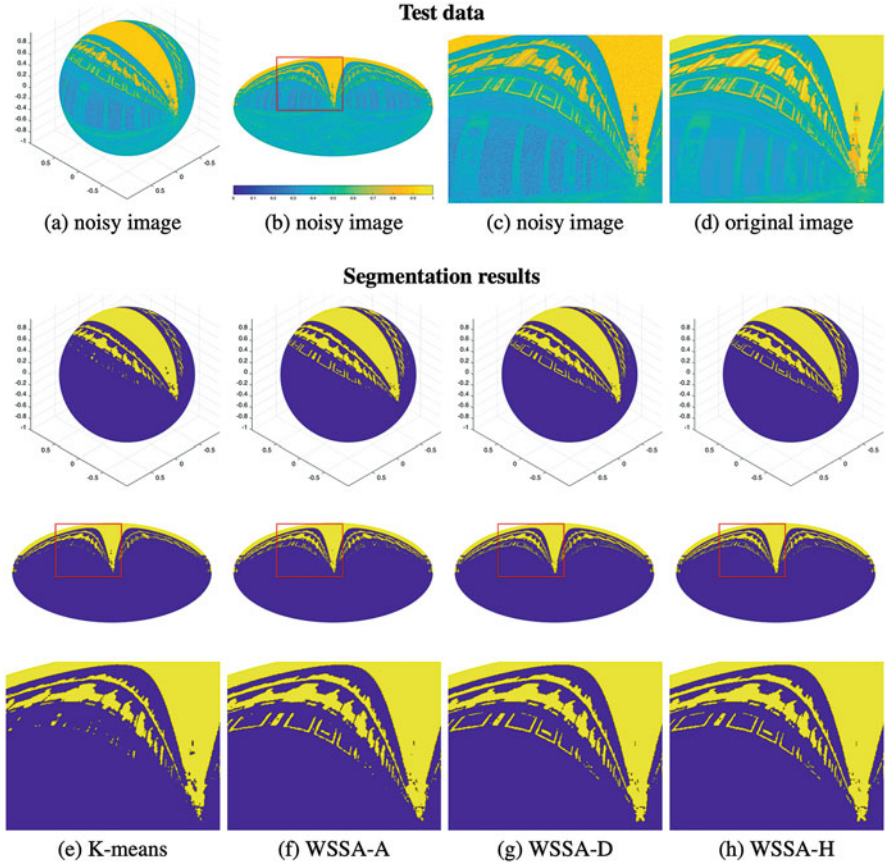


Fig. 10 Results of light probe image – the Uffizi Gallery. First row: noisy image shown on the sphere (a) and in 2D using a Mollweide projection (b) and the zoomed-in red rectangle area of the noisy (c) and original images (d), respectively; second to fourth rows from left to right: results of methods K-means (e), WSSA-A (f), WSSA-D (g) with $N = 6$ (even N), and WSSA-H (h), respectively. Note that methods WSSA-A, WSSA-D, and WSSA-H are the wavelet-based segmentation method (Cai et al. 2020), respectively, equipped with axisymmetric wavelets, directional wavelets, and hybrid wavelets defined on the sphere

motivate the curve evolution. These models can handle the intensity inhomogeneity to some extent.

In Li et al. (2020), a new three-stage segmentation framework was proposed based on the SaT method and the intensity inhomogeneity information of an image. The first stage in this framework is to perform a dimension lifting method. An intensity inhomogeneity image is added as an additional channel, which results in a vector-valued image. In the second stage, a SaT model is applied to each channel of the vector-valued image to obtain a smooth approximation. The semi-proximal alternating direction method of multipliers (sPADMM) (Han et al. 2018) is used to

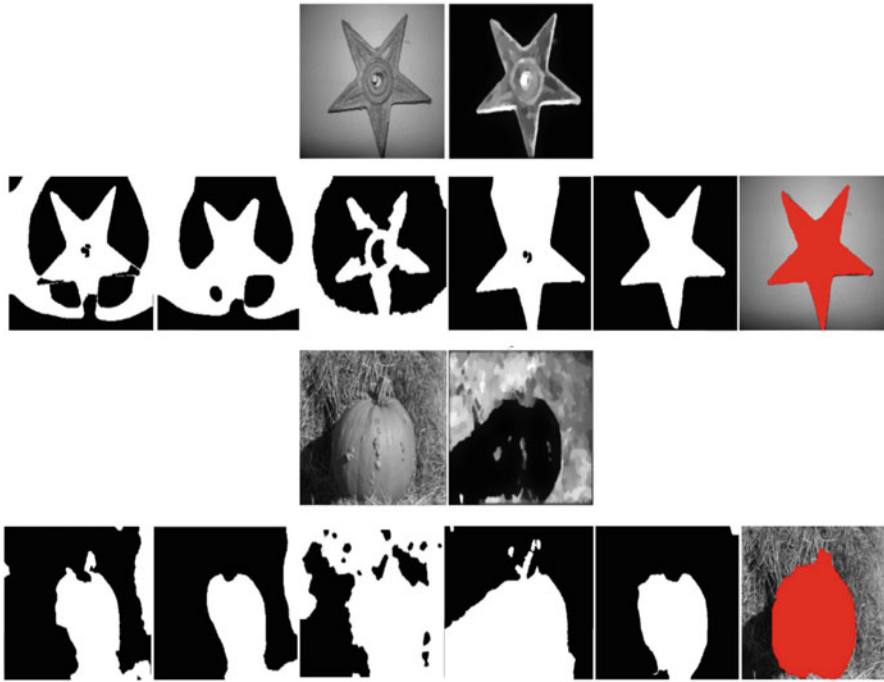


Fig. 11 Segmentation results on single-channel images. In the first and the third rows, the first column, images from the Alpert's dataset (size: 300×225) and, the second column, the corresponding intensity inhomogeneity images, respectively. In the second and the fourth rows, from the first column to the last column, segmentation results of the methods in Cai et al. (2017), Li et al. (2010, 2020), Zhi and Shen (2018), Wang et al. (2009) and the ground truth

solve this model, and it is proved that the sPADMM for solving this convex model has Q-linear convergence rate. In the last stage, a thresholding method is applied to the smoothed vector-valued image to get the final segmentation.

Figure 11 shows the great performance of the three-stage method (Li et al. 2020) incorporating intensity inhomogeneity information, and Fig. 12 demonstrates that Li et al. (2020) provides the most accurate segmentation results in comparison with five state-of-the-art methods including a deep learning approach (U-net method) (Ronneberger et al. 2015). For more detail, please refer to Li et al. (2020).

Conclusions

In this article, we introduced the SaT (smoothing and thresholding) segmentation methodology and methods developed based on this methodology with many applications in image processing. The SaT method provides an efficient and flexible methodology for image segmentations. It is easy to adapt the SaT method for

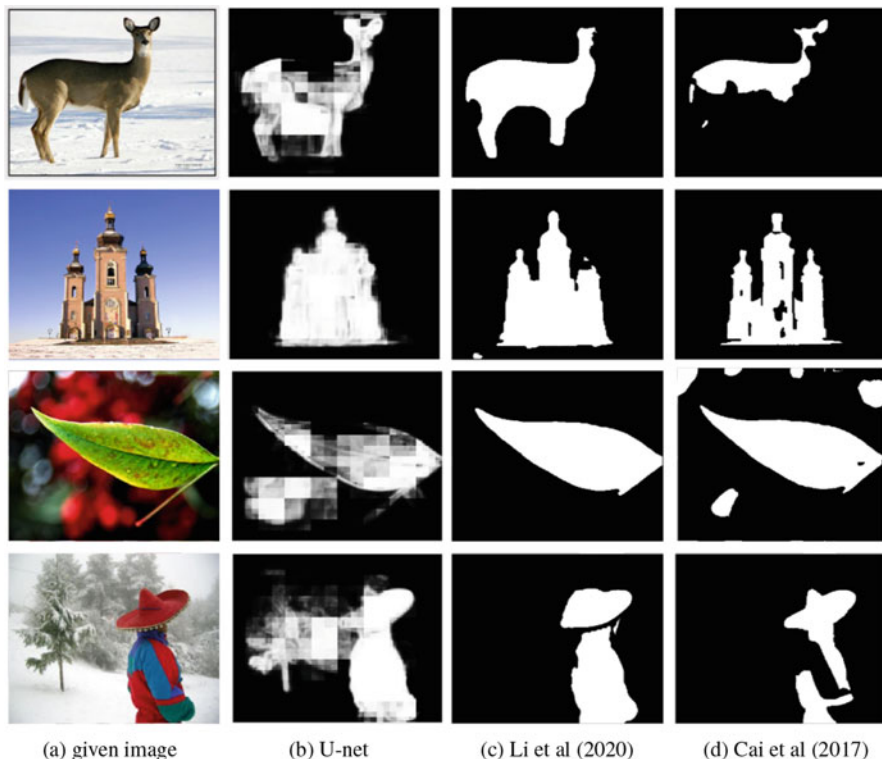


Fig. 12 Column (a), the original images from the 100 test dataset; column (b), segmentation results of the U-net method (Ronneberger et al. 2015); column (c), segmentation results of the method in Li et al. (2020); and column (d), segmentation results of the method in Cai et al. (2017)

various segmentation tasks. The SaT approach connects the segmentation problem to image restoration problem. Recent researches show that the SaT method can also be applied to classification problems. We hope that, with this article, the SaT method can reach audiences from broader areas and can inspire more cross-disciplinary researches.

Acknowledgments Supported in part by HKRGC Grants No. CUHK14306316, CUHK14301718, CityU11301120, CityU Grant 9380101, CRF Grant C1007-15G, AoE/M-05/12.

References

- Ambrosio, L., Tortorelli, V.: Approximation of functions depending on jumps by elliptic functionals via t -convergence. *Commun. Pure Appl. Math.* **43**, 999–1036 (1990)
- Aubert, G., Aujol, J.: A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.* **68**, 925–946 (2008)

- Audet, P.: Toward mapping the effective elastic thickness of planetary lithospheres from a spherical wavelet analysis of gravity and topography. *Phys. Earth Planet Inter.* **226**, 48–82 (2014)
- Baldi, P., Kerkyacharian, G., Marinucci, D., Picard, D.: Asymptotics for spherical needlets. *Ann. Stat.* **37**(3), 1150–1171 (2009)
- Bar, L., Chan, T., Chung, G., Jung, M., Kiryati, N., Mohieddine, R., Sochen, N., Vese, L.: Mumford and shah model and its applications to image segmentation and image restoration. In: *Handbook of Mathematical Imaging*, pp. 1095–1157. Springer, New York (2011)
- Bauer, B., Cai, X., Peth, S., Schladitz, K., Steidl, G.: Variational-based segmentation of biopores in tomographic images. *Comput. Geosci.* **98**, 1–8 (2017)
- Bellettini, G., Paolini, M., Verdi, C.: Convex approximations of functionals with curvature. *Math. Appl.* **2**(4), 297–306 (1991)
- Benninghoff, H., Garcke, H.: Efficient image segmentation and restoration using parametric curve evolution with junctions and topology changes. *SIAM J. Imag. Sci.* **7**(3), 1451–1483 (2014)
- Blake, A., Zisserman, A.: *Visual Reconstruction*. MIT Press, Cambridge, MA (1987)
- Bresson, X., Esedoglu, S., Vanderghynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imag. Vis.* **28**(2), 151–167 (2007)
- Brown, E., Chan, T., Bresson, X.: Completely convex formulation of the chan-vese image segmentation model. *Int. J. Comput. Vis.* **98**, 103–121 (2012)
- Brox, T., Rousson, M., Deriche, R., Weickert, J.: Colour, texture, and motion in level set based segmentation and tracking. *Image Vis. Comput.* **28**, 376–390 (2010)
- Cai, X.: Variational image segmentation model coupled with image restoration achievements. *Pattern Recogn.* **48**(6), 2029–2042 (2015)
- Cai, X., Steidl, G.: Multiclass segmentation by iterated ROF thresholding. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 237–250. Springer, Berlin/Heidelberg (2013)
- Cai, J., Chan, R., Shen, Z.: A framelet-based image inpainting algorithm. *Appl. Comput. Harmon. Anal.* **24**, 131–149 (2008)
- Cai, X., Chan, R., Morigi, S., Sgallari, F.: Framelet-based algorithm for segmentation of tubular structures. In: *SSVM. LNCS6667*. Springer (2011)
- Cai, X., Chan, R., Morigi, S., Sgallari, F.: Vessel segmentation in medical imaging using a tight-frame based algorithm. *SIAM J. Imag. Sci.* **6**(1), 464–486 (2013a)
- Cai, X., Chan, R., Zeng, T.: A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding. *SIAM J. Imag. Sci.* **6**(1), 368–390 (2013b)
- Cai, X., Fitschen, J., Nikolova, M., Steidl, G., Storath, M.: Disparity and optical flow partitioning using extended potts priors. *Inf. Inference J. IMA* **4**, 43–62 (2015)
- Cai, X., Chan, R., Nikolova, M., Zeng, T.: A three-stage approach for segmenting degraded color images: smoothing, lifting and thresholding (SLaT). *J. Sci. Comput.* **72**(3), 1313–1332 (2017). <https://doi.org/10.1007/s10915-017-0402-2>
- Cai, X., Chan, R.H., Schönlieb, C.B., Steidl, G., Zeng, T.: Linkage between piecewise constant Mumford–Shah model and Rudin–Osher–Fatemi model and its virtue in image segmentation. *SIAM J. Sci. Comput.* **41**(6), B1310–B1340 (2019)
- Cai, X., Wallis, C.G.R., Chan, J.Y.H., McEwen, J.D.: Wavelet-based segmentation on the sphere. *Pattern Recogn.* **100** (2020). <https://doi.org/10.1016/j.patcog.2019.107.081>
- Candés, E., Donoho, D.: Continuous curvelet transform: II. Discretization and frames. *Appl. Comput. Harmon. Anal.* **19**(2), 198–222 (2005)
- Cardelino, J., Caselles, V., Bertalmio, M., Randall, G.: A contrario selection of optimal partitions for image segmentation. *SIAM J. Imag. Sci.* **6**(3), 1274–1317 (2013)
- Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) *Energy Minimization Methods in Computer Vision and Pattern Recognition – EMMCVPR 2005. Lecture Notes in Computer Science*, vol. 3757, pp. 136–152. Springer, Berlin (2005)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.* **40**(1), 120–145 (2011)

- Chambolle, A., Caselles, V., Novaga, M., Cremers, D., Pock, T.: An introduction to total variation for image analysis. *Theor. Found. Numer. Methods Sparse Recover. Radon Ser. Comput. Appl. Math.* **9**, 263–340 (2010)
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chan, T.F., Sandberg, B.Y., Vese, L.A.: Active contours without edges for vector-valued images. *J. Vis. Commun. Image Represent.* **11**(2), 130–141 (2000)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006a)
- Chan, T.F., Esedoglu, S., Park, F., Yip, A.: Total variation image restoration: overview and recent developments. In: *Handbook of Mathematical Models in Computer Vision*, pp. 17–31. Springer, New York (2006b)
- Chan, R., Yang, H., Zeng, T.: A two-stage image segmentation method for blurry images with Poisson or multiplicative Gamma noise. *SIAM J. Imag. Sci.* **7**(1), 98–127 (2014)
- Chan, J., Leistedt, B., Kitching, T., McEwen, J.D.: Second-generation curvelets on the sphere. *IEEE Trans. Sig. Proc.* **65**(1), 5–14 (2017)
- Chan, R., Yang, H., Zeng, T.: Total Variation and Tight Frame Image Segmentation with Intensity Inhomogeneity (2019). arXiv e-prints arXiv:1904.01760
- Chan, R., Kan, K.K., Nikolova, M., Plemmons, R.J.: A two-stage method for spectral-spatial classification of hyperspectral images. *J. Math. Imag. Vis.* **62**, 790–807 (2020)
- Chapman, B., Parker, D., Stapelton, J., Parker, D.: Intracranial vessel segmentation from time-of-flight mra using pre-processing of the mip z-buffer: accuracy of the ZBS algorithm. *Med. Image Anal.* **8**(2), 113–126 (2004)
- Chen, J., Amini, A.: Quantifying 3d vascular structures in mra images using hybrid pde and geometric deformable models. *IEEE Trans. Med. Imag.* **23**(10), 1251–1262 (2004)
- Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *Int. J. Comput. Vis.* **72**(2), 195–215 (2007)
- Datt, B., McVicar, T., Van Niel, T., Jupp, D., Pearlman, J.: Preprocessing eo-1 hyperion hyperspectral data to support the application of agricultural indexes. *IEEE Trans. Geosci. Remote Sens.* **41**(6), 1246–1259 (2003)
- Dong, B., Chien, A., Shen, Z.: Frame based segmentation for medical images. *Commun. Math. Sci.* **32**, 1724–1739 (2010)
- Dong, B., Chien, A., Shen, Z.: Frame based segmentation for medical images. *Commun. Math. Sci.* **9**(2), 551–559 (2011)
- Durand, S., Fadili, J., Nikolova, M.: Multiplicative noise removal using l1 fidelity on frame coefficients. *J. Math. Imag. Vis.* **38**, 201–226 (2010)
- Eismann, M., Stocker, A., Nasrabadi, N.: Automated hyperspectral cueing for civilian search and rescue. *Proc. IEEE* **97**(6), 1031–1055 (2009)
- Fauvel, M., Tarabalka, Y., Benediktsson, J., Chanussot, J., Tilton, J.: Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **101**(3), 652–675 (2013)
- Franchini, E., Morigi, S., Sgallari, F.: Composed segmentation of tubular structures by an anisotropic pde model. In: Tai, X.-C., et al. (eds.) *SSVM 2009*. LNCS5567, pp. 75–86 (2009)
- Franchini, E., Morigi, S., Sgallari, F.: Segmentation of 3D tubular structures by a PDE-based anisotropic diffusion model. In: Dæhlen, M., et al. (eds.) *MMCS 2008*. LNCS5862, pp. 224–241 (2010)
- Ge, Q., Liang, X., Wang, L., Zhang, Z., Wei, Z.: A hybrid active contour model with structured feature for image segmentation. *Sig. Process* **108**, 147–158 (2015)
- Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984)
- Goldstein, T., Osher, S.: The split Bregman method for l1-regularized problems. *SIAM J. Imag. Sci.* **2**(2), 323–343 (2009)
- Gooya, A., Liao, H., et al.: A variational method for geometric regularization of vascular segmentation in medical images. *IEEE Trans. Image Process.* **17**(8), 1295–1312 (2008)

- Gowen, A., O'Donnell, C., Cullen, P., Downey, G., Frias, J.: Hyperspectral imaging-an emerging process analytical tool for food quality and safety control. *Trends Food Sci. Technol.* **18**(12), 590–598 (2007)
- Han, D., Sun, D., Zhang, L.: Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Math. Oper. Res.* **43**(2), 622–637 (2018)
- He, Y., Hussaini, M.Y., Ma, J., Shafei, B., Steidl, G.: A new fuzzy c-means method with total variation regularization for image segmentation of images with noisy and incomplete data. *Pattern Recogn.* **45**, 3463–3471 (2012)
- Hörig, B., Kühn, F., Oschütz, F., Lehmann, F.: Hymap hyperspectral remote sensing to detect hydrocarbons. *Int. J. Remote Sens.* **22**(8), 1413–1422 (2001)
- Jung, Y.M., Kang, S.H., Shen, J.: Multiphase image segmentation via Modica-Mortola phase transition. *SIAM J. Appl. Math.* **67**(5), 1213–1232 (2007)
- Kay, D., Tomasi, A., et al.: Color image segmentation by the vector-valued Allen–Cahn phase-field model: a multigrid solution. *IEEE Trans. Image Process.* **18**(10), 2330–2339 (2009)
- Kim, W., Kim, C.: Active contours driven by the salient edge energy model. *IEEE Trans. Image Process.* **22**, 1667–1673 (2013)
- Kirbas, C., Quek, F.: A review of vessel extraction techniques and algorithms. *CV Comput. Surv.* **36**, 81–121 (2004)
- Krissian, K., Malandain, G., Ayache, N., Vaillant, R., Troussset, Y.: Model-based detection of tubular structures in 3d images. *CVIU* **80**, 130–171 (2000)
- Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. *J. Math. Imag. Vis.* **27**, 257–263 (2007)
- Leistedt, B., McEwen, J., Vanderghynst, P., Wiaux, Y.: S2let: a code to perform fast wavelet analysis on the sphere. *Astron. Astrophys.* **558**(A128), 1–9 (2013)
- Lellmann, J., Schnörr, C.: Continuous multiclass labeling approaches and algorithms. *SIAM J. Imag. Sci.* **44**(4), 1049–1096 (2011)
- Li, S., Hai, Y.: A full-view spherical image format. In: *ICPR*, pp. 2337–2340 (2010)
- Li, C., Kao, C., Gore, J., Ding, Z.: Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.* **17**, 1940–1949 (2008)
- Li, F., Ng, M., Zeng, T., Shen, C.: A multiphase image segmentation method based on fuzzy region competition. *SIAM J. Imag. Sci.* **3**(2), 277–299 (2010)
- Li, X., Yang, X., Zeng, T.: A three-stage variational image segmentation framework incorporating intensity inhomogeneity information. *SIAM J. Imag. Sci.* **13**(3), 1692–1715 (2020)
- Lorigo, L., Faugeras, O., Grimson, E., et al.: Curves: curve evolution for vessel segmentation. *Med. Image Anal.* **5**, 195–206 (2001)
- Manolakis, D., Shaw, G.: Detection algorithms for hyperspectral imaging applications. *IEEE Sig. Process. Mag.* **19**(1), 29–43 (2002)
- Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *ICCV*, vol. 2, pp. 416–423 (2001)
- McEwen, J., Hobson, M., Mortlock, D., Lasenby, A.: Fast directional continuous spherical wavelet transform algorithms. *IEEE Trans. Sig. Process.* **55**(2), 520–529 (2007a)
- McEwen, J., Vielva, P., Wiaux, Y., et al.: Cosmological applications of a wavelet analysis on the sphere. *J. Fourier Anal. Appl.* **13**(4), 495–510 (2007b)
- McEwen, J., Durastanti, C., Wiaux, Y.: Localisation of directional scale-discretised wavelets on the sphere. *Appl. Comput. Harm. Anal.* **44**(1), 59–88 (2018)
- Michailovich, O., Rathi, Y.: On approximation of orientation distributions by means of spherical ridgelets. *IEEE Trans. Sig. Proc.* **19**(2), 461–477 (2010)
- Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **XLII**, 577–685 (1989)
- Paschos, G.: Perceptually uniform color spaces for color texture analysis: an empirical evaluation. *IEEE Trans. Image Process.* **10**(6), 932–937 (2001)
- Patel, N., Patnaik, C., Dutta, S., Shekh, A., Dave, A.: Study of crop growth parameters using airborne imaging spectrometer data. *Int. J. Remote Sens.* **22**(12), 2401–2411 (2001)

- Pock, T., Chambolle, A., Cremers, D., Bischof, H.: A convex relaxation approach for computing minimal partitions. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE, pp. 810–817 (2009a)
- Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth mumford-shah functional. In: ICCV (2009b)
- Rathi, Y., Michailovich, O., Setsompop, K., et al.: Sparse multi-shell diffusion imaging. MICCAI, Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. **14**(2), 58–65 (2011)
- Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
- Rotaru, C., Graf, T., Zhang, J.: Color image segmentation in HSI space for automotive applications. J. Real-Time Image Process. **3**(4), 311–322 (2008)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
- Sandberg, B., Kang, S., Chan, T.: Unsupervised multiphase segmentation: a phase balancing model. IEEE Trans. Image Process. **19**, 119–130 (2010)
- Schmitt, J., Starck, J., Casandjian, J., Fadili, J., Grenier, I.: Multichannel Poisson denoising and deconvolution on the sphere: application to the Fermi Gamma-ray Space Telescope. *Astron. Astrophys.* **546**(A114) (2012). https://www.aanda.org/articles/aa/full_html/2012/10/aa18234-11/aa18234-11.html
- Shi, J., Osher, S.: A nonlinear inverse scale space method for a convex multiplicative noise model. *SIAM J. Imag. Sci.* **1**, 294–321 (2008)
- Simons, F., Loris, I., Nolet, G., et al.: Solving or resolving global tomographic models with spherical wavelets, and the scale and sparsity of seismic heterogeneity. *Geophys. J. Int.* **187**, 969–988 (2011)
- Starck, J., Moudou, Y., Abrial, P., Nguyen, M.: Wavelets, ridgelets and curvelets on the sphere. *Astron. Astrophys.* **446**(3), 1191–1204 (2006)
- Steidl, G., Teuber, T.: Removing multiplicative noise by Douglas-Rachford splitting methods. *J. Math. Imag. Vis.* **36**(2), 168–184 (2010)
- Stein, D., Beaven, S., Hoff, L., Winter, E., Schaum, A., Stocker, A.: Anomaly detection from hyperspectral imagery. *IEEE Sig. Process. Mag.* **19**(1), 58–69 (2002)
- Storath, M., Weinmann, A.: Fast partitioning of vector-valued images. *SIAM J. Imag. Sci.* **7**(3), 1826–1852 (2014)
- Sum, K., Cheung, P.: Vessel extraction under non-uniform illumination: a level set approach. *IEEE Trans. Biomed. Eng.* **55**(1), 358–360 (2008)
- Vese, L., Chan, T.: A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vis.* **50**(3), 271–293 (2002)
- Wallis, C., Wiaux, Y., McEwen, J.: Sparse image reconstruction on the sphere: analysis and synthesis. *IEEE Trans. Image Process.* **26**(11), 5176–5187 (2017)
- Wang, L., Li, C., Sun, Q., Xia, D., Kao, C.Y.: Active contours driven by local and global intensity fitting energy with application to brain MR image segmentation. *Comput. Med. Imag. Graph.* **33**(7), 520–531 (2009)
- Wang, X., Huang, D., Xu, H.: An efficient local Chan-Vese model for image segmentation. *Pattern Recogn.* **43**, 603–618 (2010)
- Wang, X., Tang, Y., Masnou, S., Chen, L.: A global/local affinity graph for image segmentation. *IEEE Trans. Image Process.* **24**(4), 1399–1411 (2015)
- Yan, P., Kassim, A.: MRA image segmentation with capillary geodesic active contours. *Med. Image Anal.* **10**, 317–329 (2006)
- Yuan, J., Bae, E., Tai, X.C.: A study on continuous max-flow and min-cut approaches. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2217–2224. IEEE (2010a)
- Yuan, J., Bae, E., Tai, X.C., Boykov, Y.: A continuous max-flow approach to potts model. In: European Conference on Computer Vision, pp. 379–392 (2010b)

-
- Zach, C., Gallup, D., Frahm, J.-M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vision, Modeling, and Visualization Workshop (2008)
- Zhang, Y., Matuszewski, B., Shark, L., Moore, C.: Medical image segmentation using new hybrid level-set method. In: 2008 Fifth International Conference BioMedical Visualization: Information Visualization in Medical and Biomedical Informatics, pp. 71–76 (2008)
- Zhi, X., Shen, H.: Saliency driven region-edge-based top down level set evolution reveals the asynchronous focus in image segmentation. *Pattern Recogn.* **80**, 241–255 (2018)
- Zonoobi, D., Kassim, A., Shen, W.: Vasculature segmentation in mra images using gradient compensated geodesic active contours. *J. Sig. Process. Syst.* **54**, 171–181 (2009)



Recent Development of Medical Shape Analysis via Computational Quasi-conformal Geometry

41

Hei-Long Chan and Lok-Ming Lui

Contents

Introduction	1414
The Quasi-conformal Teichmüller Theory	1414
Conformal Mappings	1414
Quasi-conformal Mappings	1415
Teichmüller Mappings	1417
Medical Image Segmentation and Registration by Quasi-conformal Theory	1418
Image Segmentation	1419
Image Registration and Fusion	1422
Other Imaging Applications	1423
Surface Analysis for Medical Applications	1424
3D Surface Registration	1424
High-Dimensional Shape Deformation	1427
Disease Diagnosis and Classification by Quasi-conformal Geometry	1430
Classification of the Alzheimer's Disease	1430
Other Classification Model	1432
Conclusion	1433
References	1435

Abstract

Medical analysis is closely related to mathematics in many aspects. Over the past decades, mathematicians have designed numerous mathematical models and algorithms to aid medical researches. However, the space for joint-forcing mathematics with the medical industry is very limited in early years due to immature implementation and technological support. Those models are mostly limited to simple applications of the probability and statistics theory. It is until

H.-L. Chan · L.-M. Lui (✉)
Chinese University of Hong Kong, Hong Kong, China
e-mail: hlchan@math.cuhk.edu.hk; lmlui@math.cuhk.edu.hk

recent years when computational geometry comes into appliance, and it opens up a huge room for the incorporation of mathematics with medical analysis. For instance, medical imaging, geometric modeling for medical surfaces, and machine learning for disease classification are crucial topics nowadays having heavy reliance on image processing and geometric analysis. There are many streams in applying the study of geometry. Among those, the application of the quasi-conformal Teichmüller theory has shown to be very successful in recent years. This article serves to conclude some most updated models having solid contributions to the medical science in different aspects.

Keywords

Shape analysis · Quasi-conformal geometry · Computational geometry · Medical imaging · Disease classification

Introduction

This is an expository article aiming at introducing the most updated mathematical models incorporating the quasi-conformal (QC) Teichmüller theory with the medical science. The article concerns the applications of the QC theory on different medical aspects such as medical imaging, medical surface analysis, disease diagnosis, etc. Every model involved will just be discussed in brief by a short paragraph only. Readers should refer to the corresponding paper if they find any interest in understanding the whole formulation or analysis of the models.

In the following sections, we will include some backgrounds of the quasi-conformal Teichmüller theory at first. The applications of the theory in different medical aspects will be discussed in the later sections.

The Quasi-conformal Teichmüller Theory

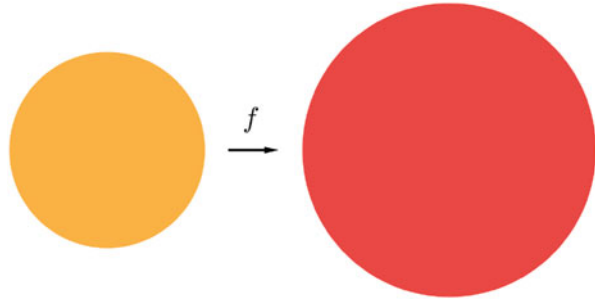
Firstly, we will briefly introduce the quasi-conformal Teichmüller theory.

Conformal Mappings

Suppose $f : M \rightarrow N$ is a diffeomorphism from a surface M to another surface N . Denote by TM , the tangent bundle of M and N , respectively. Let $df : TM \rightarrow TN$ be the differential of f . Under this setting, f is defined to be conformal if there exists a smooth function $\lambda : M \rightarrow \mathbb{R}$ such that for any $x \in M$ and for any $u, v \in T_x M$, the mapping $d_x f : T_x M \rightarrow T_{f(x)} N$ satisfies

$$\langle d_x f(u), d_x f(v) \rangle = \lambda(x) \langle u, v \rangle. \quad (1)$$

Fig. 1 Illustration of a conformal mapping that maps an infinitesimal disk into another infinitesimal disk



Here the smooth function λ is called the conformal factor of the conformal mapping f .

From the definition, we can see that a conformal mapping preserves the surface metric on M up to the conformal factor as a multiplying factor. Infinitesimally, a conformal mapping f maps a disk into another disk, as illustrated by Fig. 1.

Equivalently, conformal mapping can be defined as a diffeomorphism $f : M \rightarrow N$ satisfying the Cauchy-Riemann equation:

$$\frac{\partial f}{\partial \bar{z}} = 0. \tag{2}$$

where $\frac{\partial}{\partial \bar{z}} = \frac{\partial}{\partial x} + i \frac{\partial}{\partial y}$. While the former one provides a more straightforward understanding on the local geometry preserving property of conformal mappings, the latter one is the more convenient definition for us to generalize the notion of conformal mappings into quasi-conformal mappings.

Quasi-conformal Mappings

Suppose M, N are the same surfaces as above, and let $f : M \rightarrow N$ be a mapping having continuous partial derivative. We say that f is quasi-conformal if it follows the Beltrami equation:

$$\frac{\partial f}{\partial \bar{z}} = \mu \cdot \frac{\partial f}{\partial z}, \tag{3}$$

where $\frac{\partial}{\partial \bar{z}} = \frac{\partial}{\partial x} - i \frac{\partial}{\partial y}$. Here, $\mu : M \rightarrow \mathbb{C}$ is a Lebesgue measurable complex function and satisfies

$$\|\mu\|_\infty < 1. \tag{4}$$

The function μ defined in equation (3) is called the Beltrami coefficient associated to f . An immediate observation is that $\mu = 0$ if and only if f is conformal.

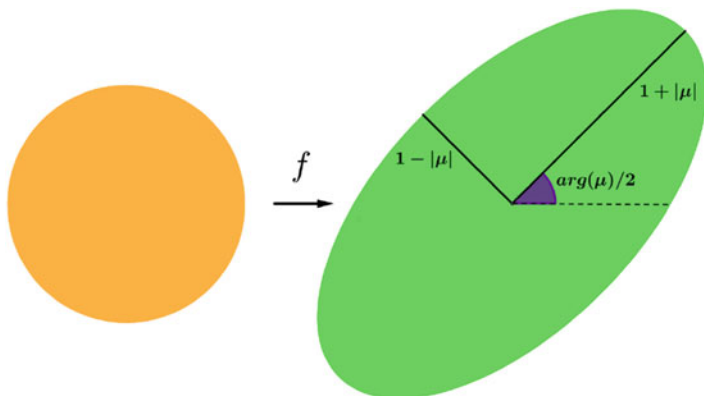


Fig. 2 Illustration of a quasi-conformal mapping that maps an infinitesimal disk into an infinitesimal ellipse

Therefore, conformal mapping is just a special case in the notion of quasi-conformal mappings. Intuitively, quasi-conformal mapping is a generalization of conformal mapping such that, instead of mapping infinitesimal disks to infinitesimal disks, a quasi-conformal mapping maps infinitesimal disks to infinitesimal ellipse. Figure 2 shows the illustration of the infinitesimal behavior of a quasi-conformal mapping.

Mathematically, for any $z \in Nbd(x, \delta)$ where $x \in M$ and $\delta > 0$ is small enough, a quasi-conformal mapping f has the local parametric expression:

$$f(z) \approx f(x) + f_z(x)z + f_{\bar{z}}(x)\bar{z} = f(x) + f_z(x)(z + \mu(x)\bar{z}) \tag{5}$$

Note that in Equation (5), the term $f(x)$ and the term $f_z(x)$ are just the translation term and the dilation term, respectively, which are both conformal. Therefore, the non-conformality of f is completely originated from the term $D(z) = z + \mu(x)\bar{z}$. Hence, analyzing the conformality of f can be simplified into the analysis of the Beltrami coefficient μ . Indeed, as for the infinitesimal behavior of a quasi-conformal mapping f , the angle of maximum magnification is $arg(\mu(x))/2$ with magnifying factor being $1 + |\mu(x)|$ while the angle of maximum contraction is $arg(\mu(x) - \pi)/2$ with contracting factor being $1 - |\mu(x)|$. Therefore, there is a very close relationship between a quasi-conformal mapping f and its associated Beltrami coefficient μ .

One important relationship between f and μ is that, the diffeomorphic property of f can be totally replaced by a norm constraint on μ , as described by the following theorem:

Theorem 1. *Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be a complex mapping having continuous partial derivative. Define*

$$\mu = \frac{\partial f}{\partial \bar{z}} \bigg/ \frac{\partial f}{\partial z}, \tag{6}$$

then $\|\mu\|_\infty < 1$ implies f is an orientation preserving homeomorphism.

Therefore, a quasi-conformal mapping f must be an orientation preserving homeomorphism. Another important relationship between f and μ is stated by the following theorem:

Theorem 2 (Measurable Riemann Mapping Theorem). *Suppose $\mu : \mathbb{C} \rightarrow \mathbb{C}$ is Lebesgue measurable satisfying $\|\mu\|_\infty < 1$ and then there exists a quasi-conformal homeomorphism f from the unit disk to itself, which is in the Sobolev space $W^{1,2}(\mathbb{C})$ and satisfies the Beltrami equation in the distribution sense. Furthermore, assuming the mapping is stationary at 0, 1, and ∞ , the associated quasi-conformal mapping f is uniquely determined.*

From the Beltrami Equation (3) and the measurable Riemann mapping theorem, there is a one-to-one correspondence between f and μ under suitable normalization. In other words, most constraints on a mapping f can be regarded as constraints on the space of the corresponding Beltrami coefficient.

Concerning about the composition of quasi-conformal maps, if f and g are two quasi-conformal mappings associated with the Beltrami coefficients μ_f and μ_g , respectively, then the Beltrami coefficient of the composition mapping $g \circ f$ is given by

$$\mu_{g \circ f} = \frac{\mu_f + (\mu_g \circ f)\tau}{1 + \bar{\mu}_f(\mu_g \circ f)\tau}, \tag{7}$$

where $\tau = \bar{f}_z/f_z$.

Teichmüller Mappings

Teichmüller maps are quasi-conformal maps whose Beltrami coefficients have a constant norm. That is, a Teichmüller map has a uniform conformal distortion over the entire domain. Mathematically, the definition of Teichmüller map is:

Definition 1 (Teichmüller map). Let $f : S_1 \rightarrow S_2$ be a quasi-conformal map. f is said to be a Teichmüller map (T-map) associated with the quadratic differential $q = \varphi dz^2$ where $\varphi : S_1 \rightarrow \mathbb{C}$ is a holomorphic function if its associated Beltrami coefficient is of the form

$$\mu(f) = k \frac{\bar{\varphi}}{|\varphi|}, \tag{8}$$

for some constant $k < 1$ and quadratic differential $q \neq 0$ with $\|q\|_1 = \int_{S_1} |\varphi| < \infty$.

Teichmüller maps are closely related to a class of maps called extremal quasi-conformal maps, defined by:

Definition 2 (Extremal quasi-conformal map). Let $f : S_1 \rightarrow S_2$ be a quasi-conformal map. f is said to be an extremal quasi-conformal map if for any quasi-conformal map $h : S_1 \rightarrow S_2$ isotopic to f relative to the boundary, we have

$$K(f) \leq K(h), \quad (9)$$

where $K(f)$ is the maximal quasi-conformal dilation of f . It is uniquely extremal if the inequality (9) is strict when $h \neq f$.

The two concepts are connected by the following theorem:

Theorem 3 (Landmark-matching Teichmüller map). Let $g : \partial\mathbb{D} \rightarrow \partial\mathbb{D}$ be an orientation-preserving diffeomorphism of $\partial\mathbb{D}$, where \mathbb{D} is the unit disk. Suppose further that $g'(e^{i\theta}) \neq 0$ and $g''(e^{i\theta})$ is bounded. Let $\{l^k\}_{k=1}^n \in \mathbb{D}$ and $\{q^k\}_{k=1}^n \in \mathbb{D}$ be the corresponding interior landmark constraints. Then there exists a unique Teichmüller map $f : (\mathbb{D}, \{l^k\}_{k=1}^n) \rightarrow (\mathbb{D}, \{q^k\}_{k=1}^n)$ matching the interior landmarks, which is the unique extremal extension of g to \mathbb{D} . Here $(\mathbb{D}, \{l^k\}_{k=1}^n)$ denotes the unit disk \mathbb{D} with prescribed landmark points $\{l^k\}_{k=1}^n$.

Therefore, besides equipped with uniform conformal distortion, Teichmüller maps are extremal in the sense that they minimize the maximal quasi-conformal dilation. Furthermore, Teichmüller maps induce a natural metric, called the Teichmüller distance, which can be used to measure the difference between two shapes in terms of local geometric distortion.

Definition 3 (Teichmüller distance). For every i , let S_i be a Riemann surface with landmarks $\{p_i^k\}_{k=1}^n$. The Teichmüller distance between (f_i, S_i) and (f_j, S_j) is defined as

$$d_T((f_i, S_i), (f_j, S_j)) = \inf_{\varphi} \frac{1}{2} \log K(\varphi), \quad (10)$$

where $\varphi : S_i \rightarrow S_j$ varies over all quasi-conformal maps with $\{p_i^k\}_{k=1}^n$ corresponding to $\{p_j^k\}_{k=1}^n$, which is homotopic to $f_j^{-1} \circ f_i$, and K is the maximal quasi-conformal dilation.

Medical Image Segmentation and Registration by Quasi-conformal Theory

Medical imaging concerns the understanding of medical images, for instance, X-ray images, MR images, and CT images. In most applications, either important anatomical structures should be located and segmented from an image, or the correspondence between pairs of scanned images should be elaborated for further

medical analysis. These corresponds to the study of image segmentation and image registration in computational mathematics. In particular, the QC theory helps in building a convenient interface of the image processing models while generating consistent and accurate results.

Image Segmentation

Segmenting the relevant anatomical structures from a medical image is always a challenging task, especially in the case of occlusions due to inevitable manual and machine artifacts. Those artifacts may change the topology of the target organ or hinder parts of the boundary of it. On the one hand, traditional intensity-based segmentation algorithms usually fail to elude the occlusions. On the other hand, common shape-prior-based segmentation models are too restrictive to capture the subject's boundary by a rough template.

Quasi-conformal geometry finds a good application in dealing with image occlusions. According to the QC theory, a deformation mapping on the image domain can be described by the corresponding Beltrami coefficient. Therefore, by putting constraints on the corresponding Beltrami coefficient, a manual template object can be deformed diffeomorphically to capture the target object. Hence, the topology of the segmented region can be directly prescribed by the template object.

In (2018), Chan et al. proposed an image segmentation model by introducing a notion called the Beltrami representation of shapes. The Beltrami representation $\mathcal{B}_g(D)$ of a shape D (subset of the image domain Ω) is defined by

$$\mathcal{B}_g(D) = \mu \quad (11)$$

such that

$$f^\mu(\hat{D}_g) = D, \quad f^\mu = Id \text{ in } \mathbb{C} \setminus \Omega. \quad (12)$$

where μ is the Beltrami coefficient of the deformation mapping f^μ . The idea of the Beltrami representation is to make use of the one-to-one correspondence between a mapping and its Beltrami coefficient to implement the diffeomorphic deformation constraint on the space of Beltrami coefficients. The model restricts the deformation to be diffeomorphic without imposing further constraint on the deformation. As such, the template object adapts to the target subject with high flexibility while eluding any topological occlusion. In implementing the idea, Chan et al. proposed a variational model:

$$E(\mu) = \int_{\Omega} |\mu|^2 + \eta \int_{\Omega} (I \circ f^\mu - J)^2 + \lambda \int_{\Omega} |\nabla \mu|^2 + \sigma \int_{\Omega} (|u|^2 + |\nabla u|^2), \quad (13)$$

involving a diffeomorphic property constraining term, an intensity matching term and two deformation smoothing terms, respectively. As for the derivation, the

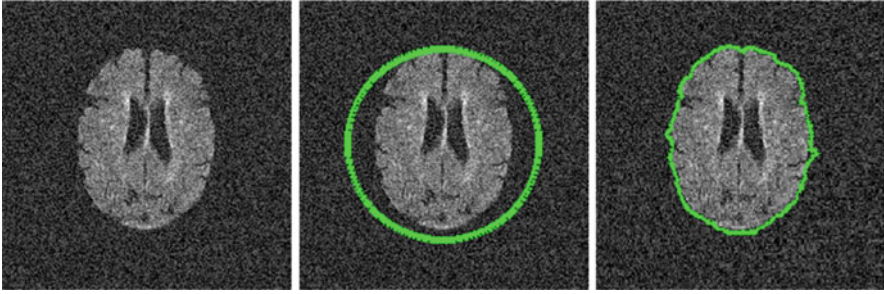


Fig. 3 An example of the segmentation by the proposed model extracted from Chan et al. (2018). Left: input image; middle: initial template object superimposed on the input image; right: segmentation result

existence of solution to the model, and the minimization process of the variational model (13), readers are referred to Chan et al. (2018).

Figure 3 demonstrates an example using the QC model to segment a noisy brain image. While the center part of the brain (the hippocampus) shows significantly different color which will usually be taken as occlusions by other segmentation models, the fuzzy noise on the image also adds extra difficulty in segmenting the brain as a whole. Nevertheless, the QC model still captures the brain with an accurate boundary. This demonstrates the effectiveness of applying the QC theory to prescribe the topology of the target region.

The application of the QC theory in medical image segmentation does not only lie on topology preservation. By discretizing the QC theory and joint-forcing the notion of dihedral angles on the meshed image domain, convexity can also be prescribed on particular portions on the segmented region. Therefore, the segmentation process can be much more adaptive to the given target subject, without relying too much on a given shape prior.

In Siu et al. (2020), Chan et al. advanced their topology preserving segmentation model to a convexity preserving segmentation model. They employed the notion of dihedral angle and implemented the QC segmentation model in a discrete setting. In particular, the dihedral angle plays the role to determine and constrain the convexity of the triangulated image domain. In advance, convexity can be constrained on just sectors of the template object. That is, the model enables the constraint of partial convexity on the segmented region. Given a portion $\Gamma \subset D$ at which the user wants to prescribe convexity on it, their QC segmentation model with partial convexity prior reads

$$\begin{aligned} \min_{\mu_V, \nu_V} E(\mu_V, \nu_V) = & \sum_{v \in V} |\nu_V|^2 + \eta \sum_{v \in V} (I \circ f_V^\mu - J)^2 + \lambda \sum_{v \in V} |\nabla \nu_V|^2 \\ & + \sigma \sum_{v \in V} (|u_V|^2 + |\nabla u_V|^2) + \delta \sum_{v \in V} |\nu_V - \mu_V|^2 \end{aligned} \quad (14)$$

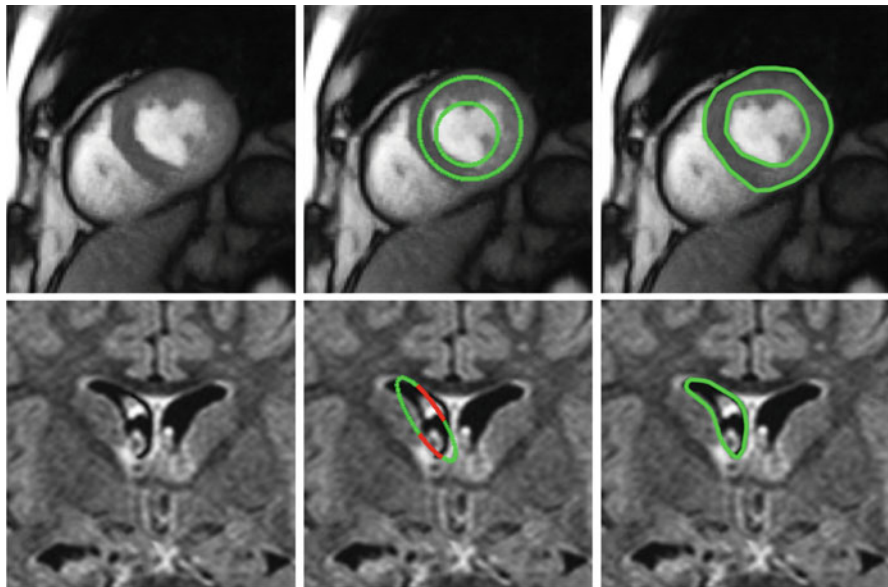


Fig. 4 Two examples of the segmentation by the proposed model extracted from Siu et al. (2020). Left: input image; middle: initial template object superimposed on the input image; right: segmentation result

$$\text{subject to } \sum_{e \in E_v} \theta_{f(e)} \geq (|F_v| - 1)\pi \text{ for all } v \in \Gamma, \text{ where } \Gamma \subset \partial D. \quad (15)$$

The model involves an alternating minimization of a discrete model, subjected to a convexity constraint on the portion $\Gamma \subset D$ based on the dihedral angles.

Partial convexity has a wide range of applications in medical imaging. Figure 4 demonstrates two applications. In particular, the first example demonstrates the segmentation of a genus-1 region (the vascular wall) under the full convexity setting. And the second example demonstrates the segmentation under the partial convexity setting, in which the occluded hippocampus is accurately segmented by the QC model. Further analysis and experiments can be found in Siu et al. (2020).

Deep learning is a popular tool to leverage in recent days. In (2020), Zhang et al. proposed to apply deep neural network on the quasi-conformal framework to segment brain tumor from images. They introduced a novel differential geometry-based quasi-conformal mapping augmentation technique to augment the brain tumor images. The method lets the user specify or randomly generate a complex-valued function on the image domain via Beltrami coefficient. By solving the Beltrami equation with given Beltrami coefficient, the quasi-conformal mapping, which can further guide the deformation of the image, is able to generate all possible linear and nonlinear image warpings, and it is flexible to allow the user to fully control the global and local deformations.

Image Registration and Fusion

Image registration is important in medical imaging in elaborating a meaningful correspondence between images for surface reconstruction and disease analysis. The fusion of images with different modality is challenging. For example, cross-platform nonrigid registration of CT with MR images is a crucial yet difficult task. Quasi-conformal theory finds a good application in this problem using similar concepts and implementations as in image segmentation. In Lam et al. (2014) and Lui et al. (2012), Lam et al. proposed a quasi-conformal registration model to handle image and surface registration with large deformations. In (2015), Lam et al. proposed a quasi-conformal hybrid multimodality registration model. Their strategy is to find the optimizer of an energy functional involving the Beltrami coefficients term and restrict the class of registration transformation to quasi-conformal mapping for the image fusion problem. The diffeomorphism associated to the optimized Beltrami coefficient will automatically satisfy the landmark constraints and maximize the mutual information between the source and target images. Their modeling of the registration mapping reads

$$f = \arg \min_g \text{Similar}(M \circ g), \quad g : M \rightarrow S \tag{16}$$

subject to

$$f \text{ is diffeomorphic,} \tag{17}$$

$$f(p_i) = q_i \quad i = 1, 2, \dots, m. \tag{18}$$

In other words, the desired mapping is a diffeomorphism that deforms the moving images M (Fig. 5a, e) to adapt to the static images S (Fig. 5b, f), while matching the landmark points p_i 's to q_i 's, respectively.

To obtain such a deformation mapping f , Lam et al. apply the quasi-conformal theory and formulate the variational model:

$$\begin{aligned}
 (\bar{\mu}, f) = \arg \min_{v, g} & \int_{\Omega} |\nabla v|^2 + \alpha \int_{\Omega} |v|^p \\
 & + \frac{1}{2} \left[\int_{\Omega} (S_T - M \circ g)^2 + \int_{\Omega} (S - M_T \circ g)^2 \right]
 \end{aligned} \tag{19}$$

subject to

$$\|\bar{\mu}\|_{\infty} < 1 \tag{20}$$

$$f(p_i) = q_i, \quad i = 1, 2, \dots, m, \tag{21}$$

$$\mu(f) = \bar{\mu}. \tag{22}$$

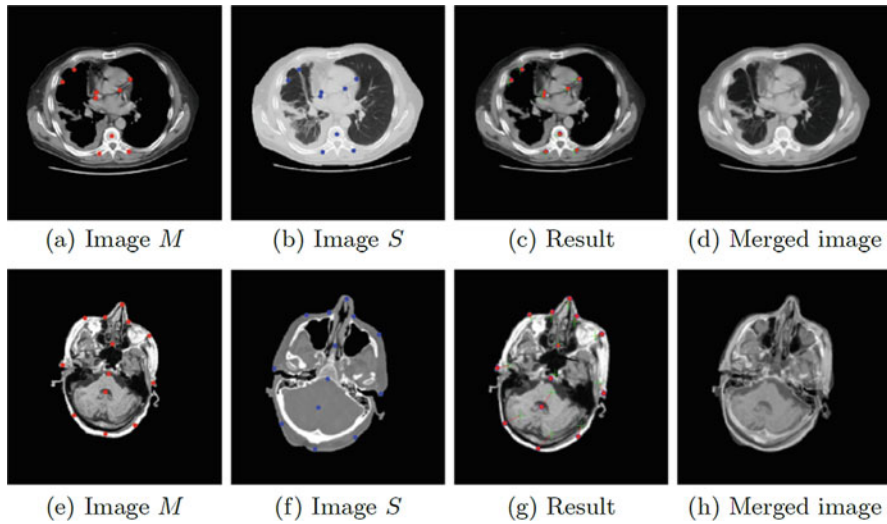


Fig. 5 Examples for medical image registration and fusion extracted from Lam and Lui (2015) (a) Image M (b) Image S (c) Result (d) Merged image (e) Image M (f) Image S (g) Result (h) Merged image

The model measures the similarity between the deformed moving image and the static image by the mutual-transformed intensity difference and searches for the transformation mapping maximizing the mutual information with the least conformality distortion.

The QC model has the merit to control the conformality distortion of the transformation, which in turns controls the smoothness of the mapping without losing bijectivity and diffeomorphicity. Figure 5 demonstrates two examples of the image registration and fusion by the QC model.

In (2018), Zhang and Chen proposed another quasi-conformal image registration model. They introduced a novel, unbiased, and robust regularizer which is reformulated from Beltrami coefficient framework to ensure a diffeomorphic transformation. With a suitable approximation of the exact Hessian matrix which is necessary to derive a convergent iterative method, their model not only get a diffeomorphic registration even when the deformation is large but also possess high accuracy as compared with other existing models.

Other Imaging Applications

In (2008), Saucan et al. presented a method and algorithm of flattening folded surfaces, for two-dimensional representation and analysis of medical images. The method is based on an application to triangular meshes of classical results of Gehring and Vaisala regarding the existence of quasi-conformal and quasi-isometric

mappings. They demonstrated their algorithm to be robust and effective. Further applications of the algorithm, for image processing in general, are also considered.

In (2013), Jones et al. presented a method for the generation of a smooth morphometric mapping between two planar domains which matches a number of homologous points to characterize the diversity of planar shapes. Specifically, they focused on aspects of shape as characterized by local rotation and shear, quantified using quasi-conformal maps that are defined precisely in terms of these fields. They implemented the algorithm using a variational principle that optimizes the coefficients of the quasi-conformal map between the two regions. If applied to the medical industry, it is believed to promote advantages over existing methods.

In (2001), Heisterkamp et al. presented a novel approach to ranking relevant images for retrieval. Distance in the feature space associated with a kernel is used to rank relevant images. An adaptive quasi-conformal mapping based on relevance feedback is used to generate successive new kernels. The proposed model created by the quasi-conformal kernel is used to measure the distance between the query and the images in the database. This model can be advantageous in medical imaging applications.

In (2017), Dong et al. proposed and realized a two-dimensional flattened Luneburg lens using quasi-conformal mapping in the acoustic regime, allowing geometries with curved shapes to be converted into flat systems while the broadband and low-loss properties are preserved. Their results may give rise to various applications, including medical treatments and medical imaging systems.

Surface Analysis for Medical Applications

Surface processing and analysis tools are crucial bridges between medical data and other applications. For example, given a segmentation of anatomical structures, the corresponding surface can be simulated and meshed for further clinical analysis. The simulated surfaces may be used in, for example, database generation, disease diagnosis and case study, etc. In this section, we will introduce how the quasi-conformal geometry may participate in this field.

3D Surface Registration

Surface registration plays an important role in defining a meaningful correspondence between surfaces. Here, the quasi-conformal geometry finds its impact in greatly improving the efficiency of the registration process.

In particular, in (2015), Choi et al. proposed an algorithm, called the FLASH, for cortical surface registration with landmark matching. The FLASH algorithm computes the optimized spherical harmonic parametrization with consistent landmark alignment. It achieves fast computation since the quasi-conformal theory allows linearizing the whole implementation model.

Given two cortical surfaces to be registered, the FLASH algorithm reformulates the computation of the spherical conformal parametrization for the surfaces as finding an optimized harmonic mapping matching the given landmarks. The idea is to obtain a conformal parametrization of each surface by solving the sparse linear system:

$$\begin{cases} \sum_{[u,v] \in K} k_{uv}(\phi(u) - \phi(v)) = 0 \text{ if } u \neq v_{j_1}, v_{j_2}, v_{j_3}; \\ \phi(v_{j_t}) = b_t \text{ if } t = 1, 2, 3. \end{cases}, \quad (23)$$

then compute the spherical mesh by inverse stereographic projection. The landmark aligned spherical map is therefore obtained by an extra stereographic projection and solving a Laplace equation and compositing with a quasi-conformal mapping within the process. Figure 6 demonstrates an example of the registration result between two

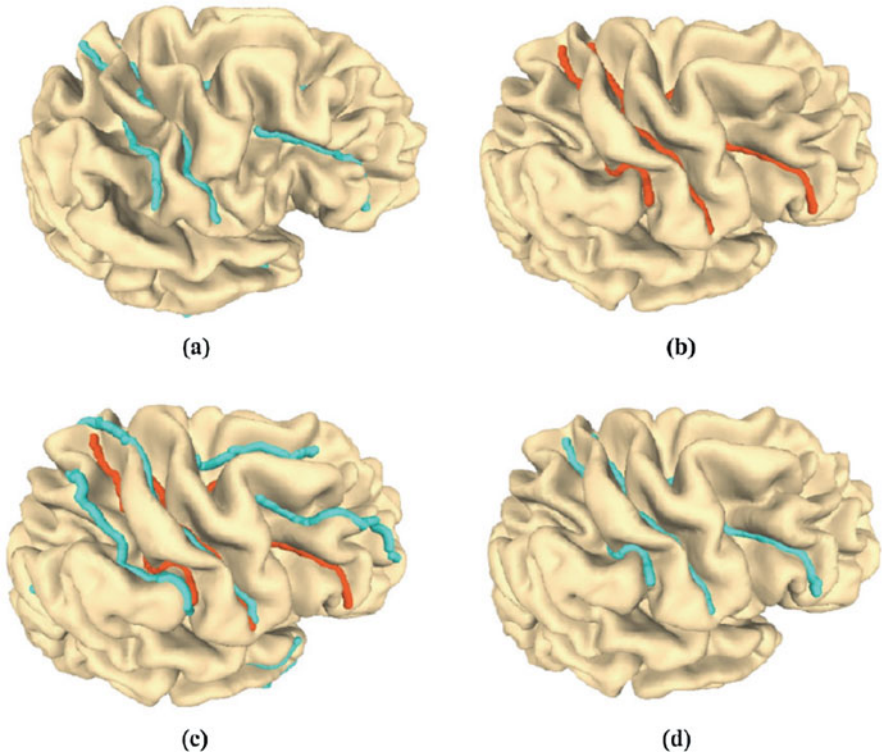


Fig. 6 An example demonstrating the registration result of the FLASH algorithm. The sulcal landmarks are highlighted. (a) and (b) show the source cortical surface and the target cortical surface, respectively. (c) shows the conformal registration without any landmark constraints. One can observe that the landmark curves are not matched. (d) shows the registration obtained using FLASH:

cortical surfaces using the FLASH algorithm. The FLASH algorithm usually takes several seconds to finish the whole registration process which is hundred times faster than other currently used models.

It should be emphasized that the quasi-conformal geometry plays a crucial role in the FLASH algorithm. After computing the landmark-aligned mapping ϕ in (23), the Beltrami differential μ_ϕ of ϕ is computed. And μ_ϕ is smoothed to be a Beltrami coefficient μ by the variational model:

$$\mu_{\text{smooth}} = \arg \min_{\mu} \int (|\nabla \mu|^2 + |\mu - \mu_\phi|^2 + A(T)|\mu|^2), \quad (24)$$

where $A(T)$ is the area of the triangular face T on the plane. This step ensures the mapping f corresponding to μ is smooth, in which ϕ does not necessarily possess this property. The variational model above is solved by the linear Beltrami solver (LBS).

The framework, in particular, can be applied to hippocampal surface registration. According to medical research, the hippocampus would undergo abnormal deformation in the prodromal stage of the Alzheimer's disease. However, the hippocampus shows no obvious landmark on the surface. It is a challenging task to correspond the hippocampal surfaces and analyze the deformation.

Motivated by the situation, in (2020), Chan et al. propose a registration model, ACC-REG, for hippocampal surfaces. Given two hippocampal surfaces, ACC-REG automatically generates two landmark curves using the eigen-graph on the surfaces. A histogram matching mapping is applied onto the two eigen-graphs to calibrate the propagation of the landmark curves along the surfaces. Afterwards, ACC-REG employs the FLASH algorithm to register the two hippocampal surfaces.

Figure 7 demonstrates an example of the calibrated eigen-graph on a hippocampal surface. Figure 8 shows two experiments of the ACC-REG model. The results have demonstrated the effectiveness of the ACC-REG model to obtain an accurate registration between hippocampal surfaces.

In (2011), Zeng and Gu proposed a quasi-conformal model for surface registration by solving Beltrami equations using curvature flow. The proposed model can attain at global minimum which is unique up to a three-dimensional transformation group.

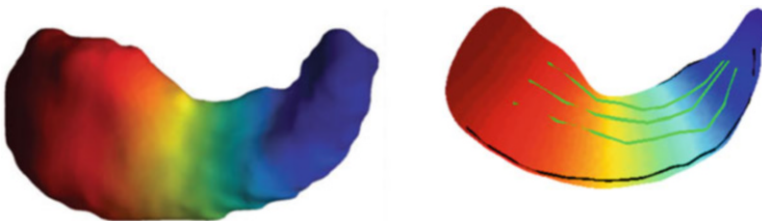


Fig. 7 An example of the eigen-graph and landmark curves on a hippocampal surface extracted from Chan et al. (2020). (Left) Eigen-graph with function values goes from blue (0) to red (1); (right) landmark curves (green and black) on the surface

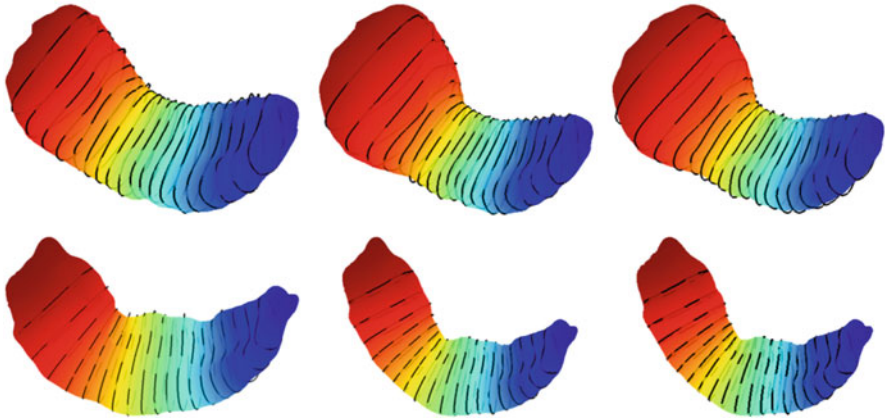


Fig. 8 Two examples of the surface registration of ACC-REG. (Left) Original surfaces; (middle) artificially deformed surfaces; (right) registered surfaces by the ACC-REG model

In (2019), Ma et al. applied the optimal mass transport mapping (OMT-Map) and Teichmüller mapping (T-Map) to solve for a unique bijective surface registration with landmark constraints in case of large deformations. Their model is advantageous in enforcing the robustness by avoiding large area distortion and producing diffeomorphisms with all landmarks matched consistently. Medical applications of the model is believed to be promising.

In Gu et al. (2004) and Wang et al. (2007), analyzed a family of quasi-conformal maps including harmonic maps, conformal maps, and least-squares conformal maps with regard to 3D shape matching and hence proposed a novel and computationally efficient shape matching framework by using least-squares conformal maps. Their model achieves high accuracy and efficiency in 3D shape matching. Their model, if applied to the medical industry, is believed to be one another powerful tool to use.

High-Dimensional Shape Deformation

In some situation, 3D surfaces may not be a good candidate to represent a complex anatomical structure. 4D surface (in other words, 3D volumetric data) may be employed for medical analysis. To deal with the registration of 3D volumetric data, the traditional quasi-conformal models can be generalized to general n -dimensional spaces, in particular, the 3D space.

In (2016), Lee et al. proposed to generalize the notion of quasi-conformality distortion by extending the concept that quasi-conformal maps deform infinitesimal disk to infinitesimal ellipse. Given a mapping f of 3D volumetric data, Lee et al. defined the 3D conformality distortion by

$$Kf(x) := \begin{cases} \frac{\|Df(x)\|_F^2}{\det(Df(x))^{2/n}}, & \text{if } \det Df(x) > 0, \\ +\infty, & \text{otherwise} \end{cases} \quad (25)$$

Conceptually, $Kf(x)$ determines the local distortion of an infinitesimal ball to an infinitesimal ellipsoid under the mapping f . A registration model is formulated by minimizing the 3D conformality distortion together with a smoothness regularization:

$$\inf_{f \in F} \|Kf(x)\|_1 + \frac{\sigma}{2} \|\delta f(x)\|_2^2 dx \quad (26)$$

subject to the landmark matching constraint

$$f(p_i) = q_i, \quad i = 1, 2, \dots, m. \quad (27)$$

The model helps to register between volumetric data. In particular, Fig. 9 demonstrates two examples of 3D lung data registration using the high-dimensional QC model.

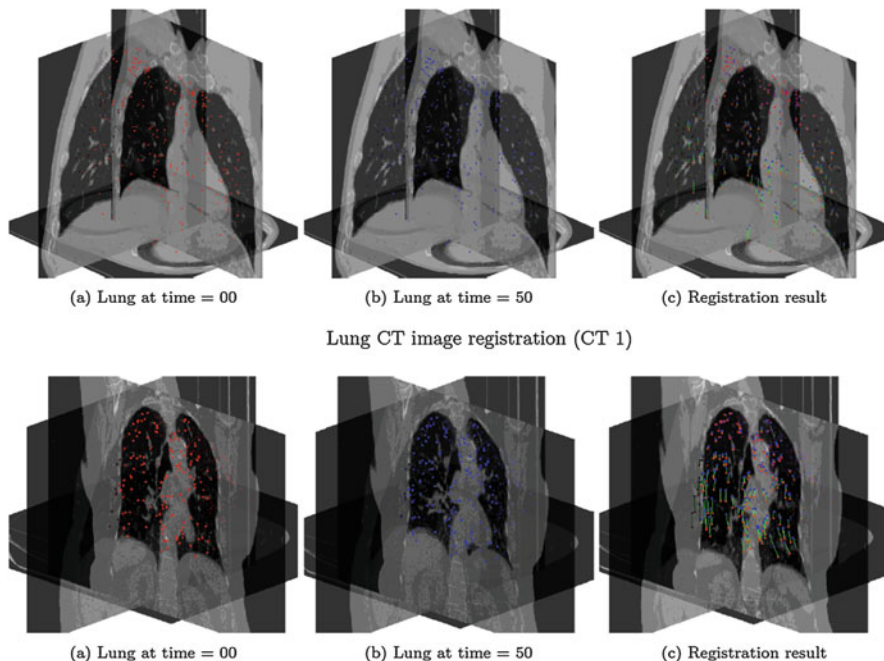


Fig. 9 Two examples of the 3D lung data registration extracted from Lee et al. (2016). The red and blue dots on the left and middle images indicate the landmark points. The vectors at each vertex on the right image indicates the alignment of the landmark points by the QC model

The extension of quasi-conformal geometry does not allow registration between higher-dimensional data. In (2014), Chan et al. proposed a n -dimensional shape deformity quantifier called the anisotropic indicator. The indicator measures the extended conformality distortion of a high-dimensional mapping which reads

$$Aid_f(x) = \frac{L_f(x) - 1}{L_f(x) + 1}$$

where $L_f : M \rightarrow N$ is defined by

$$L_f(x) := \lim_{r \rightarrow 0} \sup_{\substack{u, v \in S_x^M(r) \\ u \neq v}} \frac{|f(u) - f(x)|}{|f(v) - f(x)|}.$$

The anisotropic indicator is a local geometric measurement for a n -dimensional shape deformation. Inspired by the infinitesimal behavior of a quasi-conformal mapping to map an infinitesimal disk to an infinitesimal ellipse, the anisotropic indicator determines the local property of a deformation by analyzing its behavior on an infinitesimal n -d ball.

Given a n -d deformation mapping between two volumetric data, the anisotropic indicator reports a number varies from 0 to 1 at each vertex and can be visualized on the data by coloring the transparent plot of the 3D volumetric data. Experiments on the brain data and the lung data are demonstrated in Chan et al. (2014). A selection of those experiments are shown in Fig. 10.

In (2015), Naitat et al. introduced methods for assessing the extent of the local and global volumetric deformation by means of the amount of conformal distortion produced. They first illustrated basic three-dimensional quasi-conformal

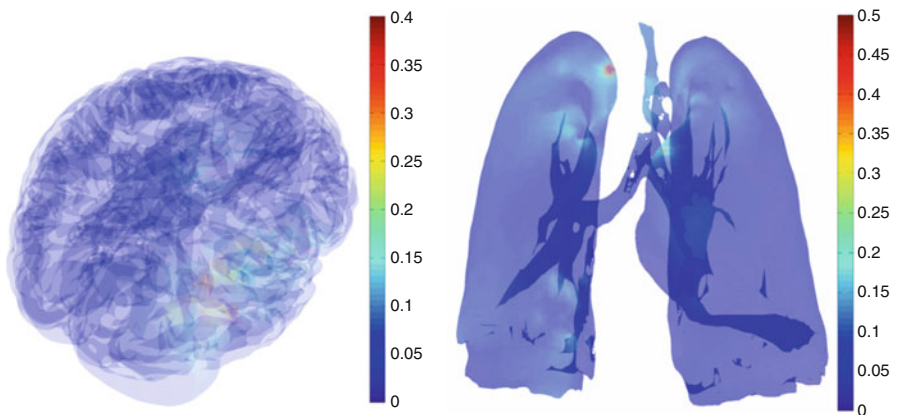


Fig. 10 Two examples of the color plot of the anisotropic indicator on 3D volumetric data extracted from Lee et al. (2016). (Left) Brain data; (right) lung data

deformations that are produced by parameterization techniques and highlighted theoretical issues associated with spatial quasi-conformal mappings, and the relation that exists between the geometry of the domain and conformal distortion. Their study may be applied to study volumetric deformation of medical tissues.

Disease Diagnosis and Classification by Quasi-conformal Geometry

Machine learning models have been extensively applied in disease diagnosis. Despite their usefulness, determining features with high discriminating power on the examining subjects is definitely necessary to boost up their practicability in real-world applications. Conformal/quasi-conformal mappings find a great use in this aspect. They can be applied to reveal geometric differences of anatomical structures between different class of subjects.

Classification of the Alzheimer's Disease

The Alzheimer's disease is a no-cure disease. One of the crucial tasks in dealing with this disease is to detect it in the early stage. It is evident that the hippocampus would show abnormal deformation in the early stage of the disease. In (2016), Chan et al. proposed an Alzheimer's disease (AD) classification model which analyzes the hippocampal surfaces by considering their local geometric distortions. The key is to combine local shape deformities including the conformality distortion, the Gaussian curvature distortion, and the mean curvature distortion of the deformation of a subject's hippocampal surface along the longitudinal direction (i.e., different time frame). More specifically, Chan et al. proposed a shape index:

$$E_{\text{shape}}^i(\mathbf{v}_i^j) = \gamma|\mu(f_i)(\mathbf{v}_i^j)| + \alpha|H_0(\mathbf{v}_i^j) - H_1(f_i(\mathbf{v}_i^j))| + \beta|K_0(\mathbf{v}_i^j) - K_1(f_i(\mathbf{v}_i^j))| \quad (28)$$

The shape index is a complete descriptor of the local deformation of the hippocampal surface mesh and is taken to be the vertex-wise feature to classify the disease. In Chan et al. (2016), the authors are given a database consisting of 99 normal control subjects and 41 AD subjects. After registering each pair of the surfaces, the shape index is computed for each surface. All the shape indexes are stacked to form a feature matrix, and a modified t-test is applied to extract features with high discriminating power. The trimmed feature matrix is then used to build a L^2 -norm-based binary classification model. The model is found to be effective in classifying the Alzheimer's disease and results in a 87.9% accuracy in a leave-one-out validation test on the given database. Here it is emphasized that the conformality distortion term $|\mu(f_i)(\mathbf{v}_i^j)|$ plays a crucial role in analyzing the infinitesimal distortion of the mapping. Without this term, the classification rate drops significantly from 87.9% to 77.1%.

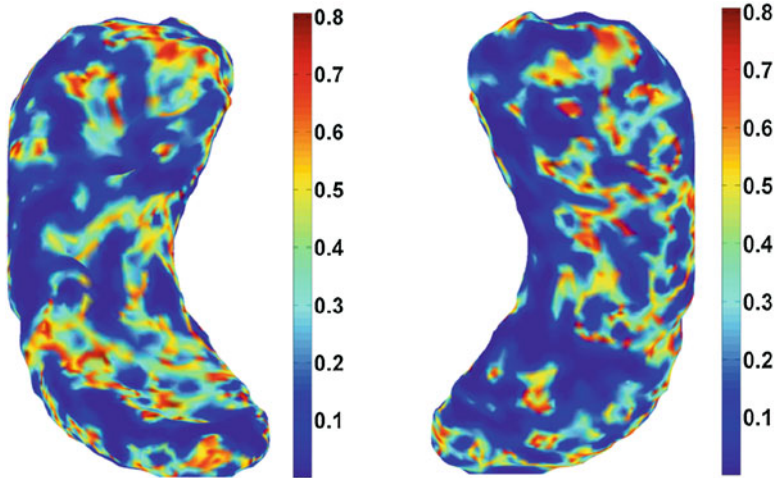


Fig. 11 An example of the color plot of shape index on a hippocampal surface extracted from Chan et al. (2020). (Left) Front view; (right) back view

The QC model does not only predict the disease status of the given subject. Since the shape index is a local indicator, one can visualize the location of the abnormal deformation by a color plot of the p-value of the shape index. Figure 11 demonstrates an example of the color plot. The QC model helped medical doctors to easily locate the regions of abnormalities which is contributing to further medical analysis.

Later in (2020), Chan et al. further proposed an AD diagnosis model by joint-forcing the quasi-conformal geometry and the spherical harmonics (SPHARM) theory. By applying the SHREC algorithm derived from the SPHARM theory, a template mean surface can be simulated by investigating the normal control subjects. Deformation of the hippocampus can therefore be regarded as that from the template surface to the subject surface. This releases the necessity for longitudinal data as in the previous model and allows instant diagnosis of the disease. The SPHARM registration also provides a set of global features, the SPHARM coefficients, on the hippocampal surface. They are combined with the quasi-conformal-based shape index, and the volume distortion from the template surface to the subject surface, to formulate a geometric feature vector for each surface:

$$\mathbf{c}_i = (e_{i,1}, e_{i,2}, \dots, e_{i,N} | r_{i,1}, r_{i,2}, \dots, r_{i,K} | v_i), \quad (29)$$

where $e_{i,j} = E_{\text{shape}}^i(\mathbf{v}^j)$ is the shape index at each vertex, $r_{i,k} = r_{ik}$ is the collection of all SPHARM coefficients up to certain degree on the surface, and v_i is the global volume distortion. The feature vector combines both local and global geometric distortion measurements and is highly discriminative in classifying the disease. The support vector machine (SVM) is used to build the classification machine.

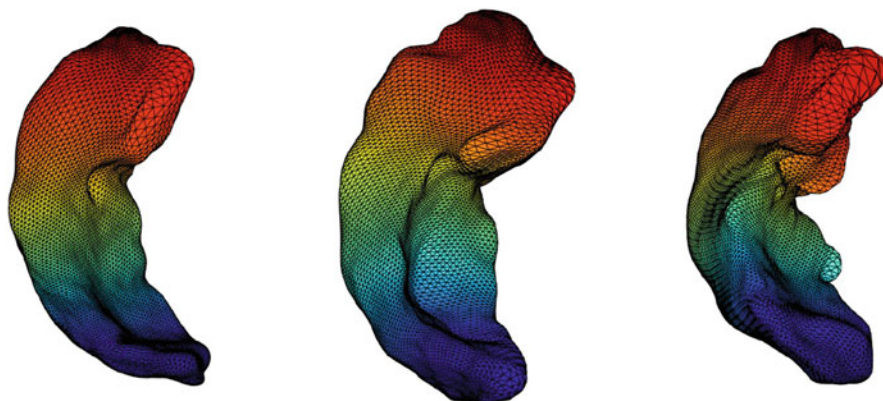


Fig. 12 Three examples of the hippocampal surfaces extracted from Chan et al. (2020). (Left and middle): normal control subjects; (right): AD subjects

In Chan et al. (2020), authors are given two sets of data. The first set of data consists of 110 normal control subjects and 110 AD subjects. In the experiment part, 140 training data are randomly chosen to build the classification model using a 10-fold cross-validation scheme. The remaining 80 data is taken as the testing data. The process is repeated for 1,000 times and our model records over 85% accuracy on average.

According to medical research, in the prodromal stage of AD, there is a medical situation called the amnesic mild cognitive impairment (aMCI). While some aMCI patients may stay stable in the current state, some patients may further progress into AD. It is a challenging task to classify the two groups of patients. In Chan et al. (2020), the authors are also concerned with the prediction of the disease progression by the QC-SPHARM model. A database consisting of 40 aMCI patients is given, in which 20 of them remain stable in aMCI for dozens of years after scanning for the hippocampal surface and the remaining 20 of them progressed into AD soon after the scan. The authors run an experiment to randomly picked 30 data to build the classification model and used the remaining 10 data to test the accuracy of the classifier. The process is repeated for 1,000 times, and the result showed that the QC-SPHARM model achieves over 81% accuracy in predicting the further development of the disease status. Figure 12 demonstrates three examples of the registered hippocampal surfaces in the database for reference.

Other Classification Model

The concept of those quasi-conformal-based disease diagnosis model also finds applications outside the medical industry. In particular, in (2020), Choi et al. proposed a surface analysis framework using the quasi-conformal Teichmüller theory (Lui et al. 2014; Meng et al. 2016) for skull dating, which is an important

task in the bioarchaeology industry. According to bioarchaeologists, it is suggested that the human tooth would show different geometry across genders and ancestries. In the paper, authors proposed to date a body by analyzing the deformation of its tooth surface from a template tooth surface. The deformation is described by the shape index:

$$E_{\text{shape}}(f_i)(v^k) = \alpha |H_i(v^k) - H(f_i(v^k))| + \beta |K_i(v^k) - K(f_i(v^k))| + \gamma d_i, \quad (30)$$

which involves the Gaussian curvature distortion term, the mean curvature distortion term, and the Teichmüller distance term. A t-test-based scheme is applied followed by the SVM to build the classification machine.

It is noteworthy that in Choi et al.'s work, they also proposed the spherical marching scheme (SMS) to optimize the parameters α , β , and γ in terms of higher classification accuracy. The spherical marching scheme makes use of the fact that the norm of the shape index has no contribution to the classification process. Therefore, the space of the parameters (α, β, γ) can be restricted on the unit sphere. And the optimized parameters can be exhaustively searched by regular gridding on the domain of the unit sphere in spherical coordinates. That is

$$(\alpha, \beta, \gamma)_{n,m} = (\sin(n\rho)\cos(m\rho), \sin(n\rho)\sin(m\rho), \cos(n\rho)), \quad (31)$$

with a density parameter ρ .

The model is tested with a database involving 70 subjects of different genders and ancestries. The results showed that it has over 97% accuracy in dating the subjects across both genders and ancestries. Figure 13 illustrates the whole pipeline of the proposed framework for reference.

As for deformation analysis, in Taimouri and Hua (2014), Taimouri et al. proposed a novel quasi-conformal metric to classify the deformations in shape space. Using the concept that shapes with similar deformation patterns follow a similar deformation curve in shape space, a geodesic curve connecting the two shapes is computed on the shape space manifold. The geodesic distance illustrates the similarity between two shapes, which is used to compute the similarity between the deformations. They applied their model on left ventricle deformations of myopathic and control subjects, achieving a sensitivity of 88.8% and a specificity of 85.7%.

Conclusion

Quasi-conformal Teichmüller theory is playing an important role in many aspects in the medical industry. It can be applied to medical imaging for image segmentation, registration and fusion, etc. The QC-based models provides high flexibility to incorporate medical knowledge about the desired results. In surface analysis, the QC theory finds its contribution in multiple tasks. For instance, it can be used to boost the efficiency of computing the registration of 3D surfaces and volumetric

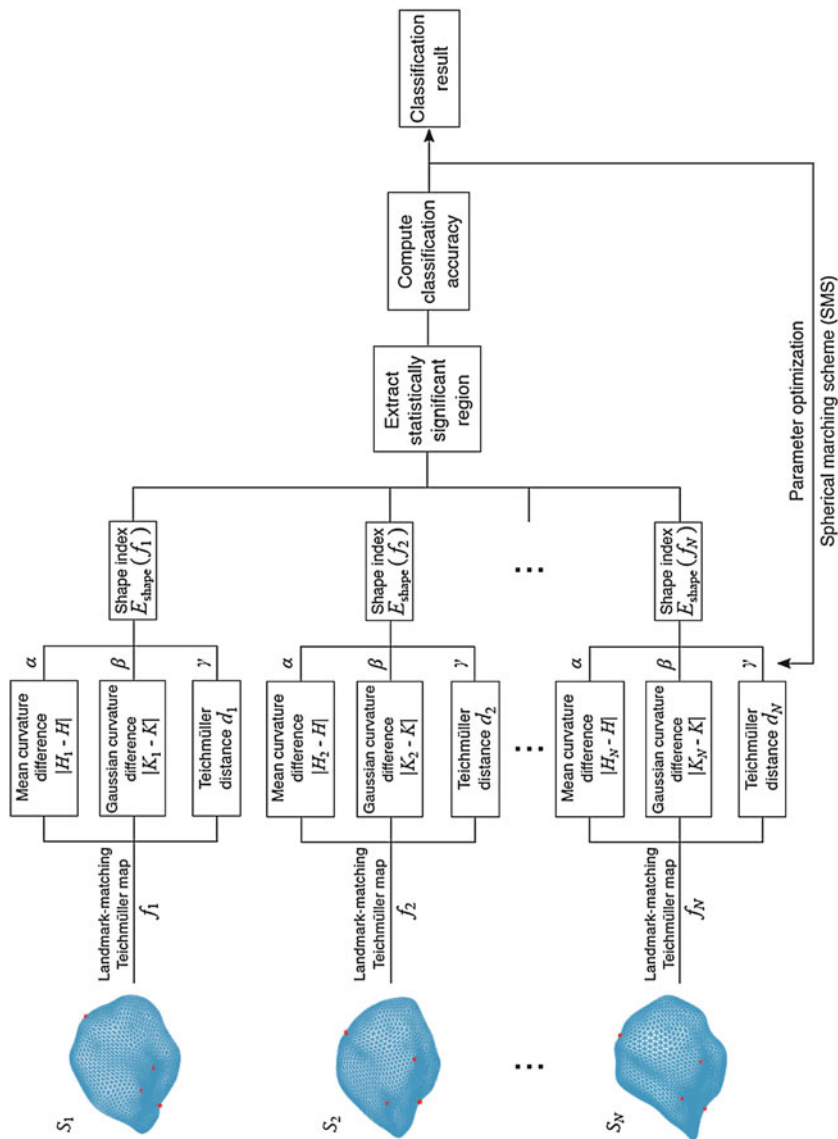


Fig. 13 The pipeline of the quasi-conformal based skull dating model extracted from Choi et al. (2020)

data. The conformality distortion can also be used to measure the abnormality of the deformation mapping. The QC theory is also helpful in disease classification. By incorporating the conformality distortion and the Teichmüller distance with other common measurements, a trustworthy disease classification model can be built with high accuracy and stability.

References

- Chan, H.-L., Lui, L.-M.: Detection of n -dimensional shape deformities using n -dimensional quasi-conformal maps. *Geometry Imaging Comput.* **1**(4), 395–415 (2014)
- Chan, H.-L., Li, H., Lui, L.-M.: Quasi-conformal statistical shape analysis of hippocampal surfaces for Alzheimer's disease analysis. *Neurocomputing* **175**, 177–187 (2016)
- Chan, H.-L., Yan, S., Lui, L.-M., Tai, X.C.: Topology-preserving image segmentation by Beltrami representation of shapes. *J. Math. Imaging Vision* **60**(3), 401–421 (2018)
- Chan, H.-L., Yam, T.-C., Lui, L.-M.: Automatic characteristic-calibrated registration (ACC-REG): Hippocampal surface registration using Eigen-graphs. *Pattern Recogn.* **103**, 107142 (2020)
- Chan, H.-L., Luo, Y., Shi, L., Lui, L.-M.: QC-SPHRAM: Quasi-conformal Spherical Harmonics Based Geometric Distortions on Hippocampal Surfaces for Early Detection of the Alzheimer's Disease. *Computerized Medical Imaging and Graphics* (Submitted 2020)
- Choi, P.-T., Lam, K.-C., Lui, L.-M.: FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM J. Imag. Sci.* **8**(1), 67–94 (2015)
- Choi, G.P.T., Chan, H.-L., Yong, R., Ranjitkar, S., Brook, A., Townsend, G., Chen, K., Lui, L.-M.: Tooth morphometry using quasi-conformal theory. *Pattern Recogn.* **99**, 107064 (2020)
- Dong, H.Y., Cheng, Q., Song, G.Y., Tang, W.X., Wang, J., Cui, T.J.: Realization of broadband acoustic metamaterial lens with quasi-conformal mapping. *Appl. Phys. Express* **10**(8), 087202 (2017)
- Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* **23**(8), 949–958 (2004)
- Heisterkamp, D.R., Peng, J., Dai, H.K.: Adaptive quasiconformal kernel metric for image retrieval. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 2. IEEE* (2001)
- Jones, G.W., Mahadevan, L.: Planar morphometry, shear and optimal quasi-conformal mappings. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **469**(2153), 20120653 (2013)
- Lam, K.-C., Lui, L.-M.: Quasi-conformal hybrid multi-modality image registration and its application to medical image fusion. In: *International Symposium on Visual Computing. Springer, Cham* (2015)
- Lam, K.-C., Lui, L.-M.: Landmark and intensity based registration with large deformations via quasi-conformal maps. *SIAM J. Imag. Sci.* **7**(4), 2364–2392 (2014)
- Lee, Y.-T., Lam, K.-C., Lui, L.-M.: Landmark-matching transformation with large deformation via n -dimensional quasi-conformal maps. *J. Sci. Comput.* **67**(3), 926–954 (2016)
- Lui, L.M., Wong, T.W., Zeng, W., Gu, X., Thompson, P.M., Chan, T.F., Yau, S.T.: Optimization of surface registrations using Beltrami holomorphic flow. *J. Sci. Comput.* **50**(3), 557–585 (2012)
- Lui, L.M., Lam, K.C., Yau, S.T., Gu, X.F.: Teichmüller mapping (T-Map) and its applications to landmark matching registrations. *SIAM J. Imag. Sci.* **7**(1), 391–426 (2014)
- Ma, M., Yu, X., Lei, N., Si, H., Gu, X.: Optimal mass transport based brain morphometry for patients with congenital hand deformities. *Vis. Comput.* **35**(9), 1311–1325 (2019)
- Meng, T.W., Choi, P.T., Lui, L.M.: TEMPO: Feature-Endowed Teichmüller extremal mappings of point clouds. *SIAM J. Imag. Sci.* **9**(4), 1922–1962 (2016)
- Naitsat, A., Saucan, E., Zeevi, Y.Y.: Volumetric quasi-conformal mappings. In: *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications* (2015)
- Saucan, E., Appleboim, E., Barak-Shimron, E., Lev, R., Zeevi, Y.Y.: Local versus global in quasi-conformal mapping for medical imaging. *J. Math. Imaging Vision* **32**(3), 293–311 (2008)

- Siu, C.-Y., Chan, H.-L., Lui, L.-M.: Image segmentation with partial convexity prior using discrete conformality structure. *SIAM J. Imag. Sci.* (Accepted 2020)
- Taimouri, V., Hua, J.: Deformation similarity measurement in quasi-conformal shape space. *Graph. Model.* **76**(2), 57–69 (2014)
- Wang, Y., Lui, L.M., Gu, X., Hayashi, K.M., Chan, T.F., Thompson, P.M., Yau, S.-T.: Brain surface conformal parameterization using Riemann surface structure. *IEEE Trans. Med. Imaging* **26**(6), 853–865 (2007)
- Zeng, W., Gu, X.D.: Registration for 3D surfaces with large deformations using quasi-conformal curvature flow. In: *CVPR 2011*. IEEE (2011)
- Zhang, D., Chen, K.: A novel diffeomorphic model for image registration and its algorithm. *J. Math. Imaging Vision* **60**(8), 1261–1283 (2018)
- Zhang, M., An, D., Young, G.S., Gu, X., Xu, X.: A quasi-conformal mapping-based data augmentation technique for brain tumor segmentation. In: *Medical Imaging 2020: Image Processing*, vol. 11313, p. 113132P. International Society for Optics and Photonics (2020)



A Survey of Topology and Geometry-Constrained Segmentation Methods in Weakly Supervised Settings

42

Ke Chen, Noémie Debroux, and Carole Le Guyader

Contents

Introduction	1438
Geometrical Constraints	1441
Characterization of Geometrical Constraints	1442
Model 1: A Simple Variational Model	1443
Model 2: A Moving Band Model	1443
Model 3: A Dual Level Model	1444
Model 4: The Use of Moment Constraint for Segmentation	1445
Model 5: Convex Segmentation Models	1446
Model 6: Convex Models Based on Geodesic Distances	1446
Other Possible Models	1447
Topological Prior Knowledge	1449
Topology Prescription	1451
Regularization Enforcement on the Evolving Front	1459
Joint Segmentation and Registration Models	1463
Motivations	1463
Overview of Existing Methods	1466
A Mixed Segmentation/Registration Model Based on a Nonlocal Characterization of Weighted Total Variation	1468
Other Related Models	1473
Optimal Flow Frameworks	1473

K. Chen (✉)

Department of Mathematical Sciences, Centre for Mathematical Imaging Techniques, University of Liverpool, Liverpool, UK
e-mail: K.Chen@liverpool.ac.uk

N. Debroux

Pascal Institute, University of Clermont Auvergne, Clermont-Ferrand, France
e-mail: noemie.debroux@uca.fr

C. Le Guyader

INSA Rouen Normandie, Laboratory of Mathematics, Normandie University, Rouen, France
e-mail: carole.le-guyader@insa-rouen.fr

Shape Priors	1474
Deep Learning Models	1474
Multi-modal Problems	1474
Conclusion	1475
References	1475

Abstract

Incorporating prior knowledge into a segmentation process— whether it be geometrical constraints such as landmarks to overcome the issue of weak boundary definition, shape prior knowledge or volume/area penalization, or topological prescriptions in order for the segmented shape to be homeomorphic to the initial one or to preserve the contextual relations between objects— proves to achieve more accurate results, while limiting human intervention. In this contribution, we intend to give an exhaustive overview of these so-called weakly/semi-supervised segmentation methods, following three main angles of inquiry: inclusion of geometrical constraints (landmarks, shape prior knowledge, volume/area penalization, etc.), incorporation of topological constraints (topology preservation enforcement, prescription of the number of connected components/holes, regularity enforcement on the evolving front, etc.), and, lastly, joint treatment of segmentation and registration that can be viewed as a special case of cosegmentation.

Keywords

Weakly supervised segmentation · Geometrical and topological priors · Selective segmentation · Digital topology · Level set-based variational models · Quasiconformal mappings · Higher-order schemes · Joint segmentation and registration · Nonlocal models

Introduction

Image segmentation is an essential step in image processing on the way to make image analysis automatic, aiming to reproduce the ability of human beings to track down significant patterns and automatically gather them into relevant and identified structures with respect to features such as color, shape, or orientation (Zhu et al. 2016). More specifically, image segmentation consists in identifying meaningful constituents of a given image (e.g., homogeneous regions, shapes, edges, textures, etc.) for quantitative analysis or visualization purposes. Due to its countless applications, among which object detection, complexity reduction, scene parsing, image montage, colorization, organ reconstruction, tumor detection, computer-aided diagnosis, and therapy planning, to name a few (see Zhu et al. 2016 for an exhaustive overview), a lot of research has been carried out during the last three decades.

Not all image shapes and patterns extracted provide useful information. The usefulness is relative to applications. For instance, an automatic car mainly cares about stationary and moving objects on its path, not objects (such as buildings or trees) that are far away, while in medical imaging, a specialist on liver diseases is not primarily interested to see patterns in lungs and abdomen. Therefore in this practical sense, only models that have topology and geometry constraints built-in to extract patterns of interest are really valuable, while other generic segmentation models capable of identifying all objects are not helpful.

Although simple to state, this task is nevertheless challenging and ill-posed as emphasized by Zhu et al. in the comprehensive segmentation survey (Zhu et al. 2016):

- (i) First, owing to the polysemy of the word *object* and because interpretation is intrinsically subjective: different human beings may have different views of what an object is. The definition of an *object* encompasses several acceptations according to human perception: it can be something material (a thing), a periodic pattern, an overall structure (e.g., a forest, the sea), or even a sub-part of a given object (e.g., a tumor in a brain MRI image).
- (ii) Second, due to the difficulty in computerizing/reproducing the human vision system, capable of synthesizing (interpolating) the observed data into a continuous whole, human tends to merge elements taking on shared similarities, to complete missing data, to favor continuous contours, etc., whereas most images in computers are represented by low-level characteristics reflecting mainly local properties and failing thus to capture the global (continuous) nature of the observed object.

These two elements together make the evaluation of segmentation techniques still an open question.

An exhaustive classification of segmentation methods into three main categories is provided in Zhu et al. (2016) and ordered according to the level of supervision or user involvement, combined with a description/analysis of each methodology as follows:

- (i) Global models: unsupervised methods which consist in partitioning a given image into meaningful constituents based only on low-level features (e.g., intensity levels, curvature, etc.) with no human intervention and without any training data and a priori knowledge of the object model. These unsupervised methods are themselves subdivided into two groups: discrete methods, setting in which the image is considered as a fixed discrete grid, including clustering-based approaches (Chuang et al. 2006; Comaniciu and Meer 2002; Ohlander et al. 1978; Rao et al. 2010) and graph-based methods (Felzenszwalb et al. 2004; Shi and Malik 2000), and continuous methods (Blake and Zisserman 1989; Caselles et al. 1997; Chambolle et al. 2012; Chan and Vese 2001; Kass et al. 1988; Li et al. 2007; Mory and Ardon 2007; Mumford and Shah 1989; Osher and Sethian 1988; Storath and Weinmann 2014; Aubert and Kornprobst

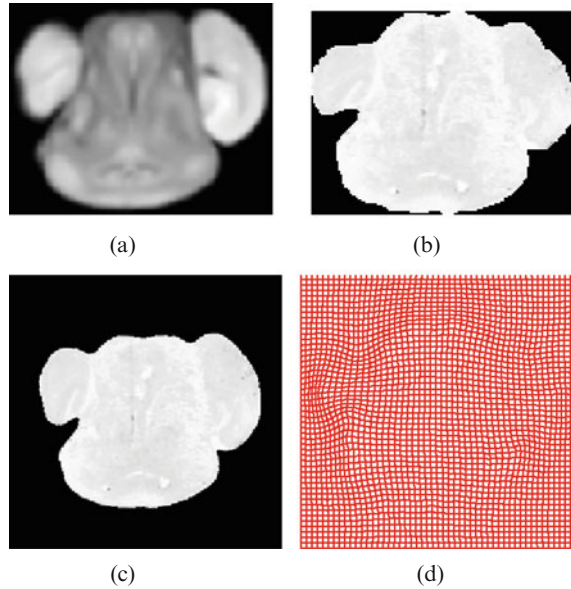
2001; Vese and Le Guyader 2015; Vese and Chan 2002; Wang et al. 2009)—framework in which the image is seen as a continuous surface, avoiding thus the grid bias artifacts inherent to discrete methods and producing visually more pleasing results—methods involving edge-based models and region-based ones.

- (ii) Local models: semi/weakly supervised methods which incorporate a small amount of high-level information and as such, are usually interactive and require human expertise and intervention to better match human perception. This class of methods is partitioned into two subclasses: (a) interactive methods that rely on a small amount of prior information provided by the user (e.g., labels of a few pixels as initial constraints) and that encompass three groups of methodology, contour tracking approaches (Osher and Sethian 1988; Mortensen et al. 1992; Liu and Yu 2012; He et al. 2013; McGuinness et al. 2010; Werlberger et al. 2009; Le Guyader and Gout 2008; Le Guyader and Vese 2008), label propagation approaches (Boykov and Jolly 2001; Grady 2006; Price et al. 2010; Bai and Sapiro 2007), and local optimization approaches (Hosni et al. 2013; Criminisi et al. 2008), and (b) image cosegmentation (Rother et al. 2006), well-suited for large-scale image dataset and which consists in identifying common objects in a set of images.
- (iii) Learning models: fully supervised methods (refer to Zhu et al. 2016, Section 4 and Garcia-Garcia et al. 2018 for an overview): they consist in training a segmentation algorithm thanks to fully annotated data—all pixels are labeled as either boundary or nonboundary—and then segmenting an unknown image. They reach high performance but the labeling is very expensive. However, more and more datasets are now available (see Zhu et al. (2016) for a list of them) with the explosion of machine learning-based algorithms and increasing computer abilities in the past few years.

In line with this classification, this chapter aims to focus on the second class of weakly supervised methods and more specifically, on interactive approaches (although the joint segmentation/registration models depicted below might be viewed as special instances of cosegmentation). The study entails the following three focal areas that can be envisioned as three distinct types of a priori knowledge included in the segmentation process and that structure the rest of the paper:

- (i) Geometrical constraints to define local objects such as incorporation of landmarks to overcome the issue of weak boundary definition or inclusion of shape prior knowledge (section “[Geometrical Constraints](#)”).
- (ii) Prescription of topological constraints in order for the final shape to be homeomorphic to the initial one, to comply with a pre-defined topology, or in order for the evolving segmenting curve to exhibit fine regularity properties (section “[Topological Prior Knowledge](#)”).
- (iii) Combined segmentation/registration models. Registration is at the crux of a wide range of applications as stressed in Chen et al. (2019); Modersitzki (2004); Oliveira et al. (2014); Sotiras et al. (2013): shape tracking; fusion of anatomical images from computerized tomography (CT), or magnetic

Fig. 1 Mapping of a 2D slice of mouse brain gene expression data to its counterpart in an atlas. (a) Reference (b) Template (c) Deformed Template (d) Deformed grid



resonance imaging (MRI) images, with functional images from positron emission tomography (PET), single-photon emission computed tomography (SPECT) or functional magnetic resonance imaging (fMRI), also called multi-modality fusion to facilitate intervention and treatment planning; computer-aided diagnosis and disease follow-up; surgery simulation; atlas generation to integrate anatomic, genetic, and physiological observations from multiple patients into a common space and to conduct statistical analysis; radiation therapy; assisted/guided surgery; anatomy segmentation; computational model building; and image subtraction for contrast-enhanced images. Given two images called template and reference, registration consists in determining an optimal diffeomorphic deformation φ mapping the template into the reference. This task is depicted in Fig. 1 taken from Ozeré et al. (2015). As structure/salient component/shape/geometrical feature matching and intensity distribution comparison rule registration, it sounds relevant to intertwine the segmentation and registration tasks into a single framework. In this approach (section “[Joint Segmentation and Registration Models](#)”), we make full use of segmentation from one image and the ability of registration to correlate different images (even across modalities).

Geometrical Constraints

We now present the first class of methods building models based on a given set of geometrical constraints.

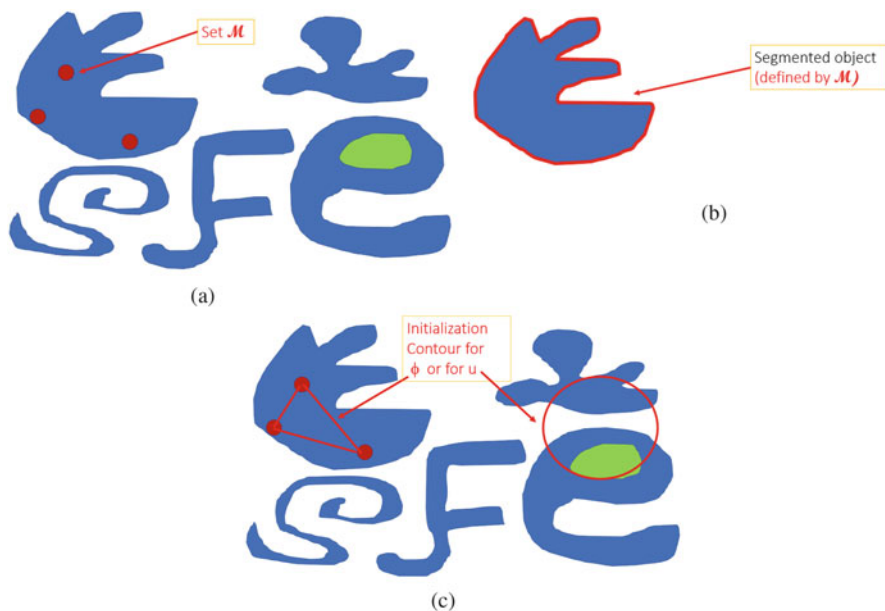


Fig. 2 Geometrical constraints and segmentation. Note the initial u is arbitrary. (a) Original image with \mathcal{M} (b) Segmented (local) object (c) Initialization: Left: ϕ (non-convex case) and Right: u (convex case)

The usefulness of identifying and extracting a single object of interest among others included in a given image has been recognized for many years. The pioneering work of Kass et al. (1988) discussed how to build spring-like forces, based on specifying the subsets (seed points) or subregions within object boundary, between control points of a snake in the energy functional to push out the snake out of a local minimum. A closely related idea of providing seed points is used in the live wire works such as Barrett and Mortensen (1997) and Chen et al. (2019) and in various references therein.

Below we focus more on variational frameworks. The first class of methods we present in this section aims to segment local image objects that are characterized by a set \mathcal{M} of landmarks or markers included in the image set Ω defined by $\mathcal{M} = \{x_i \in \Omega \mid i = 1, \dots, m\}$. Only the objects that are closest to \mathcal{M} will be segmented by those models, as illustrated in Fig. 2.

Characterization of Geometrical Constraints

The discrete set \mathcal{M} of markers needs to be converted into a representative function that can be included into a variational setting. This function, as used in Badshah and Chen (2010), Rada and Chen (2012), Zhang et al. (2014), Spencer and Chen (2015),

and Roberts et al. (2019), was defined as a Euclidean distance in Gout et al. (2005) and Le Guyader and Gout (2008)

$$d(x, y) = \prod_{i=1}^m \left(1 - \exp\left(-\frac{(x - x_i)^2}{2\sigma^2} - \frac{(y - y_i)^2}{2\sigma^2}\right) \right),$$

where σ is some scaling constant. Here $0 \leq d \leq 1$ satisfies $d \approx 0$ near \mathcal{M} and $d \approx 1$ away from it. An evolution equation of the form

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \nabla \cdot \left(d(x, y) g(|\nabla z(x, y)|) \frac{\nabla \phi(x, y, t)}{|\nabla \phi(x, y, t)|} \right)$$

is then derived (z denoting the considered image) to drive the initial level set curve to the desirable ϕ zero level line defining the intended object. Here g is an edge detector function that helps the contour evolve by ensuring that the front stops propagating when localized on meaningful contours. Extending this model to a variational framework, several models were then considered.

Initialization. In the models shown below, the level set ϕ is initialized automatically (i.e., no further user intervention is required) using the polygon formed by the marker points as displayed in Fig. 2c for 3 markers, depending on the convexity of the underlying model; we may place a small circle near the markers if the number of marker points is less than 3. However, when the underlying model is convex, initialization can be made with an arbitrary contour.

Model 1: A Simple Variational Model

The first and yet simple extension was done in Badshah and Chen (2010) by extending the Chan-Vese model (Chan and Vese 2001) to incorporate geometric constraints so that noisy images can be better segmented than in Gout et al. (2005) as follows

$$\min_{\phi, c_1, c_2} \int_{\Omega} dg(|\nabla z|) |\nabla H(\phi)| d\Omega + \int_{\Omega} \left(\lambda_1 H(\phi) |z - c_1|^2 + \lambda_2 (1 - H(\phi)) |z - c_2|^2 \right) d\Omega.$$

A weakness of Badshah and Chen (2010) lies in the fact that the obtained global solution often contains neighboring objects, which can be only avoided if one terminates iterations early. In some cases, the model reaches the same global segmentation result as the Chan-Vese solution.

Model 2: A Moving Band Model

One way to overcome the problem is to *know* when to stop the algorithm or when a model reaches the object boundary (which may not correspond to the local

minimizer of Badshah and Chen 2010 model). The idea in Zhang et al. (2014) is to evolve the initial curve from a polygon in a band fashion so that we would not step over the object boundary. The proposed model takes the form of

$$\inf_{\phi, c_1, c_2} \int_{\Omega} \left(\lambda_1 H(\phi) |z - c_1|^2 b(\phi, \gamma_{in}, \gamma_{out}) + \lambda_2 (1 - H(\phi)) |z - c_2|^2 b(\phi, \gamma_{in}, \gamma_{out}) \right) d\Omega + \int_{\Omega} dg(|\nabla z|) |\nabla H(\phi)| d\Omega,$$

where $b(\phi, \gamma_{in}, \gamma_{out}) = H(\phi - \gamma_{in}) H(\gamma_{out} - \phi)$ defines a narrow band of varying and adaptive widths $\gamma_{in}, \gamma_{out}$. The idea of “not stepping over” the desirable Γ when $\phi = 0$ is achieved by an adaptive searching algorithm based on checking local intensity variations (Zhang et al. 2014). Of course, the use of a varying band domain in the formulation leads to a highly non-convex model, making it hard to develop a theory.

Model 3: A Dual Level Model

To overcome the issue raised in Badshah and Chen (2010), alternatively, the idea of Rada-Chen (2012) in Rada and Chen (2012) is to compute the Chan-Vese solution by a level set function ϕ_G and also to compute the desirable solution (of an object only) via an embedded level set function ϕ_L , resulting in the dual level set formulation

$$\begin{aligned} \min_{\phi_G, \phi_L, c_1, c_2} & \left(\mu_L \int_{\Omega} dg(|\nabla z|) |\nabla H(\phi_L)| H(\phi_G + \gamma) d\Omega \right. \\ & + \mu_G \int_{\Omega} g(|\nabla z|) |\nabla H(\phi_G)| d\Omega \\ & + \int_{\Omega} \left(\lambda_{1G} H(\phi_G) |z - c_1|^2 + \lambda_{2G} (1 - H(\phi_G)) |z - c_2|^2 \right) d\Omega \\ & + \int_{\Omega} \left(\lambda_{1L} H(\phi_L) |z - c_1|^2 + \lambda_{2L} (1 - H(\phi_L)) H(\phi_G) |z - c_1|^2 \right) d\Omega \\ & \left. + \int_{\Omega} \lambda_{3L} |z - c_2|^2 (1 - H(\phi_L)) (1 - H(\phi_G)) d\Omega \right), \end{aligned}$$

where the first component balanced by μ_L essentially selects the desirable solution ϕ_L from the global solution ϕ_G , while the last term weighted by λ_{3L} helps separate the “true” background from the foreground intensity. The latter is because both objects included in ϕ_G and the desirable object included in ϕ_L often have the same intensity c_1 , i.e., objects included in ϕ_G but not desirable to our selection are quite different from the true background intensity c_2 . Parameter γ is an integer included to increase the search domain (by a small band) in the computation of the weighted

length of Γ_L . Its introduction increases the model robustness until final convergence. The above model was found extremely robust for selective segmentation, but its implementation doubles the amount of work one normally needs because of the use of two level set functions instead of the usual one.

Model 4: The Use of Moment Constraint for Segmentation

To stay with one level set function and yet to overcome the drawback of getting redundant objects beyond set \mathcal{M} , a useful idea is to impose the so-called moment constraints. The early work (Ayed et al. 2008) of 0th order moment (or area constraint) uses

$$\min_{\Gamma} \oint_{\Gamma} g(z) ds + \frac{\mu}{A_p^2} \left(\int_{R_{\Gamma}} d\Omega - A_p \right)^2 \int_{R_{\Gamma}^C} g(z) d\Omega$$

where A_p is an area prior (given), $g(z) = g(|\nabla z|)$ is an edge detector, R_{Γ} is the domain inside the closed curve Γ , while R_{Γ}^C is the outside domain. A level set reformulation is

$$\min_u \left(\int_{\Omega} g(z) |\nabla H(u)| d\Omega + \frac{\mu}{A_p^2} \left(\int_{\Omega} H(u) d\Omega - A_p \right)^2 \int_{\Omega} g(z) H(-u) d\Omega \right)$$

where $u > 0$ in domain R_{Γ} , $H(-u)$ indicates the outside domain (note the typo in the definition of level set function u in Ayed et al. (2008) which stated wrongly $u < 0$ in R_{Γ}). The above reformulation was surveyed in Nosrati and Hamarneh (2016) which unfortunately replaced the latter term $\int_{\Omega} g(z) H(-u) d\Omega$ by $\int_{\Omega_{in}} g(z) d\Omega = \int_{\Omega} g(z) H(u) d\Omega$ which is a major typo.

High-order moments were considered in Klodt and Cremers (2011). Denoting by u the indicator function, i.e., 1 inside object and 0 outside, the proposal in Klodt and Cremers (2011) for area constraint is

$$\min_u E(u) = \int_{\Omega} f u d\Omega + \int_{\Omega} g |Du| d\Omega + \lambda \left(\int_{\Omega} u d\Omega - A_p \right)^2,$$

where the last term to impose the area A_p of the targeted object is a simple version of a more general constraint

$$a_1 \leq \int_{\Omega} u d\Omega \leq a_2.$$

Here f is taken as the log likelihood ratio for observing $z(x, y)$ at a point (x, y) given that (x, y) is part of the background or the object. Higher-order moments refer to tensors of high order, e.g., centroid (first order) and covariance (second

order). Note that the set defined by the inequality is convex which could potentially be explored.

The model by Rada-Chen (2013) builds the area constraint into selection

$$\min_{\phi, c_1, c_2} \int_{\Omega} dg(|\nabla z|)|\nabla H(\phi)|d\Omega + \int_{\Omega} (\lambda_1 H(\phi)|z - c_1|^2 + \lambda_2 (1 - H(\phi))|z - c_2|^2) d\Omega + \nu \left(\int_{\Omega} H(\phi) d\Omega - A_1 \right)^2 + \nu \left(\int_{\Omega} (1 - H(\phi)) d\Omega - A_2 \right)^2,$$

where A_1 is the area defined by the polygon formed by markers in set \mathcal{M} assuming $m \geq 3$, while $A_2 = \text{mes}(\Omega) - A_1$ is the area outside this polygon.

Model 5: Convex Segmentation Models

The work of Chan et al. (2006) proposes a convex relaxation idea for the Chan-Vese model that is nowadays widely used, where the relaxation consists in replacing $\{0, 1\}$ by $[0, 1]$ after substituting $H(\phi)$ by u . This idea is emphasized by Spencer and Chen (2015) who propose a convex selective segmentation model

$$\min_{\substack{u, c_1, c_2 \\ 0 \leq u \leq 1}} \int_{\Omega} dg(|\nabla z|)|\nabla u|d\Omega + \int_{\Omega} (\lambda_1 u|z - c_1|^2 + \lambda_2 (1 - u)|z - c_2|^2) d\Omega + \theta \int_{\Omega} P_d u d\Omega$$

where P_d is the scaled distance to the polygon \mathcal{P} formed by \mathcal{M} and specifically $P_d = 0$ if (x, y) belongs to the polygon \mathcal{P} , encouraging $u = 0$ in $\Omega \setminus \bar{\mathcal{P}}$. The model works well only if θ is appropriately chosen which may not be always easy to do.

Model 6: Convex Models Based on Geodesic Distances

A deep idea to explore more the marker set \mathcal{M} is to design a new distance function $d(x, y)$ that takes into account several factors: the set \mathcal{M} itself as before, large edges of image z , the previously used Euclidean distance P_d , and possible anti-markers (to define a set \mathcal{AM} of points that are definitely not in the intended object). Then a geodesic distance encompassing all these constraints and denoted by \mathcal{D} is defined in Roberts et al. (2019) to replace the previous distance d and satisfies the Eikonal-type equation:

$$|\nabla \mathcal{D}| = f(x, y) = \varepsilon + \beta |\nabla I|^2 + \nu P_d + d_{AM},$$

where ε is a small parameter; I is a denoised version of z , which can be the smoothed image $G * z$ or the image resulting from a few iterations of a denoising process applied to z ; P_d is the Euclidean distance as used before; and d_{AM} is an anti-marker distance exhibiting the following properties: it penalizes pixels close to the set \mathcal{AM} and it ensures rapid decay of the penalty away from the set \mathcal{AM} . The design of d_{AM} may be based on a geodesic distance \tilde{d}_{AM} satisfying $|\nabla \tilde{d}| = \varepsilon + \beta |\nabla I|^2 + \nu P_d(\mathcal{AM})$. Then in Roberts et al. (2019), it is suggested to take $d_{AM} = (\exp(\alpha(1 - \tilde{d})) - 1) / (\exp(\alpha) - 1)$ which can highlight the contribution of \mathcal{AM} while reducing its influence on \mathcal{M} .

Note that the Eikonal equation is equipped with Dirichlet boundary conditions (i.e., $\tilde{d} = 0$ at \mathcal{M}), thus falling within the framework of boundary value problems, and can be solved efficiently by a $O(N)$ implementation of the fast marching algorithm (see Yatziv et al. 2006).

It yields the following model Roberts et al. (2019)

$$\begin{aligned} \min_{u, c_1, c_2} \int_{\Omega} g(|\nabla z|) |\nabla u| d\Omega &+ \int_{\Omega} (\lambda_1 u |z - c_1|^2 + \lambda_2 (1 - u) |z - c_2|^2) d\Omega \\ &+ \mu \int_{\Omega} \mathcal{D} u d\Omega + \alpha \int_{\Omega} v(u) d\Omega \end{aligned}$$

where $v(u)$ enforces $0 \leq u \leq 1$ as done in Chan et al. (2006). The shifting of \mathcal{D} from the first component to a separate term was motivated by Liu et al. (2018).

While Model 6 is perhaps the most robust up to now, it still assumes that the underlying given image is approximately of piecewise constant intensities and without textures, just as in the Chan-Vese model (Chan and Vese 2001). To allow more generality, there are scopes and needs to design new models.

Other Possible Models

There exist many interesting ideas that remain to be fully explored.

(i). Assuming that a user has provided 2 sets of input as before, markers \mathcal{M} within the object and anti-markers \mathcal{AM} within the background, (Cremers et al. 2007) define a distance-like label function

$$L(x, y) = \begin{cases} +1 & \text{if } (x, y) \in \mathcal{M}, \\ -1 & \text{if } (x, y) \in \mathcal{AM}, \\ 0 & \text{elsewhere,} \end{cases}$$

which may be used to replace or enhance our distance function. The suggestion in Cremers et al. (2007) is to add a regularization term to influence optimization for the level set function (object $\phi > 0$)

$$E_{\text{user}}(\phi) = - \int_{\Omega} L(x, y) \text{sign}(\phi(x, y)) d\Omega.$$

This idea is directly related to our works as demonstrated.

(ii). To ensure that an indicator function based on user input can re-adjust any classified (segmented) pixels, Ben-Zadok et al. (2009) suggested the new definition

$$L(x, y) = H(\phi) + [1 - 2H(\phi)] \int_{z \in \mathcal{N}(x, y)} M(z) dz$$

where

$$M(z) = \sum_{i=1}^m \delta(z - (x_i, y_i)) \quad \text{with} \quad (x_i, y_i) \in \mathcal{M}$$

and $\mathcal{N}(x, y)$ denotes an infinitesimal neighborhood of (x, y) . Here, $L(x, y) = 0$ for pixels in set \mathcal{M} and 1 for pixels in set \mathcal{AM} . However $L(x, y) = H(\phi(x, y))$ at other pixels away from markers. The new H that can reflect the feedback of a user is found by

$$E_{\text{user}}(\phi) = \int_{\Omega} \int_{\Omega} (L(x', y') - H(\phi(x, y)))^2 K(x, x', y, y') d\Omega d\Omega'$$

where K is a Gaussian kernel defined by

$$K(x, x', y, y') = \frac{1}{2\pi\sqrt{|\Lambda|}} \exp \left\{ -\frac{1}{2} (x - x' \ y - y') \Lambda^{-1} \begin{pmatrix} x - x' \\ y - y' \end{pmatrix} \right\}$$

and Λ is a 2×2 covariance matrix. In this method, only the set \mathcal{AM} is required because all pixels in this set have to be fixed, i.e., if such pixels from the set are in the foreground, they will be assigned to background and vice versa through matching $H(\phi)$ to L . This idea, mainly directed to post-processing steps, is a bit different from other formulations.

(iii). The current selection models, as discussed so far, assume that a given image exhibits a piecewise constant distribution of intensities (due to their use of Chan-Vese-like fitting terms). There exist newer and convex models (Alberti et al. 2003; Cai et al. 2013) derived from the Mumford-Shah (Mumford and Shah 1989) model, well-suited for more general images. Hence the models depicted in this section and dedicated to geometric constraint enforcement may be extended to such works.

(iv). Geometric constraints may be phrased in other forms apart from a set \mathcal{M} . An interesting (and challenging) problem is to define a shape constraint, often useful when one intends to overcome problems related to missing data due to occlusions (e.g., in cells imaging, medical imaging, or vehicles recognition). See works in Thiruvankadam et al. (2008), Fuzhen and Xuhong (2010), and Kihara (2016). Such methods may involve ideas borrowed from image registration requiring small deformations or large rigid deformations.

(v) Another interesting class of research directions falls within the scope of boundary convexity constraint. This helps segmentation in case of very noisy images or missing data. See the recent works of Liu et al. (2020), Luo et al. (2019), and Siu et al. (2020).

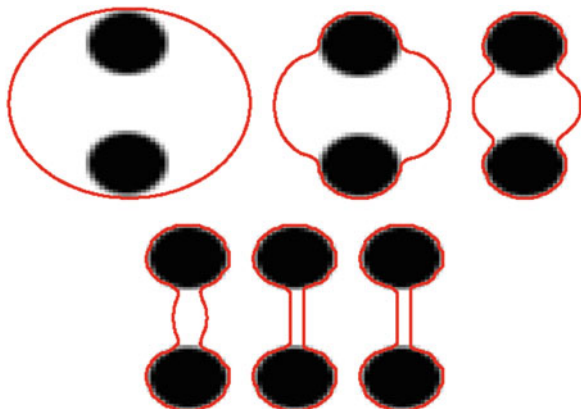
Finally it should be noted that the above discussed models can be extended in a simple manner to deal with 3D images; in fact many were implemented and tested in 3D. As also remarked above, it remains to develop models based on frameworks beyond piecewise constant intensities that can process texture images effectively. This will be a future direction.

Topological Prior Knowledge

The second class of methods we depict hereafter intends to partition a given image into semantically significant constituents while fulfilling some topological requirements. As stressed in Ségonne et al. (2007), integrating such constraints is a difficult task for two reasons. First, due to the dual nature of topology which is both a global property and a local one, small and localized changes on a geometrical shape may modify its global connectivity. Second, topology is a continuous concept whose properties are difficult to transpose in the discrete setting. Two different kinds of prior knowledge are investigated:

- (a) prescribed topology enforcement in the sense that the segmented target should be homeomorphic to the original shape supplied by the user—two objects being homeomorphic provided they can be deformed into each other by a continuous, invertible mapping—or should exhibit a prescribed number of connected components/holes. Topology enforcement in segmentation is particularly important when a user's requirement is not in visual agreement with the data, i.e., in the case where, without including those topological constraints, most segmentation models would fail: Fig. 3 would show two objects, while a single cell would be outlined in Fig. 4.

Fig. 3 Segmentation steps of a synthetic image with two disks when topological constraints are applied



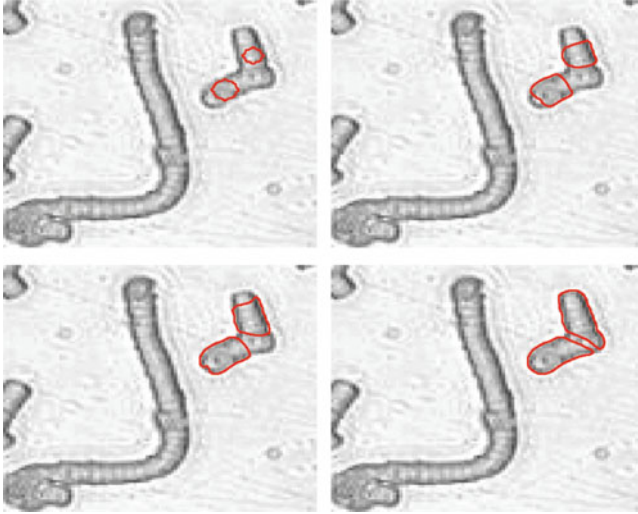


Fig. 4 Segmentation steps of two blood cells close to each other when topological constraints are applied

Figures 3 and 4 taken from Le Guyader and Vese (2008) provide two illustrations of prescribed topology enforcement. In Fig. 3, we aim at segmenting the two disks while maintaining the same topology throughout the process, meaning that we expect to get a simply connected shape. Figure 4 illustrates the case where the initial condition is made of two disjoint closed curves. We expect to have both curves evolving without merging. If visually the blood cells look glued, individual cell segmentation is required. Here, the final segmentation shows that the two cells are disconnected, in compliance with the user's requirement.

- (b) regularity enforcement on the edge set of the segmentation, thus influencing the topology of the segmenting curves/shapes, with an emphasis on variational models, due to their ability to include multiple criteria. This kind of approach does not fall exactly within the scope of topological prior-based methods—since it does not intend to prescribe the topology of the targeted object—but influences nevertheless the regularity and so in some way the topology of the final shape by removing undesirable small patterns. In this regard, Fig. 5 taken from Alvarez et al. (2018) (courtesy of Luis Alvarez, Universidad de Las Palmas de Gran Canaria, Spain) illustrates how a geometrical partial differential equation can be used for level set regularization, i.e., to remove small-scale features and spurious oscillations. By choosing adequately a forcing term appearing in the partial differential equation (PDE) that dictates the front evolution, one can keep more or less detail in the final segmenting curve.

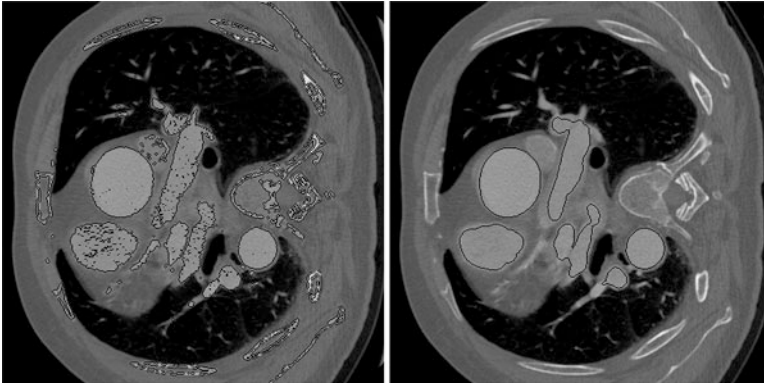


Fig. 5 On the left, the initial condition Γ_0 is obtained by thresholding the intensity values of the image I , i.e., $\Gamma_0 = \partial \{x \mid c_1 \leq I(x) \leq c_2\}$, c_1 and c_2 being two fixed values. On the right, obtained regularized segmentation

Topology Prescription

The necessity of designing topology-preserving processes arises in many applications, e.g., in medical imaging to preserve the contextual relations between organs or when reconstructing, for instance, the human brain (Chen and Freedman 2011). Despite its highly folded nature (Fischl et al. 2001; Ségonne et al. 2007), the intrinsic unfolded structure of the cortex is the one of a 2D sheet, and assuming that the midline hemispheric connections are artificially closed, each cortical hemisphere is homeomorphic to the sphere, implying that this topological feature must be fulfilled by the segmentation. In the context of curve evolution-based models and more specifically level set-based approaches (Osher and Sethian 1988), two main channels of thought have been investigated: methods resting on digital topology (Boutry et al. 2018; Kong and Rosenfeld 1989), in particular on the concepts of simple point, multisimple point, and well-composedness, and methods relying on specifically designed criteria in the objective function (purely continuous ones), penalizing curvature and thus affecting the shape of the curve.

Digital Topology

It is generally agreed that the implicit framework of the level set setting displays several advantages over parametric methods when tracking a front that propagates: the evolving contour is embedded in a higher dimensional level set function, thus avoiding parameterization issues; the model is intrinsic, i.e., invariant to a reparameterization of the curve, able to handle topological changes (merging and splitting—note nevertheless that some works aim to reconcile parametric implementations with handling of topological changes as in Precioso et al. (2002), for instance, where the authors address the issue of moving object segmentation

using interpolating B-spline curves in a spatial multi-resolution approach and with a penalty over the length of the contour. Topological changes are managed by making the most of the variation diminishing property exhibited by B-splines); geometrical properties of the front such as curvature or outward unit normal vectors are easily derived from the level set function; and its evolution is straightforwardly phrased in an Eulerian framework. Nevertheless, this topological flexibility proves to be undesirable in many applications (e.g., mitotic cell tracking Geiping 2014). That is to say, parametric methods seem naturally more suited to deal with these topological constraints. Nevertheless, the level set formulation remains widely used in topology-preserving segmentation models, partly due to its popularity and the fine properties it exhibits. The topology enforcement takes the form of constraints applied to the implicit contour to fulfill the user's requirements.

In the spirit of front propagation approaches combined with digital topology requirements (reconciling then the intrinsic discrete nature of digital images with the continuous representation with which they are identified in variational frameworks), (Han et al. 2002; Han and Xu 2003) propose a model preserving the topology of the implicit contour while the embedding level set function is evolving. The key idea of the model lies in the concept of simple points (Bertrand 1994, 1996) identified as those points whose addition or removal leave the topology unchanged. Their characterization relies on the computation of two topological numbers: the equivalence between the fact for a point of being simple, and the specific values of these two topological characteristics is proved in Bertrand (1994). It is complemented by the basic definitions necessary to effectively calculate these two numbers. The algorithm reads as follows, the topology of the zero level set being assumed equivalent to the topology of the boundary of the digital object it defines: at each iteration, one monitors the changes of sign of the level set function and prevents it from changing sign at grid points that are not simple. The derived algorithm is thus pixel-based, applicable on lattice structures only, not on arbitrary data sets, and the result produced is highly dependent on the order in which the points are treated in the narrow band.

Still in a level set evolution framework, Ségonne (2008) investigates the concept of multisimple points identified as those points whose addition or removal do not create or delete handles in the image. The formal mathematical characterization of these points—derived from the one of simple points—can be found in Ségonne (2008, Subsection 3.1). Unlike Han et al. (2002) and Han and Xu (2003), the resulting algorithm allows connected components to merge, split, or vanish, ensuring at the same time that the genus (topological invariant describing the number of handles) of the initial active contour is preserved, and relaxes then the constraints on the initial condition.

Contrasting with these concepts of simple points and multisimple points, Tustison et al. (2011) propose a topological variant of well-composedness for maintaining the topology of the evolving digital front—in n dimensions, a set being said well-composed if and only if the boundary of its continuous counterpart is a $(n - 1)$ -manifold (Boutry et al. 2018), thus obviating the requirement of specifying one of the classical connectivity relations since in 2D it reduces to the connectivity

(4,4), ((6,6) in 3D). This characterization implies that a correspondence between a n -D digital set and the boundary surface of its continuous analog is required. This correspondence is made explicit in [Latecki](#), Section 1. This connectivity condition alone is not sufficient to constrain the topology of the evolving digital front, thus requiring the extension of the concept of well-composedness to its topological variant by incorporating critical configurations. In this regard, [Tustison et al. \(2011\)](#) identify points that preserve both the well-composedness property and the image topology, sole points that can be added or removed while the front progresses. Once the evolution of the initial configuration subject to strict topology preservation via topological well-composedness is achieved, and since the interface that separates the well-composed genus zero components satisfies the digital Jordan separation theorem—discrete counterpart of the Jordan separation theorem that states that a surface homeomorphic to the sphere might be separated into two objects by a simple, continuous, closed curve (*Jordan curve*) along the surface ([Nakahara 2003](#))—a procedure to glue together the neighboring components by adding the part of the Jordan surface separating them is initiated, yielding a single well-composed genus zero object.

In the context of cellular morphology analysis, [Yu et al. \(2010\)](#) address the critical issue of individual cell segmentation. This task is challenging since a cell may stick to other ones, sharing common boundaries, or even overlap others with no clear outline. The authors propose a novel algorithm (that constitutes again a compromise between discrete and continuous formulations): once the nuclei of the cells are identified and segmented, a front propagation is performed using generalized Voronoï diagrams to avoid overlaps.

In a purely discrete setting, [Chen et al. 2011](#) propose incorporating topological prior knowledge into random field image segmentation to encode more global topological properties such as connectedness of a labeled region. The labeling functional is classically designed as the sum of unary potentials and pairwise potentials and is subject to topological constraints prescribing the number of connected components and holes. [Waggoner et al. \(2015\)](#) develop a new Markov random field based multi-labeling technique to enforce topology in multi-label image segmentation. They apply both specific adjacency relations between each pair of segments and a connectedness property on each region. More precisely, the algorithm consists in defining a segmentation template exhibiting the desired topology and in spreading it toward a target image.

Among a posteriori (also termed as retrospective) topology correction methods, one can cite the work by [Ségonne et al. \(2007\)](#) dedicated to geometrically accurate topology correction of cortical surfaces. Denoting by C the cortical surface—typically instantiated as a polygonal tessellation with vertices, edges, and faces—by S a sphere and by $N : S \rightarrow C$ a mapping from S to C , their work is based on the observation that a necessary and sufficient condition for C to carry the desired spherical topology is that the mapping N is a homeomorphism. Worded in these terms, the problem is difficult since large deformations are required to retrieve the less smooth highly convoluted surface. The authors propose instead to find a mapping $M : C \rightarrow S$. If the cortical surface carries the desired topology, i.e.,

$\chi(C) = 2$, $\chi(C)$ denoting the Euler characteristic which is a topological invariant, nothing must be done. If not (which is most often the case in practice since C is likely to exhibit topological defects), the proposed algorithm (also inspired by Fischl et al. 2001) operates as follows. The authors first seek a mapping $M : C \rightarrow S$ which is a quasi-homeomorphism, i.e., a homeomorphism on as much of the surface as possible. This step is achieved by unfolding and smoothing the folded cortical surface by spherical inflation so that the obtained surface resembles the one of a sphere with origin the centroid of the initial surface. The position of a given vertex is evolved according to a geometric flow involving movement toward the centroid of its neighboring vertices, while projecting out the average inward movement it creates over the whole surface, with in addition a radial term driving each vertex to the surface of a sphere with prescribed radius. Once this spherical inflation step is achieved, a quasi-homeomorphic mapping M is generated by minimizing a function that penalizes regions in which the determinant becomes negative or zero. Topological defects are then detected as locally noninvertible regions (where M^{-1} is multivalued), i.e., as parts of the sphere displaying overlapping triangles, and the correction procedure is applied, taking into account geometrical accuracy (expected local curvature, local intensity distribution) and topological consistency. More precisely, correcting a topological defect amounts to identifying the number of handles (or equivalently holes) it contains, a handle being characterized by the existence of non-contractible curves (also termed as non-separating loops), simple closed curves that cannot be continuously deformed on the manifold into a single point. This concept is of importance since a non-separating loop associated to a hole gives two strategies to remove this hole: either one fills the area enclosed by the non-separating loop to patch the hole or one empties the area inside to open the hole. Motivated by this characterization, the algorithm reads as follows. Given a handle, several non-separating loops are randomly generated. For each produced curve, the faces that form the loop are removed from the topological defect mesh, and the resulting open mesh is sealed (either by filling the hole or by cutting the handle). The accuracy of the resulting candidate solution is optimized, based on active contour patches. In the end, the candidate configuration that optimizes a Bayesian energy functional is selected, both maximizing goodness of fit of the produced surface with respect to the available image information and complying with topological consistency.

In the same vein as in Ségonne et al. (2007) and motivated by the fact that surface reconstruction methods go beyond the simple use of volumetric images alone in the context of structural and functional brain data analysis, Yotter et al. (2011) propose to retrospectively repair topological and geometrical defects (that impair the true nature of the cortical anatomy) on the brain surface mesh, using spherical harmonics, this theory allowing to quantify structural differences between shapes. Spherical harmonics are functions defined on the surface of a sphere. They form a complete set of orthogonal functions on the sphere and might be used to represent functions defined on the surface of a sphere in the same spirit as Fourier series.

The processing chain reads as follows: (i) The uncorrected surface mesh is mapped onto a sphere (this mapping is not a homeomorphism) meshed with triangles. (ii) A regularly sampled grid is then overlaid on the previously obtained

sphere surface. (iii) For each regularly sampled spherical point, one identifies the intersecting triangle of the sphere mesh (by finding the closest triangle of the tessellated sphere surface, a specific strategy being developed to favor some of the triangles related to topological defects). Then using barycentric coordinates in this triangle, the coordinates of the fiber vertex lying on the original mesh surface are determined, yielding three functions defined on the sphere: each regularly sampled spherical point is associated with the three coordinates giving its location on the original cortical surface. (iv) Every function is expanded in the spherical harmonic basis where the coefficients are defined as the L^2 -inner product of the function and the basis functions. Using the computed coefficients, two surfaces are then reconstructed: the first one is a high-frequency surface employing all coefficients, while the second one is a smooth surface reconstructed from filtered coefficients using a low-pass filter. Vertices from the low-pass filtered reconstruction are patched into the high-frequency one in regions that previously contained defects (and that are likely to display pikes after the spherical harmonic-based reconstruction). At last, a post-processing step is applied to correct self-intersections.

Despite the fact that deep learning-based methods are beyond the scope of this contribution, we would like to end this part by highlighting some methods that intertwine the soundness of these approaches (that yield remarkable results whenever sufficient labeled data can be collected) with variational models and specially active contour-based models, capable of encoding high-level shape features such as topology. In Thierbach et al. (2018), Thierbach et al. propose to combine convolutional neural networks with topology-preserving geometric deformable models (Bogovic et al. 2013) in the context of neural cell bodies segmentation from light sheet microscopy, while limiting manual annotations. The training step is achieved with simple cell centroid annotations and the final segmentation provides accurate results complying with the topological requirements (no cell splitting/merging).

Purely Continuous Methods

If the above methods make a trade-off between the intrinsic discrete nature of digital images and the continuous formulation of front propagation, some take the side of focusing only on continuous aspects. In Sundaramoorthi and Yezzi (2005), the authors incorporate a novel nonlocal geometric flow into image-based evolutions of active contours in order to preserve topology. The relevant term that is minimized with respect to the curve C is inspired by the knot energy and is defined by $E(C) = \frac{1}{2} \int \int_{C \times C} \frac{dp dp'}{\|C(p) - C(p')\|^\gamma}$ ($\|\cdot\|$ denoting the Euclidean norm and γ being a tuning parameter), thus penalizing spatial proximity of the curve points and subsequently its curvature.

The work Alexandrov and Santosa (2005) introduces a curve evolution method based on level sets for shape optimization models arising in material science and is directed toward the class of problems involving constraints on the number of connected components. The algorithm is designed in order for the narrow band of the evolving contour to avoid overlaps. More precisely, their model minimizes $F_\mu(\Phi) = F(\Phi) + \mu R(\Phi)$ with $\mu \ll 1$, Φ being the evolving level set function assumed to be a signed distance function, F being a general shape optimization

functional, while $R(\Phi)$ is the topological constraint. As previously sketched, this constraint that includes shape topology (i.e., prescribed number of connected components and holes), component size, and distance between components ensures that the evolving topology is equivalent to the initial one—the prior on the initial topology is thus strong—and is incorporated into the optimization problem via a logarithmic barrier technique as

$$R(\Phi) = - \int_{\partial D} \log [\Phi(x + d \nabla \Phi(x))] ds - \int_{\partial D} \log [-\Phi(x - l \nabla \Phi(x))] ds,$$

with $D = \{x \mid \Phi(x) > 0\}$ and $d > 0, l > 0$ given parameters. Parameters d and l influence on the distance between distinct connected components and on the size of the connected components themselves. Although devoted to a different application, the work Rochery et al. (2006) uses a similar idea to the one developed in Sundaramoorthi and Yezzi (2005) to prevent pieces of the same curve from colliding, merging, or breaking. The goal is to track thin long objects that evolve, with applications to the automatic extraction of road networks in remote sensing images. The authors propose interesting nonlocal regularizations on the curve C parameterized by $p \in [0, 1]$ phrased as $E(C) = - \int_0^1 \int_0^1 \mathbf{t}(p) \cdot \mathbf{t}(p') \Psi(\|C(p) - C(p')\|) dp dp'$, with $\mathbf{t}(p)$ denoting the tangent vector to the curve at point $C(p)$ and $\|C(p) - C(p')\|$ being the Euclidean distance between the curve points $C(p)$ and $C(p')$. The function Ψ is chosen to be $\Psi(l) = \sinh^{-1}(1/l) + l - \sqrt{1 + l^2}$, thus decreasing on $[0, +\infty[$. Other nonlocal forms are considered as well, and geometric motions of thin long objects are obtained. The implicit representation by level sets is used for the implementation. In Rochery et al. (2005), the authors carry on their ideas but, this time, in a phase field approach.

Still in the prospect of a local treatment of topology preservation, Cecil (2003, Section 4) is dedicated to the tracking of interfaces with fixed topology. The model relies on a coupled system of PDEs involving the level set function φ embedding the propagating front, and the arclength function Ψ — Ψ being conjugate to φ in the sense that the two form an orthogonal coordinate system on the zero level set of φ — and on an accurate estimate of the Jacobian J of the interface function and its conjugate (J tends to 0 at merge points, while it tends to ∞ at pinch points). Motivated by geometric considerations, Le Guyader and Vese (2008) complement the classical geodesic active contour model by a nonlocal component interpreted as a repelling force. More precisely, the level set function Φ being assumed to be a signed distance function to the evolving contour C and $l > 0$ denoting a tuning parameter, the following functional is incorporated into the classical geodesic active contour model phrased in the level set framework:

$$E(\Phi) = - \int_{\Omega} \int_{\Omega} G(\|x - y\|^2) \langle \nabla \Phi(x), \nabla \Phi(y) \rangle H(\Phi(x) + l) H(l - \Phi(x)) H(\Phi(y) + l) H(l - \Phi(y)) dx dy,$$

the potential function G measuring the closeness of the two points x and y , H denoting the 1D Heaviside function. Again, the goal is to penalize spatial proximity of curve points belonging to a narrow band around the zero level set and subsequently the curvature of the level lines. This idea is then revisited in Schaeffer and Duggan (2014) in the context of region-based active contours.

An energy involving a fixed-width band around the evolving curve as in Le Guyader and Vese (2008) is also introduced in Mille (2009) to achieve a proper trade-off between local features of gradient-like terms and global region characteristics, and to weaken the strong assumption of uniformity of intensity over regions classical region-based models rely on.

More recently, still in an effort to ensure orientation preservation and to address the issue of stability with respect to noise—for instance, when the shapes exhibit multiple disjoint objects that should be viewed as a whole—a new framework based on quasiconformal mappings has been introduced in Chan et al. (2018). Given an image containing an object to be segmented together with the desired prescribed topology (that can be viewed as a shape prior), a simple template image is deformed so that it matches the boundary of the target object. In this regard, the model may be seen as a joint segmentation/registration one. The deformation undergone by the moving shape is dictated by the Beltrami equation and relies on the fine properties of quasiconformal mappings. Quasiconformal mappings can be defined as follows (Lehto and Virtanen 1973) (*we restrict ourselves to quasiconformal mappings that are homeomorphisms between plane domains*):

A sense-preserving homeomorphism f of the domain G is called quasiconformal if its maximal dilation $K(G)$ is finite. If $K(G) \leq K < \infty$, then f will be called K -quasiconformal.

As from the one hand, the maximal dilatation of a non-conformal sense-preserving homeomorphism is always greater than 1, and from the other hand, if f is conformal, $K(G) = 1$, $K(G)$ can be viewed as a measure of deviation from conformality. The following result gives a necessary and sufficient condition ensuring that a given homeomorphism $f : \Omega \rightarrow \Omega'$ in $W_{loc}^{1,2}(\Omega)$ is K -quasiconformal.

Theorem 1 (Astala et al. 2009, Theorem 2.5.4). *Suppose $f : \Omega \rightarrow \Omega'$ is a homeomorphic $W_{loc}^{1,2}$ -mapping. Then f is K -quasiconformal if and only if*

$$\frac{\partial f}{\partial \bar{z}}(z) = \mu(z) \frac{\partial f}{\partial z}(z) \text{ for almost every } z \in \Omega,$$

where μ called the Beltrami coefficient of f is a bounded measurable function satisfying

$$\|\mu\|_{\infty} \leq \frac{K-1}{K+1} < 1.$$

With $\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - i \frac{\partial f}{\partial y} \right)$ and $\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right)$, setting $f(z) = f(x + iy) = u(x, y) + iv(x, y)$ (so in the context of registration, the sought deformation is $\varphi = (u, v)^T$), the Jacobian is defined by $J_f(z) = \left| \frac{\partial f}{\partial z}(z) \right|^2 - \left| \frac{\partial f}{\partial \bar{z}}(z) \right|^2 = \frac{\partial u}{\partial x}(x, y) \frac{\partial v}{\partial y}(x, y) - \frac{\partial u}{\partial y}(x, y) \frac{\partial v}{\partial x}(x, y) = \det \nabla \varphi(x, y)$. Consequently, if $f, W_{\text{loc}}^{1,2}$ -homeomorphism, is K -quasiconformal,

$$J_f(z) = \left| \frac{\partial f}{\partial z}(z) \right|^2 - \left| \frac{\partial f}{\partial \bar{z}}(z) \right|^2 = \left| \frac{\partial f}{\partial z}(z) \right|^2 \left(1 - |\mu(z)|^2 \right),$$

entailing that $J_f(z) = \det \nabla \varphi(x, y) > 0$ a.e., since $\|\mu\|_\infty < 1$.

Ensuring boundedness of the Beltrami coefficient ($\|\mu\|_\infty < 1$) thus implies positivity of the related deformation Jacobian determinant.

Equipped with this material, the authors state the main theoretical result that says that for any object in \mathbb{C} with arbitrary topology, there exists a homeomorphism between the object and the simple circular domain with the same topology, which motivates the introduction of the concept of Beltrami representation of shapes. Given a natural integer $g \in \mathbb{N}$, a g -holed circular domain, $\widehat{D}_g \subset \Omega \subset \mathbb{C}$, and a shape D with the same topology as \widehat{D}_g , as there exists a quasiconformal mapping $f^\mu : \mathbb{C} \rightarrow \mathbb{C}$ associated with Beltrami coefficient μ such that $f^\mu(\widehat{D}_g) = D$ and $f^\mu = \text{Id}$ in $\mathbb{C} \setminus \Omega$, the shape D can thus be represented by μ , which is called the Beltrami representation of D . It yields the following minimization problem relying on the Beltrami representation of shapes. Note that the Beltrami representation is defined by the deformation from \widehat{D}_g to D . Denoting by $\Omega \subset \mathbb{C}$ an image domain, by $I : \Omega \rightarrow \mathbb{R}$ an image including an object $D \subset \Omega$, D being supposed of genus g , by $J : \Omega \rightarrow \mathbb{R}$ an image of \widehat{D}_g , called topological prior and modeled as a two-phase partition defined by:

$$J(z) = \begin{cases} c_1 & \text{if } x \in \widehat{D}_g, \\ c_2 & \text{if } x \in \Omega \setminus \widehat{D}_g, \end{cases}$$

the authors propose minimizing the following energy with respect to μ :

$$E(\mu) = \int_{\Omega} |\mu|^2 + \eta \int_{\Omega} (I \circ f^\mu - J)^2 + \lambda \int_{\Omega} |\nabla \mu|^2,$$

subject to Dirichlet boundary condition $\mu = 0$ on $\partial\Omega$ and inequality constraint $\|\mu\|_\infty < 1$ on Ω . The model was extended in Zhang and Chen (2018) to impose $\|\mu\|_\infty < 1$ and without having to computing μ directly while allowing the use of a converging Gauss-Newton method.

Thus the optimization problem encompasses three components: while the pairing of the intensities between I and J is ensured by the second term of the

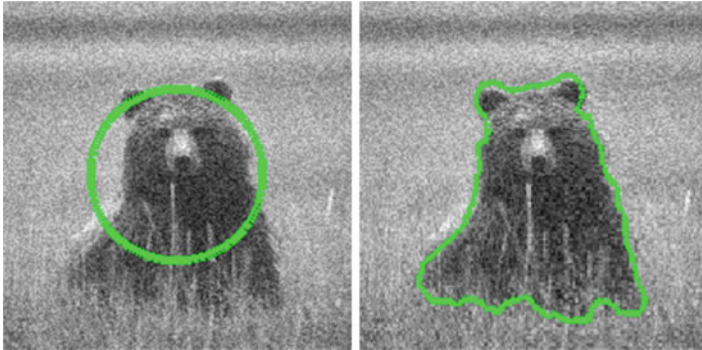


Fig. 6 On the left, prescribed topology superimposed on the input image. On the right, obtained segmentation

functional, the last one guarantees some smoothness on the deformation mapping f^μ whereas the constraint on $\|\mu\|_\infty$ insures admissibility of the Beltrami representation of the object in D . The connection to the Chan-Vese data fidelity term can be made explicit by making a simple change of variable in the second component. Also the regularization of the Beltrami coefficient is now a substitute for the classical shape perimeter minimization. High accuracy and stability of the proposed model with different segmentation tasks are then exemplified as demonstrated in Fig. 6 in which we try to preserve the 0-genus property of the shape—note nevertheless that the algorithm can deal with shapes of higher genres; for instance, if we go back to the example of Fig. 3, it would suffice to build a proper topological prior image J made of two separate shapes. Classical segmentation models would fail to segment the desired shape as a whole because of the visible occluded regions: the bear ears would be disconnected from the body.

Regularization Enforcement on the Evolving Front

Although not directly related to topological prior-based methods, models relying on regularity enforcement prove to be efficient to remove undesirable small patterns and oscillations. In that, they influence both the global and local topology of the recovered shape and are connected in some way to the models depicted in section “[Topology Prescription](#)”.

Regularization by Geometric Flows

Controlling the asymptotic states of geometric equations with respect to a prescribed forcing term is the subject of Alvarez et al. (2018), in the context of level set regularization. With suitable forcing terms, stabilization in a finite time of radial solutions can be demonstrated, making this modeling relevant for nonlinear image filtering or segmentation, in order to remove, for instance, spurious oscillations

or small-scale objects, to control the size of an object or on the contrary to enforce merging. Beyond the theoretical study of particular geometrical partial differential equations it includes, this work intends to provide some insight on how to build PDEs in order to solve image-related problems, whether it be segmentation or filtering. More precisely, the considered PDEs are geometrical ones—these equations defining a hypersurface evolution—among which parabolic perturbed mean curvature-based equations of the form:

$$\begin{cases} \frac{\partial u}{\partial t} = F(\nabla u, \nabla^2 u) + k(x) |\nabla u| \text{ in } (0, T) \times \mathbb{R}^n, \\ u(0, x) = u_0(x) \text{ on } \mathbb{R}^n, \end{cases} \quad (1)$$

with $F(\nabla u, \nabla^2 u) = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) |\nabla u|$, this latter quantity being the mean curvature (thus ensuring smoothness of the evolving curve), while k denotes a forcing term, k being assumed bounded Hölder continuous (so with rather weak required regularity—only Hölder continuity in comparison to the classical Lipschitz regularity). This function k that may depend on the image contents is the parameter that can be adjusted by the user according to his needs. Note also that this class of PDEs falls within a broader framework studied in Giga et al. (1991), for instance. The geometric feature of these PDEs is conveyed by the fact that $F : \mathbb{R}^n \setminus \{0\} \times \mathcal{S}^n \rightarrow \mathbb{R}$ satisfies $F(\lambda p, \lambda X + \sigma p \otimes p) = \lambda F(p, X)$, $\forall \lambda > 0$, $\forall \sigma \in \mathbb{R}$, $\forall p \in \mathbb{R}^n$, \otimes denoting the tensor product in \mathbb{R}^n and \mathcal{S}^n denoting the space of $n \times n$ real symmetric matrices. It expresses that the zero level set of function u only depends on the zero level set of the initial condition and not on the initial condition itself, and the composition of any solution with a nondecreasing function remains a solution of the equation. Such PDEs fall within the framework of the viscosity solution theory (Crandall et al. 1992). Theoretical issues/qualitative properties of the hypersurface evolution are investigated like comparison principle or existence/uniqueness of the solution, with fewer requirements on function k (Hölder continuity only). Special care is taken to the qualitative properties of the hypersurface evolution. This analysis is fully meaningful once we have studied the shape of radial solutions, making it possible to derive some qualitative properties (asymptotic behavior) of the (unique) solution of the problem associated with an unspecified (but smooth enough) initial condition.

Considering as initial hypersurface Γ_0 of \mathbb{R}^n the boundary of a bounded open set U_0 , and denoting by $u_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ any bounded uniformly continuous function (Lipschitz continuous if k is merely Hölder continuous) that satisfies:

$$u_0(x) = \begin{cases} u_0(x) < 0 & \text{if } x \in U_0 \\ u_0(x) > 0 & \text{if } x \in (\mathbb{R}^n - \overline{U_0}) \\ 0 & \text{if } x \in \Gamma_0 = \partial U_0, \end{cases}$$

we define the sets $\{U_t\}_{t \geq 0}$ and $\{\Gamma_t\}_{t > 0}$ as

$$U_t = \{x \in \mathbb{R}^n, u(t, x) < 0\}$$

and

$$\Gamma_t = \partial U_t,$$

with $u := u(t, x)$ the unique viscosity solution of (1) for the initial datum $u_0(x)$. The authors first prove that U_t and Γ_t are independent of the choice of u_0 and only depend on U_0 . Then they provide a comparison principle saying that if U_0 and \widehat{U}_0 are bounded open sets satisfying the inclusion relation $U_0 \subseteq \widehat{U}_0$, then for $t \geq 0$, this inclusion relation still holds, i.e., $U_t = \{x \in \mathbb{R}^n, u(t, x) < 0\} \subseteq \widehat{U}_t = \{x \in \mathbb{R}^n, \hat{u}(t, x) < 0\}$, u and \hat{u} being the unique solutions of the PDE associated, respectively, with initial condition u_0 and \hat{u}_0 . This later result will enable us to infer the asymptotic behavior of an evolving shape given an initial condition. In order to get a clear view of the asymptotic states of the solution of (1) according to the choice of the forcing term k , the authors focus on the radial solution shape analysis. In that purpose, let $U_0 = B_{r_0}(x_0)$ be the ball centered at x_0 with radius r_0 , and let us choose as initial datum u_0 , the signed distance function defined by:

$$u_0(x) = d_{B_{r_0}(x_0)}(x) = \begin{cases} |x - x_0| - r_0 & \text{if } |x - x_0| < r_0 + S \\ S & \text{otherwise,} \end{cases}$$

for some $S > 0$. Standard computations lead to the following equation in radial coordinates:

$$\frac{\partial u}{\partial t}(t, r) = \left(\frac{n-1}{r} \operatorname{sgn}(u_r(t, r)) + k(r) \right) |u_r(t, r)|, \tag{2}$$

with sgn being the sign function. In this case, U_t is given by the ball $B_{r(t)}(x_0)$ where $r(t)$ satisfies $u(t, r(t)) = 0, \forall t > 0$. By deriving this relation with respect to t and by substituting $\frac{\partial u}{\partial t}$ with the right-hand side of (2), the problem amounts to solving an ordinary differential equation in r for which an explicit expression of the solution can be provided. The shape of $r(t)$ is then investigated for several choices of the forcing term k as well as its asymptotic behavior, which suggests that by choosing the forcing term k properly, some stabilization properties of the propagating front (the radius may stabilize in a finite time) can be expected as well as some particular responses such as shrinkage of the shape or on the contrary expansion of the radius with time. These observations combined with the monotonicity principle mentioned above (preservation of the inclusion property) motivate the application of this model as level set regularization (refer to Fig. 5 for an application).

As an alternative to the design of segmentation models including explicit regularizations, some recent advances have been made in the field of numerical schemes and more precisely, on the development of higher-order schemes for level set-based segmentation methods.

Higher-Order Schemes for Level Set-Based Segmentation Models

In Falcone et al. (2020), Falcone and collaborators focus on a level-based segmentation model including a modified velocity inspired by prior related works by Malladi et al. (1995) and more precisely, on a hybrid numerical scheme designed to avoid spurious oscillations around discontinuities of the solutions and/or jumps in the gradient. The propagating front is assumed to evolve in its normal direction with a velocity v depending only on time and space variables and supposed to be of constant sign in time. Its evolution is dictated by the following first-order Hamilton-Jacobi nonlinear equation of Eikonal type:

$$\begin{cases} v_t + c(t, x, y) |\nabla v| &= 0, & (t, x, y) \in (0, T) \times \mathbb{R}^2, \\ v(0, x, y) &= v_0(x, y), & (x, y) \in \mathbb{R}^2, \end{cases} \quad (3)$$

with v_0 a proper representation of the initial front Γ_0 , satisfying

$$\begin{cases} v_0(x, y) < 0, & (x, y) \in \Omega_0, \\ v_0(x, y) = 0, & (x, y) \in \Gamma_0, \\ v_0(x, y) > 0, & (x, y) \in \mathbb{R}^2 \setminus \overline{\Omega_0} \end{cases}$$

High-order schemes have been proposed to solve (3), most of them being based on nonoscillatory local interpolation technique, for which nevertheless general convergence theorems are lacking. These limitations motivate the introduction of a new class of high-order schemes for time-dependent Hamilton-Jacobi equations grounded on filtered schemes. The design of these filtered schemes relies on a simple coupling of a monotone scheme and a high-order scheme, which allows inheriting both the property of convergence to the weak viscosity solution of the monotone scheme—known however to be at most first order accurate—and the higher accuracy of high-order schemes that prove to be in general unstable by properly connecting these two schemes, guaranteeing then global convergence. This is the main focal point of Falcone et al. (2020) after proposing a way to compute a modified (in the sense, extended) velocity c in (3) ensuring regularity of the front evolution. As the front represents the boundary of an evolving shape, and since segmentation aims to extract object shapes from a given image, the front should stop moving in the vicinity of the desired object boundaries. The question of designing a suitable image-related speed function naturally emerges. From the modeling, this speed has only meaning on the zero level set function over the entire domain. In order for the evolution equation to have consistent meaning for all the level sets, an

extension of the image-related speed is introduced, this extension being governed by the following property extracted from Malladi et al. (1995):

- the image-related speed function must be devised so that level sets moving under this speed function cannot collide. A natural way of designing it is to let the speed at a point P lying on a level set $\{v = c\}$ be the value of the speed at a point Q such that Q is the closest point to P lying on the level set $\{v = 0\}$. Q is uniquely determined whenever the normal direction in P is well-defined.

The novelty of Falcone et al. (2020) relies then on the method of construction of this extended velocity: it is based on the central premise that if the layout of the level sets is known at initial stage and if all the points in the normal direction to the zero level set evolve according to the same law, it sounds reasonable to expect that all such points will keep their relative distance unchanged as time flows. This observation leads to the following definition of the extended velocity \tilde{c} :

$$\tilde{c}(x, y, v, v_x, v_y) = c(x - d(v) \frac{v_x}{|\nabla v|}, y - d(v) \frac{v_y}{|\nabla v|}),$$

d being a distance function. The adaptive filtered scheme is then introduced, composed of two intermediate schemes—the monotone one and the high-order one—related by a filter function F that switches between the two schemes according to smoothness indicators.

Joint Segmentation and Registration Models

Segmentation and registration are cornerstone steps in many image processing chains, which, combined, can significantly improve the accuracy of both processes.

Motivations

Like segmentation, registration can be achieved with a large variety of methodologies. In Sotiras et al. (2013), Sotiras et al. provide an extensive overview of existing registration techniques in a systematic manner, by identifying the main components they consider to be part of a registration algorithm, namely, the deformation model—or how the deformation is viewed—the cost function designed to enforce the shape matching, and the optimization technique adopted to complete the minimization. The deformation φ to be searched is viewed as a minimizer (uniqueness defaults in general) of a specifically designed cost function, the problem being mathematically hard to solve, due to its under-constrained, nonlinear, and non-convex nature and its strong dependency on the considered application. For instance, when the images are of different modalities, the quality of registration is no longer assessed by intensity distribution alignment but by the measurement of shape and geometric feature matching, requiring to design specific metrics. According to

Sotiras and collaborators, an image registration algorithm consists of three main components:

- (i) a deformation model describing the setting in which the objects to be matched are interpreted and viewed and allowing to favor certain properties of the deformation: physical models, purely geometric models, models including a priori knowledge, etc.
- (ii) a cost function which generally comprises two terms: a first one quantifying the misalignment between the deformed template and the reference and the second one regularizing the deformation, regularization prescribing the nature of the deformation
- (iii) an optimization method

The deformation model, which is thus the first ingredient, actually motivates the way the deformation φ is built in order to apply to a specific task:

- (i) by analogy with physical models: for instance, elastic models (Broit 1981) in which the shapes to be matched are considered as the observations of the same body before and after being subject to constraints, fluid models (Christensen et al. 1996) in which the shapes to be matched are viewed as fluids evolving in accordance with Navier-Stokes equations, diffusion models (Fischer and Modersitzki 2002), curvature models (Fischer and Modersitzki 2003), flows of diffeomorphisms (Beg et al. 2005), and nonlinear models (Burger et al. 2013, Derfoul and Le Guyader 2014, Droske and Rumpf 2004, Le Guyader and Vese 2011, Rumpf and Wirth 2009, Rabbitt et al. 1995, Pennec et al. 2005) to allow for large deformations.
- (ii) by interpolation or approximation-driven models: it means that the deformation is described in a parameterizable set. The displacements are considered to be known on a restricted set and are then extrapolated or approximated on the whole domain. The family of interpolation strategies includes radial basis functions (Zagorchev and Goshtasby 2006), elastic body splines (Davis et al. 1997), free-form deformations (Sederberg and Parry 1986), basis functions from signal processing (Ashburner and Friston 1999), and piecewise affine models. These models are rich enough to describe the transformations, while having low degrees of freedom.
- (iii) by including a priori knowledge (through conditioning statistically image matching or biomechanical/biophysical models, for instance, tumor growth model or biomechanical model of breast tissue (Clatz et al. 2005)) or shape a priori in order to penalize configurations that diverge too much from it.

Additional constraints can be applied in order for the deformation to exhibit suitable properties such as topology or orientation preservation (one-to-one property of the deformation) (Karaçali and Davatzikos 2004, Christensen et al. 1996, Musse et al. 2001, Noblet et al. 2005), symmetry, inverse consistency (which means that interchanging the template and the reference should not impact on the produced

result) (Yanovsky et al. 2007), volume preservation (Haber and Modersitzki 2004), lower and upper bounds on the Jacobian determinant (Haber and Modersitzki 2007), etc.

The second component of an image registration method is the objective function or the matching criterion, that is, how the available data are exploited to drive the registration process. Ideally, it should be devised in order to comply with the nature of the observations to be registered and should put the emphasis on salient features. There exist numerous types of matching criterion which can be regrouped into three categories as follows:

- (i) iconic methods: these concern intensity-based methods, attribute-based methods, and information-theoretic approaches.
- (ii) geometric methods: they aim to establish correspondences between landmarks (reliable anatomical locations, for instance).
- (iii) hybrid methods that summarize both types of approaches.

Finally, the last component is the optimization method, consisting of the following types:

- (i) continuous methods in which the variables are assumed to take real values and the objective function to be differentiable: gradient descent (Beg et al. 2005), conjugate gradient (Miller et al. 2002), Newton-type methods, Levenberg-Marquardt, and stochastic gradient descent methods (Wells et al. 1996)
- (ii) discrete methods (contrary to continuous methods, they perform a global search and exhibit better convergence rates than the continuous methods): graph-based (Tang et al. 2007), belief propagation, and linear programming methods
- (iii) miscellaneous methods: greedy approaches and evolutionary algorithms

For images including several objects, registration cannot just track the changes of a particular one. Yet, in some applications, we are only interested in tracing only one of these objects, resulting in a linear process of the two tasks: segmentation should be achieved first and then registration, meaning that segmentation and registration are processed sequentially, one task after another, without correlating them, which in practice may propagate errors from step to step. Still, as structure/salient component/shape/geometrical feature matching and intensity distribution comparison rule registration, combining the segmentation and registration tasks into a single framework sounds relevant. Beyond the fact it may reduce propagation of uncertainty, jointly performing these tasks yields positive mutual influence and benefit on the obtained results as exemplified in Fig. 7. Accurate segmented structures allow to drive the registration process correctly, providing then a reliable deformation between the encoded structures, not only based on intensity distribution comparison (local criterion) but also on geometrical and topological features (nonlocal feature) and edge transfer—thus diminishing the influence of noise. Besides, registration can be viewed as the incorporation of prior information to guide the segmentation process, in particular for the questions of topology preservation (the unknown

deformation is substituted for the classical evolving contour and the related Jacobian determinant is subject to positivity constraints) and geometric priors (since the registration allows to overcome the issue of weak boundaries). An overview of prior related works dedicated to joint segmentation and registration is produced in the next section.

Overview of Existing Methods

Several scientific works suggest to combine segmentation and registration to take advantage of both processes. Yezzi et al. (2001) propose performing jointly segmentation and registration. Denoting by $R : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ and $T = \hat{R} : \hat{\Omega} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ the two images containing a common object to be registered and segmented, their goal is to find a closed curve $C \subset \Omega$ which captures the boundary of an object in image R and another closed curve $\hat{C} \subset \hat{\Omega}$ which captures the boundary of the corresponding object in image \hat{R} , these closed curves being related through the mapping $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2, g \in G$ (finite dimensional group) by $\hat{C} = g(C)$. While the fidelity criterion is defined in terms of a region-based energy, the regularizer is based on the mean curvature flow in order to ensure that the evolving contour remains smooth. A generalization of this model is introduced in Unal and Slabaugh (2005).

In Vemuri et al. (2003), the authors propose a coupled PDE model to perform both segmentation and registration. In the first PDE, the level sets of the source image (i.e., the template) are evolved along their normals with a speed defined as the difference between the target (i.e., the reference) and the evolving source image. In Le Guyader and Vese (2011), Le Guyader and Vese introduce a segmentation model based on the active contour without edges, model that is now solved using registration techniques. The shapes to be matched are viewed as Ciarlet-Geymonat materials and are implicitly modeled by level set functions. Denoting by Ω a connected bounded open subset of \mathbb{R}^3 with Lipschitz boundary $\partial\Omega$, by R the reference image, and modeling the shape of the template image T via a Lipschitz continuous function Φ_0 whose zero level line corresponds to this shape boundary, the problem consists in searching for a deformation $\tilde{\varphi} \in \mathcal{W}$ with

$$\mathcal{W} = \left\{ \Psi \in W^{1,4}(\Omega, \mathbb{R}^3), \text{Cof } \nabla \Psi \in L^2(\Omega, M_3(\mathbb{R})), \right. \\ \left. \det \nabla \Psi \in L^2(\Omega), \Psi = \Psi_0 \text{ on } \partial\Omega \text{ and } \det \nabla \Psi > 0 \text{ a.e.} \right\}$$

realizing the minimum of

$$I(\varphi) = v_1 \int_{\Omega} (R - c_1(\varphi))^2 H(\Phi_0(\varphi)) dx + v_2 \int_{\Omega} (R - c_2(\varphi))^2 (1 - H(\Phi_0(\varphi))) dx \\ + \int_{\Omega} W_{CG}(\nabla \varphi) dx,$$

with

$$c_1(\varphi) = \frac{\int_{\Omega} R H(\Phi_0(\varphi)) dx}{\int_{\Omega} H(\Phi_0(\varphi)) dx}, \quad c_2(\varphi) = \frac{\int_{\Omega} R (1 - H(\Phi_0(\varphi))) dx}{\int_{\Omega} (1 - H(\Phi_0(\varphi))) dx},$$

and W_{CG} standing for the Ciarlet-Geymonat stored energy function. It is defined by $W_{CG} : F \in M_3^+(\mathbb{R}) = \{A \in M_3(\mathbb{R}) \mid \det A > 0\} \mapsto W_{CG}(F) = a_1 \|F\|^2 + a_2 \|F^T F\|^2 + b \|\text{Cof } F\|^2 + \Gamma(\det F) + e$, $M_3(\mathbb{R})$ being the set of real square matrices of order 3, $\text{Cof } F$ denoting the cofactor matrix of F , $\|\cdot\|$ the Frobenius norm, and Γ being a convex function which penalizes expansions and contractions of the deformation that are too large. Following this work, the model by Ibrahim et al. (2016) proposed a simple extension by adopting a high-order regularizer for the deformation field that offers advantages for certain classes of problems.

In Lord et al. (2007), the authors present a unified model that simultaneously treats segmentation and registration based on metric structure comparisons (see also the more recent work Gooya et al. (2012) based on expectation-maximization algorithm that incorporates a glioma growth model for atlas seeding). In An et al. (2005), the authors propose a new variational PDE-based level set method for a simultaneous image segmentation and non-rigid registration (the expected transformation is not parametric, i.e., it is not expanded in some basis functions) using prior shape and intensity information. While the segmentation is obtained by determining a non-rigid deformation of the prior shape, the non-rigid registration consists of both a global rigid transformation (transformations are restricted to rotations and translations) and a local non-rigid deformation. A joint segmentation and registration algorithm for infant brain images with the goal to accurately characterize structure changes is presented in Wu et al. (2014). The emphasis is put on the interest of combining both tasks owing to dynamic appearance change with rapid brain development. In Gorthi et al. (2011), relevant accurate segmented structures allow to drive correctly the registration process. The proposed model integrates both the active contour framework and the dense deformation fields of optical flow framework. In Droske and Rumpf (2007), the authors aim to match the edges and the normals of the two images by applying a Mumford-Shah-type free discontinuity problem. More recently, Ozeré et al. (2015) have introduced a variational joint segmentation/registration model combining a measure of dissimilarity based on weighted total variation and a regularizer based on the stored energy function of a Saint Venant-Kirchhoff material. In the same spirit, Debroux et al. (2017) provide a nonlocal topology-preserving segmentation guided registration model, theoretically well-motivated and capable of handling large and smooth deformations. The shapes to be matched are viewed as hyperelastic materials and more precisely as Saint Venant-Kirchhoff ones and are implicitly modeled by level set functions. These are driven to minimize a functional containing both a nonlinear elasticity-based regularizer prescribing the nature of the deformation and a criterion that forces the evolving shape to match intermediate topology-preserving segmentation results. In

Boink (2016), a joint segmentation/optimal transport model is analyzed to determine the velocity of blood flow in vascular structures. A convex variational method is used, and primal-dual proximal splitting algorithms are implemented. At last, in Wirth (2016), the author wonders about the behavior of phase field approximations of the Mumford-Shah model when used for joint segmentation and registration.

We conclude this section with a special focus on a recent combined segmentation/registration framework.

A Mixed Segmentation/Registration Model Based on a Nonlocal Characterization of Weighted Total Variation

In 2018, Debroux and Le Guyader (2018) propose a unified variational model in a hyperelasticity setting. The dissimilarity measure relates local and global (region-based) information, since relying on the weighted total variation and nonlocal shape descriptors inspired by the piecewise constant Mumford-Shah model. Including the weighted total variation enables one to consider a larger class of images (not necessarily of the same modality) and to compare shapes (alignment of the level curves) rather than intensities. Also, in practice, the obtained deformed templates are more consistent with the complex topologies/thin structures involved. In addition to theoretical results (existence of minimizers, connection to the segmentation step, etc.), a nonlocal characterization of weighted semi-norms is provided as well as asymptotic results and Γ -convergence properties. We now show some details of this model. Assume $\Omega \subset \mathbb{R}^2$ is of class C^1 . Denote by $R : \bar{\Omega} \rightarrow \mathbb{R}$ the reference image assumed to be sufficiently smooth and by $T : \bar{\Omega} \rightarrow \mathbb{R}$ the template image. We assume that T is compactly supported on Ω to ensure that $T \circ \varphi$ is always defined and we assume that T is Lipschitz continuous. It can thus be considered as an element of the Sobolev space $W^{1,\infty}(\mathbb{R}^2)$. Let $\varphi : \bar{\Omega} \rightarrow \mathbb{R}^2$ be the sought deformation supposed to be a smooth orientation-preserving mapping. The deformation gradient is $\nabla\varphi : \bar{\Omega} \rightarrow M_2(\mathbb{R})$, the set $M_2(\mathbb{R})$ being the set of real square matrices of order 2. The deformation to be searched φ is seen as the minimal argument of a specifically designed objective function including a regularization on φ prescribing the nature of the deformation and which is modeled by the component $\int_{\Omega} QW(\nabla\varphi) dx$ in the functional to be minimized (its design is more precisely motivated hereafter) and a term measuring alignment or how the available data are exploited to drive the registration process. This later one is itself decomposed into three components: the first one ensuring alignment of the edges of both the deformed template $T \circ \varphi$ and the reference and expressed in terms of the weighted total variation $\text{var}_g T \circ \varphi$, the second one which is merely the L^2 -fidelity term $\|R - T \circ \varphi\|_{L^2(\Omega)}^2$ insuring intensity pairing, and the last one, $\int_{\Omega} [(R - c_1)^2 - (R - c_2)^2] T \circ \varphi dx$, inspired by the work of Bresson et al. (2007) which guarantees region matching. Again, the design of these terms is justified below. To allow large deformations, the shapes to be matched are viewed as hyperelastic materials and more precisely as

Saint Venant-Kirchhoff ones (Ciarlet 1985). This outlook rules the design of the regularization on φ which is thus based on the stored energy function of a Saint Venant-Kirchhoff material. We recall that the right Cauchy-Green strain tensor (viewed as a quantifier of the square of local change in distances due to deformation) is defined by $C = \nabla\varphi^T \nabla\varphi = F^T F$. The Green-Saint Venant strain tensor is defined by $E = \frac{1}{2} (C - I)$. Associated with a given deformation φ , it is a measure of the deviation between φ and a rigid deformation. We also need the following notations: $A : B = \text{tr} A^T B$, the matrix inner product and $\|A\| = \sqrt{A : A}$, the related matrix norm (Frobenius norm). The stored energy function of a Saint Venant-Kirchhoff material is defined by $W_{SVK}(F) = \widehat{W}(E) = \frac{\lambda}{2} (\text{tr } E)^2 + \mu \text{tr } E^2$, λ and μ being the Lamé coefficients. To ensure that the distribution of the deformation Jacobian determinants does not exhibit contractions or expansions that are too large and to avoid singularity as much as possible, we complement the stored energy function W_{SVK} by the term $\mu (\det F - 1)^2$ controlling that the Jacobian determinant remains close to 1. The weighting of the determinant component by parameter μ allows to recover a property of convexity for the function Ψ introduced later. (Note that the stored energy function W_{SVK} alone lacks a term penalizing the determinant: it does not preclude deformations with negative Jacobian. The expression of its quasiconvex envelope is more complex since involving explicitly the singular values of F . Also, when they are all lower than 1, the quasiconvex envelope equals 0, which shows bad behavior under compression). Therefore, the regularization can be written, after intermediate computations, as $W(F) = \beta (\|F\|^2 - \alpha)^2 - \frac{\mu}{2} (\det F)^2 + \mu (\det F - 1)^2 + \frac{\mu(\lambda + \mu)}{2(\lambda + 2\mu)}$, where $\alpha = 2 \frac{\lambda + \mu}{\lambda + 2\mu}$ and $\beta = \frac{\lambda + 2\mu}{8}$. Although meaningful, function W takes on a drawback since it is not quasiconvex (see Dacorogna 2008, Chapter 9 for a complete review of this notion), which raises an issue of a theoretical nature since we cannot obtain the weak lower semicontinuity property. The idea is thus to replace W by its quasiconvex envelope defined by

$$QW(\xi) = \begin{cases} W(\xi) & \text{if } \|\xi\|^2 \geq 2 \frac{\lambda + \mu}{\lambda + 2\mu}, \\ \Psi(\det \xi) & \text{if } \|\xi\|^2 < 2 \frac{\lambda + \mu}{\lambda + 2\mu}, \end{cases} \quad \text{and } \Psi, \text{ the convex mapping such that}$$

$$\Psi : t \mapsto -\frac{\mu}{2} t^2 + \mu (t - 1)^2 + \frac{\mu(\lambda + \mu)}{2(\lambda + 2\mu)} \quad (\text{see Ozeré et al. 2015 for the derivation}),$$

for which the minimal argument is $t = 2$. The regularizer is now complemented by a dissimilarity measure inspired by the unified model of image segmentation and denoising introduced by Bresson et al. (2007), designed to overcome the limitation of local minima and to deal with global minimum.

In that purpose, let $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be an edge detector function satisfying $g(0) = 1$, g strictly decreasing and $\lim_{r \rightarrow +\infty} g(r) = 0$. From now on, we set $g := g(|\nabla R|)$, and for theoretical purposes, we assume that $\exists c > 0$ such that $0 < c \leq g \leq 1$ and that g is Lipschitz continuous. We then use the generalization of the notion of function of bounded variation to the setting of BV -spaces associated with a Muckenhoupt's weight function depicted in Baldi (2001). We follow Baldi's arguments and notations to define the weighted BV -space related to weight g .

For a general weight w , some hypotheses are required (fulfilled here by g). More precisely, Ω_0 being a neighborhood of $\bar{\Omega}$, the positive weight $w \in L^1_{loc}(\Omega_0)$ is assumed to belong to the global Muckenhoupt's $A_1 = A_1(\Omega)$ class of weight functions, i.e., w satisfies the condition:

$$C w(x) \geq \frac{1}{|B(x, r)|} \int_{B(x, r)} w(y) dy \text{ a.e.} \tag{4}$$

in any ball $B(x, r) \subset \Omega_0$. Now, denoting by A_1^* the class of weights $w \in A_1$, w lower semicontinuous (lsc) and that satisfy condition (4) pointwise, the definition of the weighted BV -space related to weight w is given by:

Definition 1 (Baldi 2001, Definition 2). Let w be a weight function in the class A_1^* . We denote by $BV(\Omega, w)$ the set of functions $u \in L^1(\Omega, w)$ (set of functions that are integrable with respect to the measure $w(x) dx$) such that:

$$\sup \left\{ \int_{\Omega} u \operatorname{div}(\varphi) dx : |\varphi| \leq w \text{ everywhere, } \varphi \in \operatorname{Lip}_0(\Omega, \mathbb{R}^2) \right\} < \infty, \tag{5}$$

with $\operatorname{Lip}_0(\Omega, \mathbb{R}^2)$ the space of Lipschitz continuous functions with compact support. We denote by $\operatorname{var}_w u$ the quantity (5).

Remark 1. In Baldi (2001), Baldi defines the BV -space taking as test functions elements of $\operatorname{Lip}_0(\Omega, \mathbb{R}^2)$. Classically in the literature, the test functions are chosen in $C^1_c(\Omega, \mathbb{R}^2)$. It can be proved that these two definitions coincide thanks to mollifications and density results.

To explain (5), we give the following result (Baldi 2001, Remark 10):

Remark 2. Given a weight w sufficiently smooth, if E is a regular bounded open set in \mathbb{R}^2 , with boundary of class C^2 , then $|\partial E|(\Omega, w) = \operatorname{var}_w \chi_E = \int_{\Omega \cap \partial E} w dH^1$, which can be interpreted in the case where $w = g$ as a new definition of the curve length with a metric that depends on the reference image content.

Equipped with this material (and due to the properties of function g : it is obviously L^1 , continuous and it suffices to take $C = \frac{1}{c}$ to satisfy (4) pointwise), we propose introducing as dissimilarity measure the following functional:

$$\begin{aligned} W_{fid}(\varphi) = & \operatorname{var}_g T \circ \varphi + \frac{\nu}{2} \int_{\Omega} (T \circ \varphi(x) - R(x))^2 dx \\ & + a \int_{\Omega} \left[(c_1 - R(x))^2 - (c_2 - R(x))^2 \right] T \circ \varphi(x) dx, \end{aligned} \tag{6}$$

with $c_1 = \frac{\int_{\Omega} R(x) H_{\varepsilon}(T \circ \varphi(x) - \rho) dx}{\int_{\Omega} H_{\varepsilon}(T \circ \varphi(x) - \rho) dx}$ and $c_2 = \frac{\int_{\Omega} R(x) (1 - H_{\varepsilon}(T \circ \varphi(x) - \rho)) dx}{\int_{\Omega} (1 - H_{\varepsilon}(T \circ \varphi(x) - \rho)) dx}$ —we dropped the dependency on φ to lighten the expressions— H_{ε} denoting a regularization of the Heaviside function and $\rho \in [0, 1]$ being a fixed parameter allowing to partition $T \circ \varphi$ into two phases and yielding a binary version of the reference. ρ can be estimated by analyzing the reference histogram to discriminate two relevant regions or phases, for instance, through histogram shape-based methods, clustering-based methods, entropy-based methods, object attribute-based methods, spatial methods, or local methods (Sezgin and Sankur 2004). This proposed functional emphasizes the link between the geodesic active contour model (Caselles et al. 1997) and the piecewise constant Mumford-Shah model (Mumford and Shah 1989): if \tilde{T} is the characteristic function of the set Ω_C , bounded subset of Ω with regular boundary C , $\text{var}_g \tilde{T}$ is a new definition of the length of C with a metric depending on the reference content (so minimizing this quantity is equivalent to locating the curve on the boundary of the shape contained in the reference), while $\int_{\Omega} [(c_1 - R(x))^2 - (c_2 - R(x))^2] \tilde{T}(x) dx$ approximates R in the L^2 sense by two regions Ω_C and $\Omega \setminus \Omega_C$ with two values c_1 and c_2 . Indeed, $\text{var}_g \tilde{T} = \int_{\Omega \cap C} g dH^1$, and if c_1 and c_2 are fixed (which is in practice the case in the alternating algorithm), $\int_{\Omega} [(c_1 - R(x))^2 - (c_2 - R(x))^2] 1_{\Omega_C} dx$ is equivalent to minimizing $\int_{\Omega} (c_1 - R(x))^2 1_{\Omega_C} dx + \int_{\Omega} (c_2 - R(x))^2 1_{\Omega \setminus \Omega_C} dx$.

In the end, the global minimization problem denoted by (QP)—which stands for *quasiconvex problem*— is stated by:

$$\inf_{\varphi \in \mathcal{W} = \text{Id} + W_0^{1,4}(\Omega, \mathbb{R}^2)} \bar{I}(\varphi) = W_{fid}(\varphi) + \int_{\Omega} QW(\nabla \varphi) dx \tag{QP}$$

which is a relaxed problem from the following formulation

$$\inf_{\varphi \in \mathcal{W} = \text{Id} + W_0^{1,4}(\Omega, \mathbb{R}^2)} \bar{I}(\varphi) = W_{fid}(\varphi) + \int_{\Omega} W(\nabla \varphi) dx. \tag{P}$$

Here $\varphi \in \text{Id} + W_0^{1,4}(\Omega, \mathbb{R}^2)$ means that $\varphi = \text{Id}$ on $\partial\Omega$ and $\varphi \in W^{1,4}(\Omega, \mathbb{R}^2)$. $W^{1,4}(\Omega, \mathbb{R}^2)$ denotes the Sobolev space of functions $\varphi \in L^4(\Omega, \mathbb{R}^2)$ with distributional derivatives up to order 1 which also belong to $L^4(\Omega)$. \mathcal{W} is a suitable space due, in particular, to the $\|F\|^4$ component in $W(F)$. Note that from generalized Hölder’s inequality, if $\varphi \in W^{1,4}(\Omega, \mathbb{R}^2)$, then $\det \nabla \varphi \in L^2(\Omega)$. Now we justify that $\text{var}_g T \circ \varphi$ is well-defined. In Ambrosio and Dal Maso (1990), Ambrosio and Dal Maso prove a general chain rule for the distribution derivatives of the composite function $v(x) = f(u(x))$, where $u : \mathbb{R}^n \rightarrow \mathbb{R}^m$ has bounded variation and $f : \mathbb{R}^m \rightarrow \mathbb{R}^k$ is Lipschitz continuous. A simpler result is given when $u \in W^{1,p}(\Omega, \mathbb{R}^m)$ for some p , $1 \leq p \leq +\infty$, resulting in our case in $T \circ \varphi \in W^{1,4}(\Omega) := W^{1,4}(\Omega, \mathbb{R}) \subset BV(\Omega) \subset BV(\Omega, g)$, since $g \leq 1$. Some theoretical results can be found in Debroux and Le Guyader (2018) among which, existence of minimizers for problem (QP), connection between the minimum of

(QP) and the infimum of (P), and derivation of a related nonlocal problem, motivated by the strength and robustness of nonlocal methods exemplified in many image processing tasks such as image denoising, color image deblurring in the presence of Gaussian or impulse noise, color image inpainting, color image super-resolution, or color filter array demosaicing (Jung et al. 2011). The next part is dedicated to the derivation of a nonlocal counterpart of problem (QP). In practice, in terms of quantitative and qualitative accuracy, this nonlocal model gives better results than those obtained with the local one, with higher Dice coefficients, in particular when the shapes to be matched exhibit fine details or complex topologies.

The statement of the nonlocal problem relies on the following nonlocal approximation of the weighted total variation (or nonlocal weighted BV semi-norm) by a sequence of integral operators involving a differential quotient and a radial mollifier sequence. It is inspired by prior works by Dávila and Ponce dedicated to the design of nonlocal counterparts of Sobolev and BV semi-norms. Let $(\rho_n)_{n \in \mathbb{N}}$ be a sequence of radial mollifiers satisfying: $\forall n \in \mathbb{N}, \forall x \in \mathbb{R}^2, \rho_n(x) = \rho_n(|x|); \forall n \in \mathbb{N}, \rho_n \geq 0; \forall n \in \mathbb{N}, \int_{\mathbb{R}^2} \rho_n(x) dx = 1; \forall \delta > 0, \lim_{n \rightarrow +\infty} \int_{\delta}^{+\infty} \rho_n(r) r dr = 0$. Then the following theorem holds:

Theorem 2. *Let $\Omega \subset \mathbb{R}^2$ be an open bounded set with Lipschitz boundary and let $f \in BV(\Omega, g) \subset BV(\Omega)$ as $0 < c \leq g \leq 1$ everywhere. Consider $(\rho_n)_{n \in \mathbb{N}}$ defined previously. Then*

$$\begin{aligned} & \lim_{n \rightarrow +\infty} \int_{\Omega} g(x) \left[\int_{\Omega} \frac{|f(x) - f(y)|}{|x - y|} \rho_n(x - y) dy \right] dx \\ &= \left[\frac{1}{|S^1|} \int_0^{2\pi} \left| e \cdot \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \right| d\theta \right] \text{var}_g f = K_{1,2} \text{var}_g f, \end{aligned}$$

with e being any unit vector of \mathbb{R}^2 and S^1 being the unit sphere in \mathbb{R}^2 .

Motivated by the asymptotic properties of the above nonlocal quantity, we propose using this characterization as a substitute for the weighted total variation of $T \circ \varphi$, yielding the following nonlocal problem (NLP):

$$\begin{aligned} & \inf_{\varphi \in \text{Id} + W_0^{1,4}(\Omega, \mathbb{R}^2)} \left\{ E_n(\varphi) = \frac{1}{K_{1,2}} \int_{\Omega} g(x) \left[\int_{\Omega} \frac{|T \circ \varphi(y) - T \circ \varphi(x)|}{|x - y|} \rho_n(x - y) dy \right] dx \right. \\ & \quad + a \int_{\Omega} \left[(c_1 - R)^2 - (c_2 - R)^2 \right] T \circ \varphi dx \\ & \quad \left. + \frac{\nu}{2} \|T \circ \varphi - R\|_{L^2(\Omega)}^2 + \int_{\Omega} QW(\nabla \varphi) dx \right\}. \tag{NLP} \end{aligned}$$

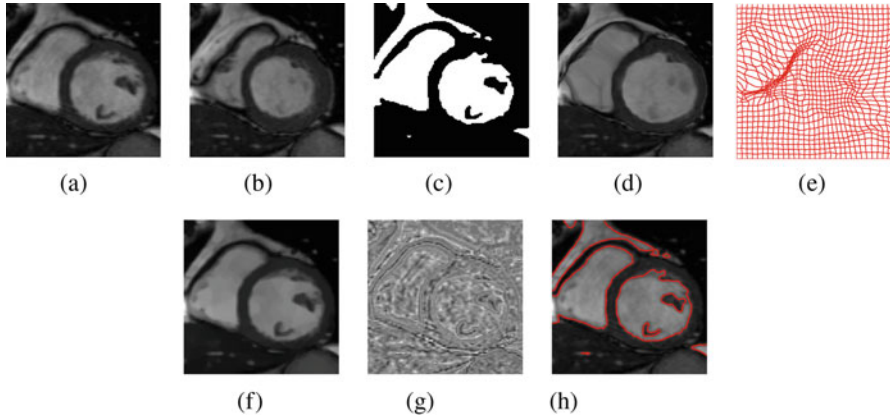


Fig. 7 Mapping of cardiac MRI images (ED-ES) (size: 150×150). (a) R (b) T (c) Binary Reference (rescaled) (d) $T \circ \varphi$ (e) Deformation grid (f) \tilde{T} (g) $R - \tilde{T}$ (h) Segmented Reference

In addition to an existence result for problem (NLP), the authors provide a Γ -convergence theorem relating the approximated problem to the original one when n tends to $+\infty$ as well as a numerical approximation based on the introduction of two auxiliary (i.e., splitting) variables, \tilde{T} , simulating $T \circ \varphi$ and V , approximating $\nabla\varphi$, the underlying idea being to transfer the nonlinearity on V . An asymptotic result is stated (Debroux and Le Guyader 2018, Theorem 3.4), together with a result (Debroux and Le Guyader 2018, Theorem 3.6) relating segmentation and registration. The method has then been applied on MRI images of a patient cardiac cycle (courtesy of Caroline Petitjean, LITIS, University of Rouen, France): the reference corresponds to end diastole (ED), i.e., when the heart is the most dilated, while the template corresponds to end systole (ES), i.e., when the heart is the most contracted. In Fig. 7, we provide the reference R , the template T , the binary reference obtained thanks to c_1 and c_2 which is rescaled to 0–1 from the nonlocal numerical method, the deformed template, the deformation grid which does not exhibit any overlap (thus yielding the physical well-definedness of the deformation), \tilde{T} the simplified version of the deformed template, the segmentation of the reference obtained thanks to \tilde{T} , and the oscillatory part resulting from $R - \tilde{T}$.

Other Related Models

The topics related to segmentation and registration are huge. The area is active and fast growing. We only briefly mention a few directions.

Optimal Flow Frameworks

When a sequence of images z_1, z_2, \dots are given, e.g., from functional MRI or from frames of a video image, segmented objects in z_1 are hoped to be registered to

the evolved features in subsequent images. This may be realized by optimal flow registration methods or by joint segmentation and registration models (Debroux and Le Guyader 2018; Brox and Malik 2010; Cohen 1993).

Shape Priors

Given a shape prior ψ_0 intended for an image z , there exist several models in the literature trying to segment an object ψ in z that is some transformed version of ψ_0 . This seemingly simple and useful task is highly non-trivial to realize, unless ψ is a parametric (e.g., affine) transform of ψ_0 ; see Cremers et al. (2002) and Gu (2017) and many references therein. One fundamental challenge is that registration models such as (QP) are highly capable to transform one shape to another (Debroux et al. 2017) and hence if not constrained, registration would attempt to find a match of objects by essentially ignoring the given shape prior.

Deep Learning Models

Deep learning models have been extremely popular for solving models in segmentation, registration, or joint segmentation and registration (Estienne et al. 2019; Xu and Niethammer 2019). In fact, supervised learning for image segmentation and unsupervised learning for image registration are widely used for various image applications. Current and emerging works show novelties in terms of new network designs for segmentation while of new energy (loss) functions.

Multi-modal Problems

Segmentation of multi-modal images as a sole task seems to pose no particular or additional challenges, although segmentation of an arbitrary image is an unfinished business due to various inherent difficulties such as low contrast, strong noise, possibly non-periodic textures, and missing data. Among many competing models, unsurprisingly, deep learning methods (Taghanaki et al. 2019) are increasingly used. Moreover, one can make use of deep learning ideas for multi-modal images to obtain a more accurate segmentation through fusion (Zhou et al. 2019).

The related task of image registration for multi-modal images is particularly challenging if one wishes to have robustness. A particularly reliable method suitable for registering multi-modal images is the use of quasi-conformal maps (Lee et al. 2016) if landmark points could be identified. If it is not possible to find reliable landmark points, the design of suitable dissimilarity measures to discriminate objects across modalities is a key. Refer to Chen et al. (2019) and Theljani and Chen (2019).

Once a dissimilar measure is defined, development of a registration model and also a joint segmentation and registration model can follow from works for single modality images (Debroux et al. 2017).

Conclusion

By covering a broad spectrum of constraint types, whether it be geometrical constraints to identify a single object among several ones or topological conditions to ensure that the segmented shape is homeomorphic to the initial one, we have intended in this survey to show the utility of including such additional a priori information to achieve more accurate results, in compliance with the physics of the problem or the anatomical reality. Still, however abundant the literature is on this topic, scientific obstacles remain in particular in the way to reconcile the intrinsic global nature of topology with its more local one, which gives tremendous prospects for the future.

References

- Alberti, G., Bouchitté, G., Dal Maso, G.: The calibration method for the Mumford-Shah functional and free-discontinuity problems. *Calc. Var. Partial Differ. Equ.* **16**(3), 299–333 (2003)
- Alexandrov, O., Santosa, F.: A topology-preserving level set method for shape optimization. *J. Comput. Phys.* **204**(1), 121–130 (2005)
- Alvarez, L., Cuenca, C., Díaz, J.I., González, E.: Level set regularization using geometric flows. *SIAM J. Imag. Sci.* **11**(2), 1493–1523 (2018)
- Ambrosio, L., Dal Maso, G.: A general chain rule for distributional derivatives. *Proc. Am. Math. Soc.* **108**(3), 691–702 (1990)
- An, J.H., Chen, Y., Huang, F., Wilson, D., Geiser, E.: A variational PDE based level set method for a simultaneous segmentation and non-rigid registration. In: Duncan, J.S., Gerig, G. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005: 8th International Conference, Palm Springs, 26–29 Oct 2005, Proceedings, Part I*, pp. 286–293. Springer, Berlin/Heidelberg (2005)
- Ashburner, J., Friston, K.J.: Nonlinear spatial normalization using basis functions. *Hum. Brain Mapp.* **7**(4), 254–266 (1999)
- Astala, K., Iwaniec, T., Martin, G.: *Elliptic Partial Differential Equations and Quasiconformal Mappings in the Plane*. Princeton University Press (2009)
- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Applied Mathematical Sciences. Springer (2001)
- Ayed, I.B., Li, S., Islam, A., Garvin, G., Chhem, R.: Area prior constrained level set evolution for medical image segmentation. In: Reinhardt, J.M., Pluim, J.P.W. (eds.) *Medical Imaging 2008: Image Processing*, vol. 6914, pp. 27–32. SPIE (2008)
- Badshah, N., Chen, K.: Image selective segmentation under geometrical constraints using an active contour approach. *Commun. Comput. Phys.* **7**, 759–778 (2010)
- Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007)
- Baldi, A.: Weighted BV functions. *Houston J. Math.* **27**(3), 683–705 (2001)
- Barrett, W., Mortensen, E.N.: Interactive live-wire boundary extraction. *Med. Image Anal.* **1**(4), 331–341 (1997)

- Beg, M., Miller, M., Trouvé, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **61**(2), 139–157 (2005)
- Ben-Zadok, N., Riklin-Raviv, T., Kiryati, N.: Interactive level set segmentation for image-guided therapy. In: *IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1079–1082 (2009)
- Bertrand, G.: Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recogn. Lett.* **15**(10), 1003–1011 (1994)
- Bertrand, G.: A Boolean characterization of three-dimensional simple points. *Pattern Recogn. Lett.* **17**(2), 115–124 (1996)
- Blake, A., Zisserman, A.: *Visual Reconstruction*. MIT Press, Cambridge (1989)
- Bogovic, J.A., Prince, J.L., Bazin, P.L.: A multiple object geometric deformable model for image segmentation. *Comput. Vis. Image Underst.* **117**(2), 145–157 (2013)
- Boink, Y.: Combined modelling of optimal transport and segmentation revealing vascular properties (2016)
- Boutry, N., Géraud, T., Najman, L.: A tutorial on Well-Composedness. *J. Math. Imaging Vision* **60**(3), 443–478 (2018)
- Boykov, Y.Y., Jolly, M.: Interactive graph cuts for optimal boundary map; region segmentation of objects in N-D images. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1, pp. 105–112 1 (2001)
- Bresson, X., Esedoğlu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.* **28**(2), 151–167 (2007)
- Broit, C.: *Optimal registration of Deformed Images*. Ph.D. thesis, Computer and Information Science, University of Pennsylvania (1981)
- Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(3), 500–513 (2010)
- Burger, M., Modersitzki, J., Ruthotto, L.: A hyperelastic regularization energy for image registration. *SIAM J. Sci. Comput.* **35**(1), B132–B148 (2013)
- Cai, X., Chan, R., Zeng, T.: A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding. *SIAM J. Imag. Sci.* **6**(1), 368–390 (2013)
- Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vis.* **22**(1), 61–79 (1997)
- Cecil, T.: *Numerical methods for partial differential equations involving discontinuities*. Ph.D. thesis, Department of Mathematics, University of California, Los Angeles (2003)
- Chambolle, A., Cremers, D., Pock, T.: A convex approach to minimal partitions. *SIAM J. Imag. Sci.* **5**(4), 1113–1158 (2012)
- Chan, H.L., Yan, S., Lui, L.M., Tai, X.C.: Topology-preserving image segmentation by Beltrami representation of shapes. *J. Math. Imaging Vis.* **60**(3), 401–421 (2018)
- Chan, T.F., Esedoğlu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *J. SIAM Appl. Math.* **66**(5), 1632–1648 (2006)
- Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chen, C., Freedman, D.: Topology noise removal for curve and surface evolution. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pp. 31–42. Springer, Berlin/Heidelberg (2011)
- Chen, C., Freedman, D., Lampert, C.H.: Enforcing topological constraints in random field image segmentation. In: *CVPR 2011*, pp. 2089–2096 (2011)
- Chen, D., Zhang, J., Cohen, L.D.: Minimal paths for tubular structure segmentation with coherence penalty and adaptive anisotropy. *IEEE Trans. Image Process.* **28**(3), 1271–1284 (2019)
- Chen, K., Lui, L.M., Modersitzki, J.: Image and surface registration. In: *Elsevier Handbook of Numerical Analysis. Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*, chap. 15, pp. 579–611. North Holland (2019)
- Christensen, G., Rabbitt, R., Miller, M.: Deformable templates using large deformation Kinematics. *IEEE Trans. Image Process.* **5**(10), 1435–1447 (1996)

- Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J.: Fuzzy C-means clustering with spatial information for image segmentation. *Comput. Med. Imaging Graph.* **30**(1), 9–15 (2006)
- Ciarlet, P.: *Elasticité Tridimensionnelle*. Masson (1985)
- Clatz, O., Sermesant, M., Bondiau, P.Y., Delingette, H., Warfield, S.K., Malandain, G., Ayache, N.: Realistic simulation of the 3-D growth of brain tumors in MR images coupling diffusion with biomechanical deformation. *IEEE Trans. Med. Imaging* **24**(10), 1334–1346 (2005)
- Cohen, I.: Nonlinear variational method for optical flow computation. In: *Proceedings of the 8th Scandinavian Conference on Image Analysis (SCIA)*, pp. 523–530. Springer (1993)
- Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002)
- Crandall, M., Ishii, H., P.-L.L.: User’s guide to viscosity solutions of second order partial differential equations. *Bull. Am. Math. Soc.* **27**(1), 1–67 (1992)
- Cremers, D., Fluck, O., Rousson, M., Aharon, S.: A probabilistic level set formulation for interactive organ segmentation. In: *Medical Imaging 2007: Image Processing*, vol. 6512, pp. 304–312. SPIE (2007)
- Cremers, D., Tischhäuser, F., Weickert, J., Schnörr, C.: Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *Int. J. Comput. Vis.* **50**(3), 295–313 (2002)
- Criminisi, A., Sharp, T., Blake, A.: GeoS: Geodesic image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *Computer Vision – ECCV 2008*, pp. 99–112. Springer, Berlin/Heidelberg (2008)
- Dacorogna, B.: *Direct methods in the calculus of variations*, 2nd edn. Springer (2008)
- Davis, M.H., Khotanzad, A., Flamig, D.P., Harms, S.E.: A physics-based coordinate transformation for 3-D image matching. *IEEE Trans. Med. Imaging* **16**(3), 317–328 (1997)
- Debroux, N., Le Guyader, C.: A joint segmentation/registration model based on a nonlocal characterization of weighted total variation and nonlocal shape descriptors. *SIAM J. Imag. Sci.* **11**(2), 957–990 (2018)
- Debroux, N., Ozeré, S., Le Guyader, C.: A non-local topology-preserving segmentation guided registration model. *J. Math. Imag. Vision* **59**, 1–24 (2017)
- Derfoul, R., Le Guyader, C.: A relaxed problem of registration based on the Saint Venant-Kirchhoff material stored energy for the mapping of mouse brain gene expression data to a neuroanatomical mouse atlas. *SIAM J. Imag. Sci.* **7**(4), 2175–2195 (2014)
- Droske, M., Rumpf, M.: A variational approach to non-rigid morphological registration. *SIAM J. Appl. Math.* **64**(2), 668–687 (2004)
- Droske, M., Rumpf, M.: Multiscale joint segmentation and registration of image morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(12), 2181–2194 (2007)
- Estienne, T., Vakalopoulou, M., Christodoulidis, S., Battistella, E., Lerousseau, M., Carre, A., Klausner, G., Sun, R., Robert, C., Mougiakakou, S., Paragios, N., Deutsch, E.: U-ReSNet: Ultimate coupling of registration and segmentation with deep nets. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.T., Khan, A. (eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pp. 310–319. Springer International Publishing (2019)
- Falcone, M., Paolucci, G., Tozza, S.: A high-order scheme for image segmentation via a modified level-set method. *SIAM J. Imag. Sci.* **13**(1):497–534 (2020)
- Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**(2), 167–181 (2004)
- Fischer, B., Modersitzki, J.: Fast diffusion registration. *AMS Contemp. Math. Inverse Prob. Image Anal. Med. Imag.* **313**, 11–129 (2002)
- Fischer, B., Modersitzki, J.: Curvature based image registration. *J. Math. Imaging Vis.* **18**(1), 81–85 (2003)
- Fischl, B., Liu, A., Dale, A.M.: Automated manifold surgery: constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Trans. Med. Imaging* **20**(1), 70–80 (2001)

- Fuzhen, H., Xuhong, Y.: Image segmentation under occlusion using selective shape priors. In: Campilho, A., Kamel, M. (eds.) *Image Analysis and Recognition*, pp. 89–95. Springer, Berlin/Heidelberg (2010)
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. *Appl. Soft Comput.* **70**, 41–65 (2018)
- Geiping, J.A.: Comparison of topology-preserving segmentation methods and application to mitotic cell tracking. Westfälische Wilhelms-Universität Münster (2014)
- Giga, Y., Goto, S., Ishii, H., Sato, M.H.: Comparison principle and convexity preserving properties for singular degenerate parabolic equations on unbounded domains. *Indiana Univ. Math. J.* **40**(2), 443–470 (1991)
- Gooya, A., Pohl, K., Bilello, M., Cirillo, L., Biros, G., Melhem, E., Davatzikos, C.: GLISTR: Glioma image segmentation and registration. *IEEE Trans. Med. Imaging* **31**(10), 1941–1954 (2012)
- Gorghi, S., Duay, V., Bresson, X., Cuadra, M.B., Castro, F.J.S., Pollo, C., Allal, A.S., Thiran, J.P.: Active deformation fields: Dense deformation field estimation for atlas-based segmentation using the active contour framework. *Med. Image Anal.* **15**(6), 787–800 (2011)
- Gout, C., Le Guyader, C., Vese, L.A.: Segmentation under geometrical conditions with geodesic active contour and interpolation using level set methods. *Numer. Algorithms* **39**(1), 155–173 (2005)
- Grady, L.: Random Walks for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1768–1783 (2006)
- Gu, Y., Rice, M., Xiong, W., Li, L.: A new approach for image segmentation with shape priors based on the Potts model. In: *Proceedings of APSIPA Annual Summit and Conference 2017*, pp. 12–15. IEEE (2017)
- Haber, E., Modersitzki, J.: Numerical methods for volume preserving image registration. *Inverse Probl.* **20**(5), 1621–1638 (2004)
- Haber, E., Modersitzki, J.: Image registration method with guaranteed displacement regularity. *Int. J. Comput. Vision* **71**(3), 361–372 (2007)
- Han, X., Xu, C., Braga-Neto, U., Prince, J.L.: Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Trans. Med. Imaging* **21**(2), 109–121 (2002)
- Han, X., Xu, C., Prince, J.L.: A topology preserving level set method for geometric deformable models. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(6), 755–768 (2003)
- He, J., Kim, C.S., Kuo, C.C.J.: *Interactive segmentation techniques: Algorithms and performance evaluation*. Springer Publishing Company, Incorporated (2013)
- Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(2), 504–511 (2013)
- Ibrahim, M., Chen, K., Rada, L.: An improved model for joint segmentation and registration based on linear curvature smoother. *J. Algorith. Comput. Technol.* **10**(4), 314–324 (2016)
- Jung, M., Bresson, X., Chan, T.F., Vese, L.A.: Nonlocal Mumford-Shah regularizers for color image restoration. *IEEE Trans. Image Process.* **20**(6), 1583–1598 (2011)
- Karaçali, B., Davatzikos, C.: Estimating topology preserving and smooth displacement fields. *IEEE Trans. Med. Imag.* **23**(7), 868–880 (2004)
- Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *Int. J. Comput. Vis.* **1**(4), 321–331 (1988)
- Kihara, Y., Soloviev, M., Chen, T.: In the shadows, shape priors shine: Using occlusion to improve multi-region segmentation. pp. 392–401. IEEE (2016)
- Klodt, M., Cremers, D.: A convex framework for image segmentation with moment constraints. In: *13th IEEE International Conference on Computer Vision (ICCV)*, pp. 2236–2243 (2011)
- Kong, T., Rosenfeld, A.: Digital topology: Introduction and survey. *Comput. Vision Graph. Image Process.* **48**(3), 357–393 (1989)
- Latecki, L.J.: D Well-composed pictures. *Graph. Models Image Process.* **59**

- Le Guyader, C., Gout, C.: Geodesic active contour under geometrical conditions: Theory and 3D applications. *Numer. Algorith.* **48**(1), 105–133 (2008)
- Le Guyader, C., Vese, L.A.: Self-repelling snakes for topology-preserving segmentation models. *IEEE Trans. Image Process.* **17**(5), 767–779 (2008)
- Le Guyader, C., Vese, L.A.: A combined segmentation and registration framework with a nonlinear elasticity smoother. *Comput. Vis. Image Underst.* **115**(12), 1689–1709 (2011)
- Lee, Y.T., Lam, K.C., Lui, L.M.: Landmark-matching transformation with large deformation via n -dimensional quasi-conformal maps. *J. Sci. Comput.* **67**(3), 926–954 (2016)
- Lehto, O., Virtanen, K.: *Quasiconformal Mappings in the Plane*. Springer (1973)
- Li, C., Kao, C., Gore, J.C., Ding, Z.: Implicit Active Contours Driven by Local Binary Fitting Energy. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2007)
- Liu, C., Ng, M.K.P., Zeng, T.: Weighted variational model for selective image segmentation with application to medical images. *Pattern Recogn.* **76**, 367–379 (2018)
- Liu, J., Tai, X.C., Luo, S.: Convex shape prior for deep neural convolution network based eye fundus images segmentation (2020). <https://arxiv.org/abs/2005.07476>
- Liu, Y., Yu, Y.: Interactive image segmentation based on level sets of probabilities. *IEEE Trans. Vis. Comput. Graph.* **18**(2), 202–213 (2012)
- Lord, N., Ho, J., Vemuri, B., Eisenschenk, S.: Simultaneous registration and parcellation of bilateral hippocampal surface pairs for local asymmetry quantification. *IEEE Trans. Med. Imaging* **26**(4), 471–478 (2007)
- Luo, S., Tai, X.C., Huo, L., Wang, Y., Glowinski, R.: Convex shape prior for multi-object segmentation using a single level set function. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 613–621 (2019)
- Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(2), 158–175 (1995)
- McGuinness, K., O'Connor, N.E.: A comparative evaluation of interactive segmentation algorithms. *Pattern Recogn.* **43**(2), 434–444 (2010). *Interactive Imaging and Vision*
- Mille, J.: Narrow band region-based active contours and surfaces for 2D and 3D segmentation. *Comput. Vis. Image Underst.* **113**(9), 946–965 (2009)
- Miller, M., Trounev, A., Younes, L.: On the Metrics and Euler-Lagrange Equations of Computational Anatomy. *Annu. Rev. B. Eng.* **4**(1), 375–405 (2002)
- Modersitzki, J.: *Numerical Methods for Image Registration*. Oxford University Press (2004)
- Mortensen, E., Morse, B., Barrett, W., Udupa, J.: Adaptive boundary detection using ‘live-wire’ two-dimensional dynamic programming. In: *Proceedings Computers in Cardiology*, pp. 635–638 (1992)
- Mory, B., Ardon, R.: Fuzzy region competition: A convex two-phase segmentation framework. In: Sgallari, F., Murli, A., Paragios, N. (eds.) *Scale Space and Variational Methods in Computer Vision*, pp. 214–226. Springer, Berlin/Heidelberg (2007)
- Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
- Musse, O., Heitz, F., Armpach, J.P.: Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Trans. Image Process.* **10**(7), 1081–1093 (2001)
- Nakahara, M.: *Geometry, Topology and Physics*. Taylor & Francis (2003)
- Noblet, V., Heinrich, C., Heitz, F., Armpach, J.P.: 3-D deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization. *IEEE Trans. Image Process.* **14**(5), 553–566 (2005)
- Nosrati, M.S., Hamarneh, G.: Incorporating prior knowledge in medical image segmentation: a survey. *arXiv e-prints* (2016). <https://arxiv.org/abs/1607.01092>
- Ohlander, R., Price, K., Reddy, D.R.: Picture segmentation using a recursive region splitting method. *Comput. Graphics Image Process.* **8**(3), 313–333 (1978)
- Oliveira, F.P., ao Manuel R.S. Tavares, J.: Medical image registration: a review. *Comput. Meth. Biomech. Biomed. Eng.* **17**(2), 73–93 (2014)

- Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
- Ozeré, S., Gout, C., Le Guyader, C.: Joint segmentation/registration model by shape alignment via weighted total variation minimization and nonlinear elasticity. *SIAM J. Imag. Sci.* **8**(3), 1981–2020 (2015)
- Pennec, X., Stefanescu, R., Arsigny, V., Fillard, P., Ayache, N.: Riemannian elasticity: A statistical regularization framework for non-linear registration. In: Duncan, J.S., Gerig, G. (eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005: 8th International Conference, Palm Springs, 26–29 Oct 2005, Proceedings, Part II*, pp. 943–950. Springer, Berlin/Heidelberg (2005)
- Precioso, F., Barlaud, M.: B-spline active contour with handling of topology changes for fast video segmentation. *EURASIP J. Adv. Signal Process.* **2002**(6), 555–560 (2002)
- Price, B.L., Morse, B., Cohen, S.: Geodesic graph cut for interactive image segmentation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3161–3168 (2010)
- Rabbitt, R., Weiss, J., Christensen, G., Miller, M.: Mapping of Hyperelastic Deformable Templates Using the Finite Element Method. In: *Proceedings SPIE*, vol. 2573, pp. 252–265. SPIE (1995)
- Rada, L., Chen, K.: A new variational model with dual level set functions for selective segmentation. *Commun. Comput. Phys.* **12**(1), 261–283 (2012)
- Rada, L., Chen, K.: Improved selective segmentation model using one level-set. *J. Alg. Comput. Technol.* **7**(4), 509–540 (2013)
- Rao, S.R., Mobahi, H., Yang, A.Y., Sastry, S.S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: Zha, H., Taniguchi, R.I., Maybank, S. (eds.) *Computer Vision – ACCV 2009*, pp. 135–146. Springer Berlin/Heidelberg (2010)
- Roberts, M., Chen, K., Irion, K.: A convex geodesic selective model for image segmentation. *J. Math. Imaging Vision* **61**(5), 482–503 (2019)
- Rochery, M., Jermyn, I., Zerubia, J.: Phase field models and higher-order active contours. In: Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1, vol. 2, pp. 970–976 (2005)
- Rochery, M., Jermyn, I.H., Zerubia, J.: Higher order active contours. *Int. J. Comput. Vis.* **69**(1), 27–42 (2006)
- Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1, pp. 993–1000 (2006)
- Rumpf, M., Wirth, B.: A nonlinear elastic shape averaging approach. *SIAM J. Imag. Sci.* **2**(3), 800–833 (2009)
- Schaeffer, H., Duggan, N., Le Guyader, C., Vese, L.: Topology preserving active contours. *Commun. Math. Sci.* **12**(7), 1329–1342 (2014)
- Sederberg, T., Parry, S.: Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.* **20**(4), 151–160 (1986)
- Ségonne, F.: Active contours under topology control—genus preserving level sets. *Int. J. Comput. Vis.* **79**(2), 107–117 (2008)
- Ségonne, F., Pacheco, J., Fischl, B.: Geometrically accurate topology-correction of cortical surfaces using nonseparating loops. *IEEE Trans. Med. Imaging* **26**(4), 518–529 (2007)
- Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **13**(1), 146–168 (2004)
- Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
- Siu, C.Y., Chan, H.L., Lui, L.M.: Image segmentation with partial convexity prior using discrete conformality structures. *SIAM J. Image Sci.* **13**(4), 2105–2139 (2020)
- Sotiras, A., Davatzikos, C., Paragios, N.: Deformable medical image registration: A survey. *IEEE Trans. Med. Imaging* **32**(7), 1153–1190 (2013)

- Spencer, J., Chen, K.: A convex and selective variational model for image segmentation. *Commun. Math. Sci.* **13**(6), 1453–1452 (2015)
- Storath, M., Weinmann, A.: Fast partitioning of vector-valued images. *SIAM J. Imag. Sci.* **7**(3), 1826–1852 (2014)
- Sundaramoorthi, G., Yezzi, A.: More-than-topology-preserving flows for active contours and polygons. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Vol. 1, Vol. 2, pp. 1276–1283 (2005)
- Taghanaki, S.A., Abhishek, K., Cohen, J.P., Cohen-Adad, J., Hamarneh, G.: Deep semantic segmentation of natural and medical images: a review. arXiv e-prints (2019). <https://arxiv.org/abs/1910.07655>
- Tang, T., Chung, A.: Non-rigid image registration using graph-cuts. In: Medical Image Computing and Computer-Assisted Intervention: MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 10, pp. 916–924 (2007)
- Thehjani, A., Chen, K.: An augmented Lagrangian method for solving a new variational model based on gradients similarity measures and high order regularization for multimodality registration. *Inverse Prob. Imaging* **13**(2), 309–335 (2019)
- Thierbach, K., Bazin, P.L., Gavriilidis, F., Kirilina, E., Jäger, C., Morawski, M., Geyer, S., Weiskopf, N., Scherf, N.: Deep Learning meets Topology-preserving Active Contours: towards scalable quantitative histology of cortical cytoarchitecture. *bioRxiv* (2018)
- Thiruvenkadam, S.R., Chan, T.F., Hong, B.-W.: Segmentation under occlusions using selective shape prior. *SIAM J. Imaging Sci.* **1**(1), 115–142 (2008)
- Tustison, N.J., Avants, B.B., Siqueira, M., Gee, J.C.: Topological well-composedness and glamorous glue: A digital gluing algorithm for topologically constrained front propagation. *IEEE Trans. Image Process.* **20**(6), 1756–1761 (2011)
- Unal, G., Slabaugh, G.: Coupled PDEs for non-rigid registration and segmentation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, pp. 168–175 (2005)
- Vemuri, B., Ye, J., Chen, Y., Leonard, C.: Image Registration via level-set motion: Applications to atlas-based segmentation. *Med. Image Anal.* **7**(1), 1–20 (2003)
- Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah Model. *Int. J. Comput. Vis.* **50**(3), 271–293 (2002)
- Vese, L.A., Le Guyader, C.: Variational Methods in Image Processing. *Mathematical and Computational Imaging Sciences Series*. Chapman & Hall/CRC, Taylor & Francis (2015)
- Waggoner, J., Zhou, Y., Simmons, J., Graef, M.D., Wang, S.: Topology-preserving multi-label image segmentation. In: 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 1084–1091 (2015)
- Wang, L., Li, C., Sun, Q., Xia, D., Kao, C.Y.: Active contours driven by local and global intensity fitting energy with application to brain mr image segmentation. *Comput. Med. Imaging Graph.* **33**(7), 520–531 (2009)
- Wells, W.M., Viola, P., Atsumi, H., Nakajima, S., Kikinis, R.: Multi-modal volume registration by maximization of mutual information. *Med. Image Anal.* **1**(1), 35–51 (1996)
- Werlberger, M., Pock, T., Unger, M., Bischof, H.: A variational model for interactive shape prior segmentation and real-time tracking. In: X.C. Tai, K. Mørken, M. Lysaker, K.A. Lie (eds.) *Scale Space and Variational Methods in Computer Vision*, pp. 200–211. Springer, Berlin/Heidelberg (2009)
- Wirth, B.: On the Γ -limit of joint image segmentation and registration functionals based on phase fields. *Interfaces Free Bound.* **18**(4), 441–477 (2016)
- Wu, G., Wang, L., Gilmore, J., Lin, W., Shen, D.: Joint segmentation and registration for infant brain images. In: Menze, B., Langs, G., Montillo, A., Kelm, M., Müller, H., Zhang, S., Cai, T.W., Metaxas, D. (eds.) *Medical Computer Vision: Algorithms for Big Data: International Workshop, MCV 2014, Held in Conjunction with MICCAI 2014, Cambridge, 18 Sept 2014, Revised Selected Papers*, pp. 13–21. Springer International Publishing (2014)
- Xu, Z., Niethammer, M.: DeepAtlas: Joint semi-supervised learning of image registration and segmentation. arXiv e-prints (2019). <https://arxiv.org/abs/1904.08465>

- Yanovsky, I., Thompson, P.M., Osher, S., Leow, A.D.: Topology preserving log-unbiased nonlinear image registration: Theory and implementation. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, pp. 1–8 (2007)
- Yatziv, L., Bartesaghi, A., Sapiro, G.: $O(N)$ implementation of the fast marching algorithm. *J. Comput. Phys.* **212**(2), 393–399 (2006)
- Yezi, A., Zollei, L., Kapur, T.: A variational framework for joint segmentation and registration. In: Mathematical Methods in Biomedical Image Analysis, pp. 44–51. IEEE-MMBIA (2001)
- Yotter, R.A., Dahnke, R., Thompson, P.M., Gaser, C.: Topological correction of brain surface meshes using spherical harmonics. *Hum. Brain Mapp.* **32**(7), 1109–1124 (2011)
- Yu, W., Lee, H.K., Hariharan, S., Bu, W., Ahmed, S.: Evolving generalized Voronoï diagrams for accurate cellular image segmentation. *Cytometry. Part A J. Int. Soc. Anal. Cytol.* **77A**(4), 379–386 (2010)
- Zagorchev, L., Goshtasby, A.: A comparative study of transformation functions for nonrigid image registration. *IEEE Trans. Image Process.* **15**(3), 529–538 (2006)
- Zhang, D., Chen, K.: A novel diffeomorphic model for image registration and its algorithm. *J. Math. Imaging Vision* **60**(8), 1261–1283 (2018)
- Zhang, J., Chen, K., Yu, B., Gould, D.: A local information based variational model for selective image segmentation. *Inverse Prob. Imaging* **8**(1), 293–320 (2014)
- Zhou, T., Ruan, S., Canu, S.: A review: Deep learning for medical image segmentation using multi-modality fusion. *Array* **3–4**, 100004 (2019)
- Zhu, H., Meng, F., Cai, J., Lu, S.: Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *J. Vis. Commun. Image Represent.* **34**, 12–27 (2016)



Recent Developments of Surface Parameterization Methods Using Quasi-conformal Geometry

43

Gary P. T. Choi and Lok Ming Lui

Contents

Introduction	1484
Previous Works on Surface Parameterization	1486
Mesh Parameterization	1486
Point Cloud Parameterization	1487
Mathematical Background	1487
Conformal Maps	1488
Quasi-conformal Maps	1488
Linear Beltrami Solver (LBS)	1491
Beltrami Holomorphic Flow (BHF)	1492
Teichmüller Maps	1493
Mesh Parameterization Using Quasi-conformal Geometry	1494
Genus-0 Closed Triangle Meshes	1495
Simply Connected Open Triangle Meshes	1501
Multiply Connected Open Triangle Meshes	1507
Point Cloud Parameterization Using Conformal and Quasi-conformal Geometry	1509
Genus-0 Point Clouds	1511
Point Clouds with Disk Topology	1512
Applications	1516
Conclusion	1518
References	1518

G. P. T. Choi

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

e-mail: ptchoi@mit.edu

L. M. Lui (✉)

Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China

e-mail: lmlui@math.cuhk.edu.hk

Abstract

Surface parameterization is of fundamental importance for many tasks in computer vision and imaging. In recent years, computational quasi-conformal geometry has become an emerging tool for the design of efficient and accurate parameterization methods for both surface meshes and point clouds. More specifically, using quasi-conformal (QC) theory, it is possible to reduce the geometric distortion and achieve conformal parameterizations for surfaces with different topology easily. It is also possible to achieve surface parameterizations that satisfy certain prescribed conditions, such as landmark constraints, with a minimal quasi-conformal distortion. In this article, we give an overview of the recent advances in surface parameterization using quasi-conformal geometry.

Keywords

Surface parameterization · Quasi-conformal geometry · Conformal map · Quasi-conformal map · Mesh · Point cloud

Introduction

Surface parameterization refers to the process of finding a one-to-one correspondence between a complicated surface and a simple parameter domain. It has widespread applications in computer graphics, vision, imaging, and also many other areas in science, engineering, and medicine, such as medical shape analysis (Zhao et al. 2019), greedy routing (Li et al. 2015), virtual broadcasting (Yueh et al. 2020), and topology optimization (Vogiatzis et al. 2018). The parameter domain depends on the topology of the given surface. For simply connected open surfaces in \mathbb{R}^3 , common choices of the parameter domain include the unit disk, the unit square, a rectangle, or a more flexible planar domain. For multiply connected open surfaces, it is common to parameterize the surfaces onto a planar circle domain with circular holes. For genus-0 closed surfaces, it is common to use the unit sphere as the parameter domain. For other high-genus surfaces, more complicated fundamental domains are often considered. Therefore, the surface topology plays an important role in the development of surface parameterization methods. Figure 1 shows several examples of parameterization of surfaces with different topology.

Given a surface and a target parameter domain, there are numerous ways of finding a parameterization mapping from the surface onto the parameter domain. In general, it is desirable to find a low-distortion parameterization such that the geometric information of the surface is preserved as much as possible in the simple domain. However, it is well-known that isometric (distance-preserving) mappings are not possible for general surfaces. In other words, geometric distortions unavoidable exist under surface parameterization. Therefore, different distortion criteria and measures have been considered in the development of surface parameterization methods. One major class of surface parameterization methods is the conformal parameterization, which preserves angles and hence the local geometry of the

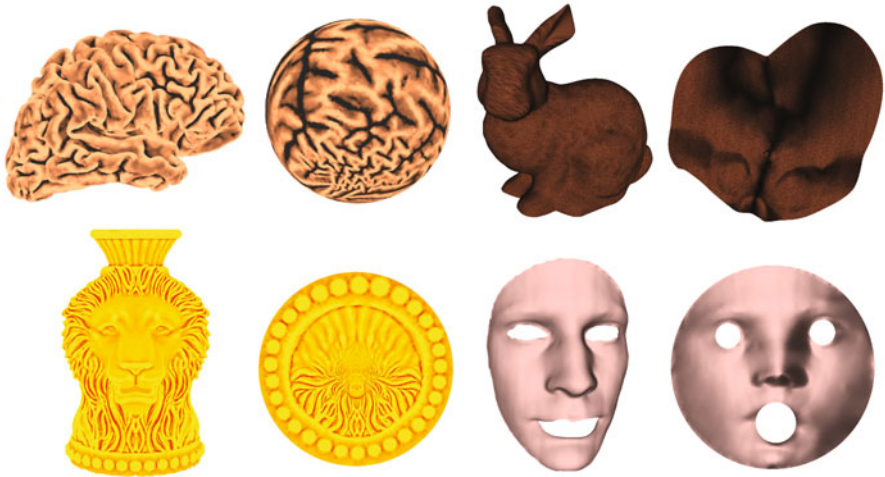


Fig. 1 Parameterization of surfaces with different topology. The top left panel shows a spherical conformal parameterization of a genus-0 closed surface. (Image adapted from Choi et al. 2015). The top right panel shows a free-boundary conformal parameterization of a simply connected open surface. (Image adapted from Choi et al. 2020a). The bottom left panel shows a disk conformal parameterization of a simply connected open surface. (Image adapted from Choi and Lui 2015). The bottom right panel shows a poly-annulus conformal parameterization of a multiply connected open surface. (Image adapted from Choi et al. 2021)

surfaces. Another major class of surface parameterization methods is the area-preserving (authalic) parameterization, which focuses on the preservation of the area elements. One may also look for parameterizations that achieve a balance between angle and area preservation or parameterizations that minimize the distortions subject to additional constraints such as prescribed landmark correspondences.

In the discrete case, surfaces are usually represented using either triangle meshes or point clouds. Each triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ consists of a set of vertices \mathcal{V} , a set of edges \mathcal{E} connecting the vertices, and a set of triangular faces \mathcal{F} . Each point cloud \mathcal{P} only consists of the vertex information but not the connectivity between the vertices. Because of the difference in the available geometric information, the developments of parameterization methods for meshes and point clouds are usually handled differently. Two examples of triangle meshes and point clouds with the parameterization results are shown in Fig. 2.

In recent years, computational quasi-conformal geometry has become a subject of great interest for the design of parameterization methods for both meshes and point clouds. Specifically, quasi-conformal theory has been utilized for reducing the conformal distortion of some prior parameterization methods to achieve conformal parameterizations. Also, for some situations where conformal parameterizations are not possible due to other prescribed constraints, quasi-conformal parameterizations with optimized conformal distortion can be obtained using computational tools based on quasi-conformal theory.

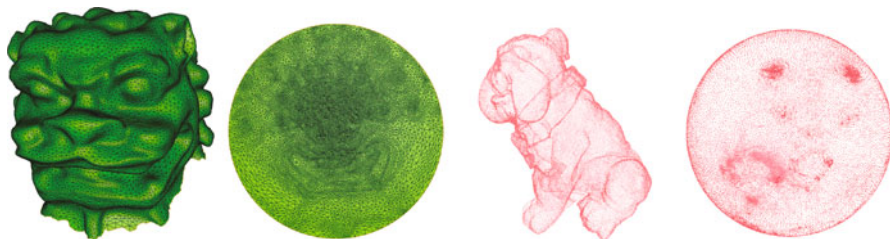


Fig. 2 Examples of mesh and point cloud parameterizations. Left: A simply connected open triangle mesh and the disk conformal parameterization. (Image adapted from Choi and Lui 2015). Right: A genus-0 point cloud and the spherical conformal parameterization. (Image adapted from Choi et al. 2016)

In this survey, we give an overview of the recent developments of surface parameterization methods using quasi-conformal geometry. Below, we first review some previous works on mesh and point cloud parameterization in section “[Previous Works on Surface Parameterization](#)”. In section “[Mathematical Background](#)”, we introduce the basic concepts of conformal and quasi-conformal maps. We then describe the recent advances in mesh parameterization and point cloud parameterization based on quasi-conformal geometry in sections “[Mesh Parameterization Using Quasi-conformal Geometry](#)” and “[Point Cloud Parameterization Using Conformal and Quasi-conformal Geometry](#)”, respectively. In section “[Applications](#)”, we review some applications of the conformal and quasi-conformal mapping methods in science, engineering, and medicine. A concluding remark is given in section “[Conclusion](#)”.

Previous Works on Surface Parameterization

Mesh Parameterization

Over the past several decades, numerous mesh parameterization methods have been developed. Readers are referred to Floater and Hormann (2005), Sheffer et al. (2006), and Hormann et al. (2007) for detailed surveys on the subject. Below, we highlight some recent works on mesh parameterization.

In recent years, conformal parameterization methods have been extensively studied (see Gu and Yau 2008; Gu et al. 2020 for a comprehensive discussion). Among all conformal parameterization methods, one common approach is to make use of harmonic energy minimization (Gu et al. 2004; Lai et al. 2014). Another common approach is to utilize surface Ricci flow (Jin et al. 2008; Yang et al. 2009; Zhang et al. 2014) (see Zhang et al. 2015 for a survey). Other notable methods for computing conformal parameterizations include the slit map (Yin et al. 2008), Koebe’s iteration (Zeng et al. 2009), metric scaling (Ben-Chen et al.

2008), boundary first flattening (Sawhney and Crane 2017), and conformal energy minimization (Yueh et al. 2017).

Area-preserving mesh parameterization methods have also been widely studied in recent years. Recent works include the Lie advection method (Zou et al. 2011), the optimal mass transportation (OMT) method (Zhao et al. 2013; Su et al. 2016; Nadeem et al. 2016; Pumarola et al. 2019; Giri et al. 2021; Lei and Gu 2021; Choi et al. 2022), stretch energy minimization (Yueh et al. 2019), and density-equalizing maps (Choi and Rycroft 2018; Choi et al. 2020b).

Besides, there are many other energy minimization approaches for computing mesh parameterizations in computer graphics. Typically, these approaches define some distortion measures and attempt to minimize them to produce the desired effects. Recent works include the advanced MIPS method (Fu et al. 2015), symmetric Dirichlet energy (Smith and Schaefer 2015), scalable locally injective mappings (SLIM) (Rabinovich et al. 2017), isometry-aware preconditioning (Claici et al. 2017), progressive parameterization (Liu et al. 2018), and efficient bijective parameterizations (Su et al. 2020).

Point Cloud Parameterization

With the advancement of 3D data acquisition techniques, the use of point clouds has been increasingly popular in recent decades. For this reason, there is also an increasing interest in the development of point cloud parameterization methods for the shape analysis and processing of point clouds.

In 2004, Zwicker and Gotsman proposed a spherical parameterization method for genus-0 point clouds. In 2006, Tewari et al. proposed a doubly periodic global parameterization method for genus-1 point clouds. In 2010, Zhang et al. developed an as-rigid-as-possible meshless parameterization method for point clouds with disk topology. In 2013, Meng et al. proposed a self-organizing radial basis function (RBF) neural network method for point cloud parameterization.

For the conformal parameterization of point clouds, one important component is the approximation of the Laplacian operator on point clouds. In recent years, several point cloud Laplacian approximation methods have been proposed, including the moving least squares (MLS) method (Belkin et al. 2009; Liang et al. 2012; Liang and Zhao 2013), the local mesh method (Lai et al. 2013; Choi et al. 2022), and the non-manifold Laplacian method (Sharp and Crane 2020).

Mathematical Background

In this section, we review the concepts of conformal and quasi-conformal maps. Readers are referred to Lehto (1973), Gardiner and Lakic (2000), and Ahlfors (2006) for more details.

Conformal Maps

Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be a map on the complex plane \mathbb{C} . Write $f(z) = f(x, y) = u(x, y) + iv(x, y)$, where $z = x + iy$, i is the imaginary number with $i^2 = -1$, and u, v are real-valued functions. Suppose the derivative of f is nonzero everywhere. f is said to be *conformal* if it satisfies the Cauchy-Riemann equations:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{1}$$

If we denote the following:

$$\frac{\partial f}{\partial \bar{z}} = f_{\bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right) \quad \text{and} \quad \frac{\partial f}{\partial z} = f_z = \frac{1}{2} \left(\frac{\partial f}{\partial x} - i \frac{\partial f}{\partial y} \right), \tag{2}$$

then Equation (1) can be rewritten as follows:

$$\frac{\partial f}{\partial \bar{z}} = 0. \tag{3}$$

Conformal maps preserve angles and hence the local geometry. Intuitively, under a conformal map, infinitesimal circles are mapped to infinitesimal circles (see Fig. 3).

Quasi-conformal Maps

Quasi-conformal maps are a generalization of conformal maps. More specifically, an orientation-preserving homeomorphism $f : \mathbb{C} \rightarrow \mathbb{C}$ is said to be *quasi-conformal* if it satisfies the Beltrami equation:

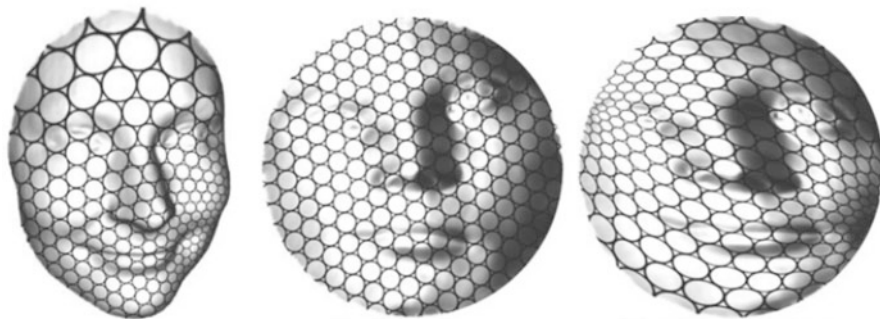


Fig. 3 An illustration of conformal and quasi-conformal maps. (Image adapted from Lui et al. 2014). Left: A surface with a circle packing texture. Middle: A conformal map of the surface onto the unit disk. Note that the small circles are mapped to small circles. Right: A quasi-conformal map of the surface onto the unit disk. Note that the small circles are mapped to small ellipses

$$\frac{\partial f}{\partial \bar{z}} = \mu_f(z) \frac{\partial f}{\partial z} \tag{4}$$

for some complex-valued function μ_f with $\|\mu_f\|_\infty < 1$. μ is called the *Beltrami coefficient* of the map f . Considering the first order approximation of f around a point p with respect to its local parameter, we have the following:

$$\begin{aligned} f(z) &= f(p) + f_z(p)(z - p) + f_{\bar{z}}(p)\overline{z - p} \\ &= f(p) + f_z(p) (z - p + \mu_f(p)\overline{z - p}). \end{aligned} \tag{5}$$

This gives the following:

$$|f(z) - f(p)| = |f_z(p)| |z - p + \mu_f(p)\overline{z - p}| \tag{6}$$

and hence:

$$|f_z(p)| \left(1 - |\mu_f(p)|\right) |z - p| \leq |f(z) - f(p)| \leq |f_z(p)| \left(1 + |\mu_f(p)|\right) |z - p|. \tag{7}$$

This shows that an infinitesimal circle is mapped to an infinitesimal ellipse with bounded eccentricity under a quasi-conformal map (see Figs. 3 and 4), where the maximal magnification factor is $|f_z(p)| (1 + |\mu_f(p)|)$, the maximal shrinkage factor is $|f_z(p)| (1 - |\mu_f(p)|)$, and the maximal dilatation of f is as follows:

$$K(f) = \frac{1 + \|\mu_f\|_\infty}{1 - \|\mu_f\|_\infty}. \tag{8}$$

Also, note that the last equality in Equation (7) holds if and only if:

$$z - p = c\mu_f(p)\overline{z - p} \tag{9}$$

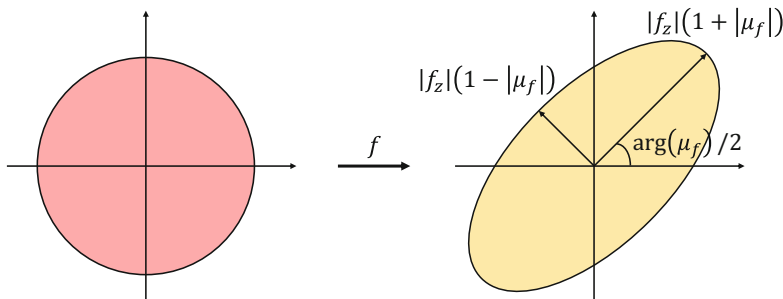


Fig. 4 An illustration of quasi-conformal maps. (Image adapted from Choi et al. 2020c)

for some $c \in \mathbb{R}$, which gives the following:

$$\arg(z - p) = \arg(\mu_f(p)) - \arg(z - p) \Leftrightarrow \arg(z - p) = \arg(\mu_f(p))/2. \tag{10}$$

This shows that the orientation change of the major axis of the ellipse is $\arg(\mu_f(p))/2$. From the above, it can be observed that the Beltrami coefficient μ encodes useful information of the quasi-conformality of the mapping f .

The bijectivity of the map f is also related to the Beltrami coefficient of it. More specifically, if $f(z) = f(x + iy) = u(x, y) + iv(x, y)$, where u, v are two real-valued functions, the Jacobian of f is given by the following:

$$\begin{aligned} J_f &= u_x v_y - u_y v_x \\ &= \frac{1}{4} \left((u_x + v_y)^2 + (u_y - v_x)^2 - (u_x - v_y)^2 - (u_y + v_x)^2 \right) \\ &= \left| \frac{1}{2}(f_x - if_y) \right|^2 - \left| \frac{1}{2}(f_x + if_y) \right|^2 \\ &= |f_z|^2 - |f_{\bar{z}}|^2 \\ &= |f_z|^2 (1 - |\mu_f|)^2, \end{aligned} \tag{11}$$

which indicates that J_f is positive everywhere if $\|\mu_f\|_\infty < 1$.

The correspondence between Beltrami coefficients and quasi-conformal maps is given by the measurable Riemann mapping theorem (Gardiner and Lakic 2000):

Theorem 1 (Measurable Riemann mapping theorem). *If $\mu : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$ be a Lebesgue measurable function with $\|\mu\|_\infty < 1$. There exists a quasi-conformal homeomorphism $\phi : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$ in the Sobolev space $W^{1,2}(\overline{\mathbb{C}})$ satisfying the Beltrami equation (4) in the distribution sense. By fixing θ, l , and ∞ , ϕ is uniquely determined for any given μ .*

In other words, a quasi-conformal map can be uniquely determined by its associated Beltrami coefficient under suitable normalization.

Given two quasi-conformal maps $f : \Omega_1 \subset \mathbb{C} \rightarrow \Omega_2 \subset \mathbb{C}$ and $g : \Omega_2 \subset \mathbb{C} \rightarrow \Omega_3 \subset \mathbb{C}$, the Beltrami coefficient of the composition map $g \circ f$ is given by the following composition formula:

$$\mu_{g \circ f} = \frac{\mu_f + \frac{\overline{f_z}}{f_z}(\mu_g \circ f)}{1 + \frac{\overline{f_z}}{f_z} \overline{\mu_f}(\mu_g \circ f)}. \tag{12}$$

In particular, if $\mu_{f^{-1}} = \mu_g$, then:

$$\mu_f + \frac{\overline{f_z}}{f_z}(\mu_g \circ f) = \mu_f + \frac{\overline{f_z}}{f_z}(\mu_{f^{-1}} \circ f) = \mu_f + \frac{\overline{f_z}}{f_z} \left(-\frac{f_z}{\overline{f_z}} \mu_f \right) = 0, \tag{13}$$

and hence $g \circ f$ is conformal. This idea of quasi-conformal composition plays an important role in many recent parameterization methods.

To define quasi-conformal maps between two Riemann surfaces, the concept of Beltrami differential is used. More specifically, given any Riemann surface \mathcal{S} , a Beltrami differential $\mu(z) \frac{\overline{dz}}{dz}$ is an assignment to each chart (U_α, ϕ_α) of an L_∞ complex-valued function μ_α defined on local parameter z_α , such that:

$$\mu_\alpha \frac{\overline{dz_\alpha}}{dz_\alpha} = \mu_\beta \frac{\overline{dz_\beta}}{dz_\beta} \tag{14}$$

on the domain which is also covered by another chart (U_β, ϕ_β) . Let $f : \mathcal{M} \rightarrow \mathcal{N}$ be an orientation-preserving diffeomorphism between two Riemann surfaces \mathcal{M}, \mathcal{N} . f is said to be quasi-conformal associated with the Beltrami differential $\mu(z) \frac{\overline{dz}}{dz}$ if for any chart (U_α, ϕ_α) on \mathcal{M} and any chart (U_β, ϕ_β) on \mathcal{N} ; the mapping $f_{\alpha\beta} := \phi_\beta \circ f \circ \phi_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha \frac{\overline{dz_\alpha}}{dz_\alpha}$.

Linear Beltrami Solver (LBS)

As described above, there is a close relationship between Beltrami coefficients and quasi-conformal maps. It is natural to ask whether one can reconstruct a quasi-conformal map f from a given complex-valued function μ easily. To achieve this task, Lui et al. developed an efficient method called the *linear Beltrami solver* (LBS) in Lui et al. (2013). The method is outlined below.

Let $f(z) = f(x + iy) = u(x, y) + iv(x, y)$ and $\mu(z) = \rho(z) + i\tau(z)$, where u, v, ρ, τ are real-valued functions. The Beltrami equation (4) can then be rewritten as follows:

$$\mu_f = \frac{(u_x - v_y) + i(v_x + u_y)}{(u_x + v_y) + i(v_x - u_y)}. \tag{15}$$

Now, we can express v_x and v_y as linear combinations of u_x and u_y :

$$\begin{aligned} -v_y &= \alpha_1 u_x + \alpha_2 u_y; \\ v_x &= \alpha_2 u_x + \alpha_3 u_y, \end{aligned} \tag{16}$$

where:

$$\alpha_1 = \frac{(\rho - 1)^2 + \tau^2}{1 - \rho^2 - \tau^2}, \quad \alpha_2 = -\frac{2\tau}{1 - \rho^2 - \tau^2}, \quad \alpha_3 = \frac{1 + 2\rho + \rho^2 + \tau^2}{1 - \rho^2 - \tau^2}. \tag{17}$$

We can also express u_x and u_y as linear combinations of v_x and v_y similarly:

$$\begin{aligned} u_y &= \alpha_1 v_x + \alpha_2 v_y; \\ -u_x &= \alpha_2 v_x + \alpha_3 v_y. \end{aligned} \tag{18}$$

Now, since $\nabla \cdot \begin{pmatrix} -v_y \\ v_x \end{pmatrix} = 0$ and $\nabla \cdot \begin{pmatrix} u_y \\ -u_x \end{pmatrix} = 0$, we have the following:

$$\nabla \cdot \left(A \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right) = 0 \text{ and } \nabla \cdot \left(A \begin{pmatrix} v_x \\ v_y \end{pmatrix} \right) = 0, \tag{19}$$

where $A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_3 \end{pmatrix}$.

In the discrete case, one can discretize the elliptic PDEs (19) as sparse positive definite linear systems. Therefore, for any given μ and some prescribed boundary conditions, one can efficiently obtain a quasi-conformal map f with the associated Beltrami coefficient being μ . See Lui et al. (2013) for more details of the computational procedure of the LBS method.

Beltrami Holomorphic Flow (BHF)

In Lui et al. (2010, 2012), Lui et al. developed another method called the *Beltrami holomorphic flow* (BHF) for reconstructing quasi-conformal maps for given Beltrami coefficients. The BHF method is based on the following theorem (Gardiner and Lakic 2000):

Theorem 2 (Beltrami holomorphic flow on $\overline{\mathbb{C}}$). *There is a 1-1 correspondence between the set of quasi-conformal maps $f : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$ that fix the points $0, 1, \infty$ and the set of smooth complex-valued functions μ on $\overline{\mathbb{C}}$ with $\|\mu\|_\infty < 1$. Here, the solution f^μ to the Beltrami equation (4) depends holomorphically on μ . Let $\{\mu(t)\}$ be a family of Beltrami coefficients, where t is a real or complex parameter. Suppose $\mu(t)$ can be written in the following form:*

$$\mu(t)(z) = \mu(z) + tv(z) + t\epsilon(t)(z), \tag{20}$$

with μ in the unit ball of $C^\infty(\mathbb{C})$, $v, \epsilon(t) \in L^\infty(\mathbb{C})$ such that $\|\epsilon(t)\|_\infty \rightarrow 0$ as $t \rightarrow 0$. Then, for all $w \in \mathbb{C}$, we have the following:

$$f^{\mu(t)}(w) = f^\mu(w) + tV(f^\mu, v)(w) + o(|t|) \tag{21}$$

locally uniformly on \mathbb{C} as $t \rightarrow 0$, where:

$$V(f^\mu, v)(w) = -\frac{f^\mu(w)(f^\mu(w) - 1)}{\pi} \int_{\mathbb{C}} \frac{v(z)(f^\mu)_z(z)^2}{f^\mu(z)(f^\mu(z) - 1)(f^\mu(z) - f^\mu(w))} dz. \tag{22}$$

In other words, given a Beltrami coefficient and the target positions of three points, one can obtain a unique quasi-conformal map. In practice, to reconstruct the quasi-conformal map, one can start with the identity map and iteratively flow the map to f^μ using BHF. See Lui et al. (2012) for more details of the computational procedure of the BHF method.

Teichmüller Maps

Teichmüller maps (T-maps) are a special class of quasi-conformal maps. A quasi-conformal map $f : \mathbb{C} \rightarrow \mathbb{C}$ is said to be a *Teichmüller map* if its associated Beltrami coefficient is of the following form:

$$\mu_f = k \frac{\bar{\phi}}{\phi}, \tag{23}$$

where ϕ is a complex-valued function and k is a constant with $k < 1$. In other words, the quasi-conformal distortion of a Teichmüller map is uniform over the entire domain. More generally, a quasi-conformal map $f : S_1 \rightarrow S_2$ between two Riemann surfaces is said to be a *Teichmüller map* associated with the quadratic differential $q = \varphi dz^2$ if its associated Beltrami differential is of the following form:

$$\mu_f = k \frac{\bar{\varphi}}{\varphi}, \tag{24}$$

where $\varphi : S_1 \rightarrow \mathbb{C}$ is a holomorphic function, $q \neq 0$ is a quadratic differential with $\|q\|_1 = \int_{S_1} |\varphi| < \infty$, and k is a constant with $k < 1$.

Another closely related concept is the extremal map. A quasi-conformal map $f : S_1 \rightarrow S_2$ is said to be *extremal* if for any quasi-conformal map $g : S_1 \rightarrow S_2$ isotopic to f relative to the boundary, we have the following:

$$K(f) \leq K(g). \tag{25}$$

Teichmüller maps and extremal maps are connected by the following theorem (Lui et al. 2014):

Theorem 3 (Landmark-matching Teichmüller map). *Let $g : \partial\mathbb{D} \rightarrow \partial\mathbb{D}$ be an orientation-preserving diffeomorphism of the boundary of the unit disk, with $g'(e^{i\theta}) \neq 0$ and $g''(e^{i\theta})$ is bounded for all θ . Let $\{p_j\}_{j=1}^n$ and $\{q_j\}_{j=1}^n$ be two sets of corresponding interior landmarks in \mathbb{D} . Then there exists a landmark-matching*

Teichmüller map $f : \mathbb{D} \rightarrow \mathbb{D}$ that is the unique extremal extension of g to \mathbb{D} , i.e., $f|_{\partial\mathbb{D}} = g$ and $f(p_j) = q_j$ for all $j = 1, 2, \dots, n$.

In other words, besides having uniform quasi-conformal distortion, Teichmüller maps are extremal in the sense that they minimize the maximal dilatation K .

In 2014, Lui et al. proposed a method called the *QC iteration* method for the computation of landmark-matching Teichmüller maps. The QC iteration method iteratively updates the Beltrami coefficient and reconstructs the associated quasi-conformal map using the LBS method until the resulting map becomes Teichmüller. More specifically, suppose the initial quasi-conformal map f_0 is associated with the Beltrami coefficient μ_0 . The method computes the following iteratively:

$$\begin{aligned} v_{n+1} &:= \mathcal{A}(\mathcal{L}(\mu_n)), \\ f_{n+1} &:= \mathbf{LBS}_{\text{LM}}(v_{n+1}), \\ \mu_{n+1} &:= \mu(f_{n+1}), \end{aligned} \tag{26}$$

until $\|v_{n+1} - v_n\|_\infty$ is less than a given stopping parameter $\epsilon > 0$. Here, \mathcal{L} is the Laplacian smoothing operator, \mathcal{A} is an averaging operator, \mathbf{LBS}_{LM} denotes the quasi-conformal map obtained by the LBS method with the prescribed landmark constraints, and $\mu(f_{n+1})$ denotes the Beltrami coefficient of f_{n+1} obtained from the Beltrami equation (4). The convergence of the QC iteration method has been proved in Lui et al. (2015).

Mesh Parameterization Using Quasi-conformal Geometry

In recent years, quasi-conformal theory has been widely used in surface mapping, registration, and visualization. For instance, Zeng et al. (2012) developed a method for computing quasi-conformal mappings between Riemann surfaces using Yamabe flow and an auxiliary metric which incorporates quasi-conformality induced from the Beltrami differential. Specifically, quasi-conformal mappings are equivalent to conformal mappings under the auxiliary metric and hence can be effectively computed. Lipman et al. (2012) computed quasi-conformal plane deformations by introducing a formula for 4-point planar warping. Weber et al. (2012) developed a method for computing piecewise linear approximations of extremal quasi-conformal maps. Lipman (2012) and Chien et al. (2016) developed methods for computing bounded distortion mappings. Wong and Zhao (2014, 2015) developed methods for computing surface mappings using discrete Beltrami flow. Zeng and Gu (2011) proposed a surface registration method using quasi-conformal curvature flow. Lui and Wen (2014) proposed a method for high-genus surface registration by computing a quasi-conformal map between the conformal embedding of the surfaces on the hyperbolic disk. Quasi-conformal theory has also been used in the development of rectilinear maps (Yang and Zeng 2020) and retinotopic maps (Tu et al. 2020; Ta

Table 1 A summary of recent mesh parameterization methods based on quasi-conformal theory

Method	Surface type	Target domain	Criterion
FLASH (Choi et al. 2015)	Topological sphere	Sphere	Conformal/ quasi-conformal
FSQC (Choi et al. 2016)	Topological sphere	Sphere	Quasi-conformal
Fast disk map (Choi and Lui 2015)	Topological disk	Disk	Conformal
Linear disk map (Choi and Lui 2018)	Topological disk	Disk	Conformal
Carotid flattening (Choi et al. 2017)	Topological disk	L-shaped	Conformal
LSQC (Qiu et al. 2019)	Topological disk	Free-boundary	Quasi-conformal
PGCP (Choi et al. 2020a)	Simply connected	Free/disk/sphere	Conformal
ACM/PACM (Choi et al. 2021)	Multiply connected	Circle domain	Conformal
QCMC (Ho and Lui 2016)	Multiply connected	Circle domain	Quasi-conformal
BHF (Ng et al. 2014)	Multiply connected	Circle domain	Teichmüller

et al. 2021). In this section, we review the latest mesh parameterization methods developed based on quasi-conformal geometry.

By the uniformization theorem, every simply connected Riemann surface is conformally equivalent to either the unit disk, the complex plane, or the Riemann sphere. Also, every multiply connected open surface is conformally equivalent to a circle domain with circular holes. Therefore, as mentioned earlier in section “[Introduction](#)”, various methods have been proposed for parameterizing surface meshes with different topology onto different parameter domains. Table 1 summarizes the recent mesh parameterization methods based on quasi-conformal theory. Below, we first introduce the parameterization methods for genus-0 closed triangle meshes and then discuss the methods for simply connected and multiply connected open triangle meshes.

Genus-0 Closed Triangle Meshes

Conformal Parameterization

In 2015, Choi et al. proposed a fast algorithm for the spherical conformal parameterization of genus-0 closed triangle meshes (see Fig. 5). More specifically, given a genus-0 closed triangle mesh \mathcal{M} , the algorithm first follows the idea in Haker et al. (2000) and punctures one triangle $T = [v_i, v_j, v_k]$ from \mathcal{M} . The punctured surface $\mathcal{M} \setminus T$ is then a simply connected open surface and hence can be mapped onto the plane by solving the Laplace equation:

$$\Delta g = 0, \tag{27}$$

where $g : \mathcal{M} \setminus T \rightarrow \mathbb{C}$ flattens the punctured mesh onto a planar triangular domain with the three mapped boundary vertices $g(v_i)$, $g(v_j)$, and $g(v_k)$ forming a

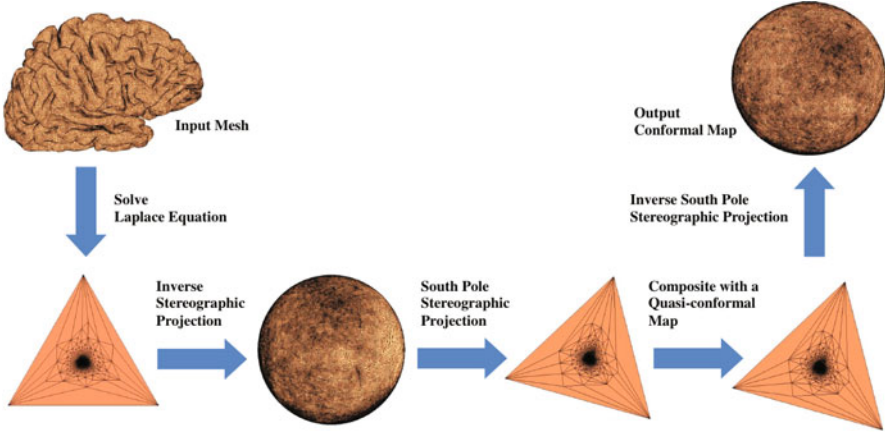


Fig. 5 An illustration of the fast spherical conformal parameterization method. (Image adapted from Choi et al. 2015)

boundary triangle with the same angle structure as T . One can then map the planar triangular domain onto the unit sphere using the inverse stereographic projection $\varphi_N^{-1} : \overline{\mathbb{C}} \rightarrow \mathbb{S}^2$, where the stereographic projection $\varphi_N : \mathbb{S}^2 \rightarrow \overline{\mathbb{C}}$ is given by the following:

$$\varphi_N(X, Y, Z) = \frac{X}{1 - Z} + i \frac{Y}{1 - Z} \tag{28}$$

and the inverse stereographic projection $\varphi_N^{-1} : \overline{\mathbb{C}} \rightarrow \mathbb{S}^2$ is given by the following:

$$\varphi_N^{-1}(z) = \left(\frac{2\text{Re}(z)}{1 + |z|^2}, \frac{2\text{Im}(z)}{1 + |z|^2}, \frac{1 - |z|^2}{1 + |z|^2} \right). \tag{29}$$

The composition map $\varphi_N^{-1} \circ g$ is then a parameterization mapping from \mathcal{M} onto the unit sphere \mathbb{S}^2 . However, the conformal distortion near the punctured triangle T , which corresponds to the north pole region of the unit sphere, is severe in the discrete case. To correct the conformal distortion there, the algorithm in Choi et al. (2015) maps the sphere to the extended complex plane using the south pole stereographic projection $\varphi_S : \mathbb{S}^2 \rightarrow \overline{\mathbb{C}}$ with the following:

$$\varphi_S(X, Y, Z) = \frac{X}{1 + Z} + i \frac{Y}{1 + Z}, \tag{30}$$

such that the south pole region of the unit sphere is mapped to the outermost part of the planar domain and the north pole region of the unit sphere is mapped to the innermost part of the planar domain. The algorithm then computes a quasi-conformal map $h : \mathbb{C} \rightarrow \mathbb{C}$ with the Beltrami coefficient $\mu_h = \mu_{(\varphi_S \circ \varphi_N^{-1} \circ g)^{-1}}$ and

with the outermost part of the domain fixed using the LBS method (Lam and Lui 2014). The composition map $h \circ \varphi_S \circ \varphi_N^{-1} \circ g$ is then conformal by the composition formula in Equation (12). Finally, the map $\varphi_S^{-1} \circ h \circ \varphi_S \circ \varphi_N^{-1} \circ g$ gives a conformal parameterization of \mathcal{M} onto the unit sphere. Moreover, the use of the Beltrami coefficients also helps ensure that the mapping is bijective (see Fig. 6).

Another spherical conformal parameterization method that utilizes quasi-conformal theory is the parallelizable global conformal parameterization (PGCP) method (Choi et al. 2020a) (see Fig. 7 for an example). The PGCP method achieves the conformal parameterization using a divide-and-conquer manner by considering

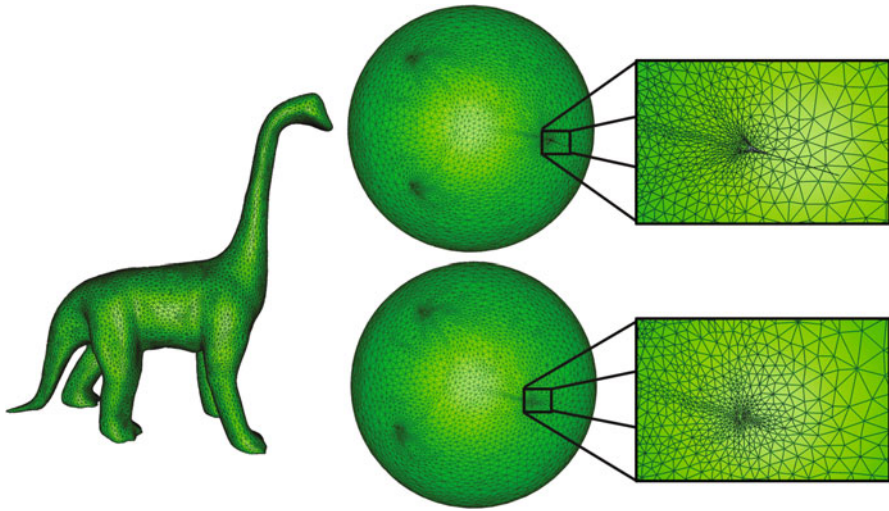


Fig. 6 The spherical conformal parameterization method in Choi et al. (2015) is capable of mapping a complicated dinosaur mesh (left) onto the unit sphere bijectively (bottom right), while the traditional method (Gu et al. 2004) (top right) produces overlaps. (Image adapted from Choi 2016)



Fig. 7 The spherical conformal parameterization of a genus-0 duck surface mesh obtained using the parallelizable global conformal parameterization (PGCP) method. (Image adapted from Choi et al. 2020a). The colors indicate the correspondence between the subdomains in the original mesh and in the parameterization result

a partition of the input triangle mesh into several submeshes. Because of the use of mesh partition, the PGCP method is capable of handling not only genus-0 closed surfaces but also simply connected open surfaces. The method will be explained in detail later in section “[Simply Connected Open Triangle Meshes](#)”.

Quasi-conformal Parameterization

In 2015, Choi et al. developed the fast landmark-aligned spherical harmonic parameterization (FLASH) method for genus-0 closed triangle meshes (see Fig. 8 for an illustration). More specifically, given two genus-0 closed triangle meshes \mathcal{S}_1 and \mathcal{S}_2 with two sets of corresponding landmarks $\{p_j\}_{j=1}^n$ and $\{q_j\}_{j=1}^n$ on \mathcal{S}_1 and \mathcal{S}_2 , respectively, denote the spherical conformal parameterization of \mathcal{S}_2 obtained by the abovementioned method in Choi et al. (2015) by $\phi_2 : \mathcal{S}_2 \rightarrow \mathbb{S}^2$. The FLASH method aims to find a spherical parameterization $f : \mathcal{S}_1 \rightarrow \mathbb{S}^2$ such that $f(p_j)$ matches $\phi_2(q_j)$ as accurately as possible for all $j = 1, 2, \dots, n$, and the conformal distortion of f is also as small as possible. To achieve this, the method first computes the spherical conformal parameterization $\phi_1 : \mathcal{S}_1 \rightarrow \mathbb{S}^2$. It then solves for a quasi-conformal map $\psi : \mathbb{S}^2 \rightarrow \mathbb{S}^2$ that minimizes the following combined energy:

$$E_{\text{combined}}(\psi) = \int |\nabla\psi|^2 + \lambda \sum_{j=1}^n |\psi(\phi_1(p_j)) - \phi_2(q_j)|^2, \tag{31}$$

where $\lambda \geq 0$ is a weighting factor for balancing the conformality and the landmark mismatch. In particular, a large λ yields a quasi-conformal map with a smaller landmark mismatch but a larger conformal distortion, while a small λ yields a smaller conformal distortion but the landmark mismatch will be larger. ϕ can be obtained by solving the following equation:

$$\Delta\psi + \lambda\delta_E(\psi - \phi_2(q_j)) = 0, \tag{32}$$

where $\delta_E(w)$ is the smooth approximation of the characteristic function:

$$\chi_E(w) = \begin{cases} 1 & \text{if } w = \phi_2(q_j) \text{ for some } j, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

The desired landmark-aligned spherical parameterization is then given by $f = \psi \circ \phi_1$. The bijectivity of the parameterization can be further enforced by modifying the norm of the Beltrami coefficient and reconstructing the associated quasi-conformal map iteratively. Figure 9 shows for some examples of landmark-aligned spherical parameterization obtained using the FLASH method.

In 2016, Choi et al. developed the fast spherical quasi-conformal parameterization (FSQC) method for the computation of spherical parameterization of genus-0

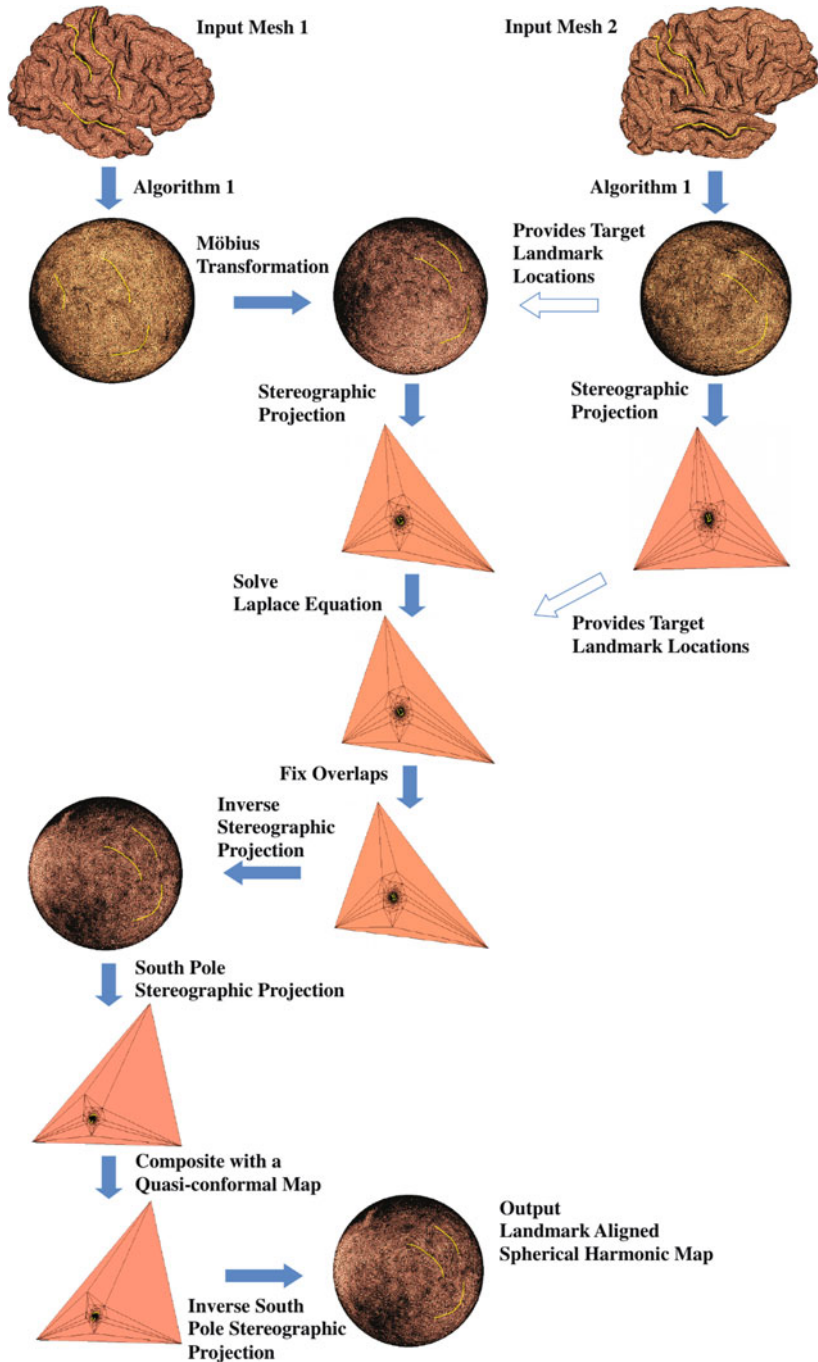


Fig. 8 An illustration of the fast landmark-aligned spherical harmonic parameterization (FLASH) method. (Image adapted from Choi et al. 2015)

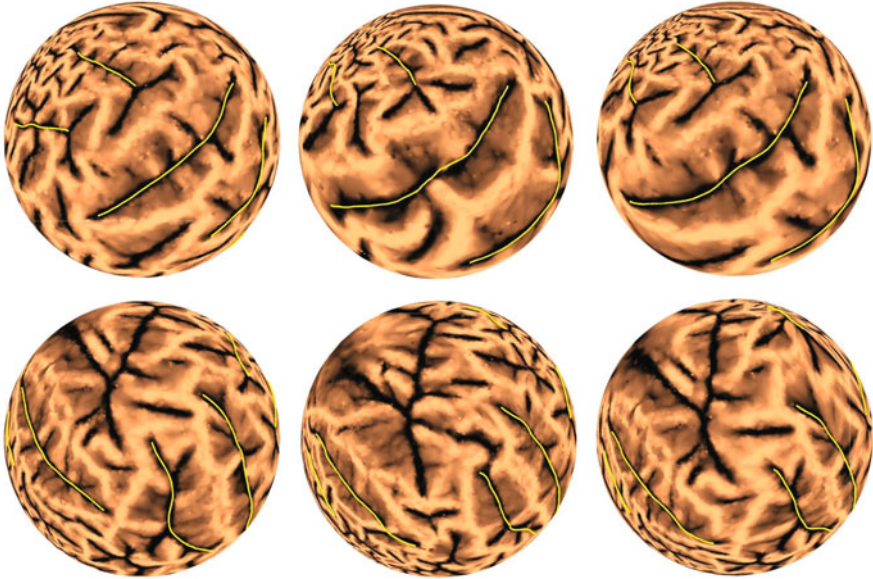


Fig. 9 Two examples of the landmark-constrained spherical quasi-conformal parameterization obtained using the FLASH method. (Image adapted from Choi et al. 2015). Each row shows an example. Left column: The spherical conformal parameterization of the source mesh. Middle column: The spherical conformal parameterization of the target mesh. Right column: The landmark-constrained quasi-conformal parameterization

closed triangle meshes with a prescribed quasi-conformal dilatation. Specifically, given any genus-0 closed triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ and a user-defined quasi-conformal dilatation $K : \mathcal{F} \rightarrow \mathbb{R}$ defined on every triangular face of the mesh, the method starts by computing the spherical conformal parameterization of \mathcal{M} using the method in Choi et al. (2015). Next, it searches for a triangle T on the spherical parameterization such that both T and its neighboring faces are the most regular and then performs a stereographic projection with respect to T to map the sphere onto the plane. Then, to achieve the prescribed dilatation K , the method constructs a Beltrami coefficient μ with the following:

$$\mu(T) = \frac{K(T) - 1}{K(T) + 1} \quad (34)$$

for every triangle T . By applying the LBS method (Lui et al. 2013) to reconstruct a quasi-conformal map on the plane associated with the Beltrami coefficient μ followed by the inverse stereographic projection, the desired spherical quasi-conformal parameterization is obtained. Figure 10 shows an example of spherical quasi-conformal parameterization obtained by the FSQC method.

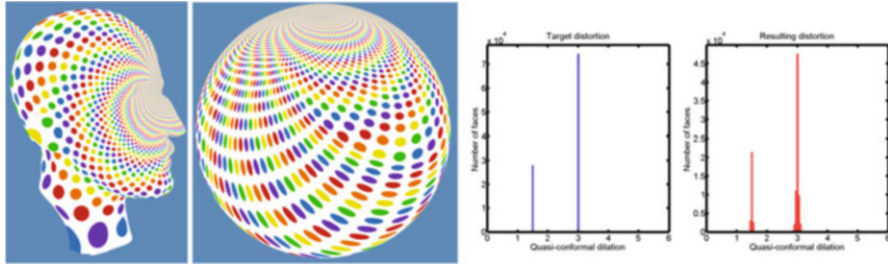


Fig. 10 An example of the fast spherical quasi-conformal parameterization (FSQC) method for genus-0 closed triangle meshes. (Image adapted from Choi et al. 2016). Left: The input genus-0 closed surface with a circle packing texture and the spherical quasi-conformal parameterization obtained by FSQC. Right: The prescribed quasi-conformal dilatation and the final dilatation of the resulting parameterization. Note that the circles on the input surface are mapped to two classes of ellipses with different eccentricity as shown in the parameterization result, which correspond to $K = 1.5$ and $K = 3$ in the target dilatation histogram, respectively

Simply Connected Open Triangle Meshes

Conformal Parameterization

In 2015, Choi and Lui proposed a fast disk conformal parameterization method for simply connected open triangle meshes (see Fig. 11). The method involves two major steps, namely, the “north pole” step and the “south pole” step. Analogous to the spherical conformal parameterization method in Choi et al. (2015), the method handles the conformal distortion at different parts of the parameter domain separately. More specifically, after getting an initial disk harmonic map by solving the Laplace equation:

$$\Delta f = 0 \tag{35}$$

subject to a circular boundary constraint, the method considers the following “north pole” step. It first maps the unit disk to the upper half plane using the Cayley transform:

$$W(z) = i \frac{1+z}{1-z}, \tag{36}$$

and composes the map with another quasi-conformal map to reduce the conformal distortion using the idea of quasi-conformal composition in Equation (12) with the boundary triangle fixed. Then, it maps the upper half plane back to the unit disk using the inverse Cayley transform:

$$W^{-1}(z) = \frac{z-i}{z+i}. \tag{37}$$

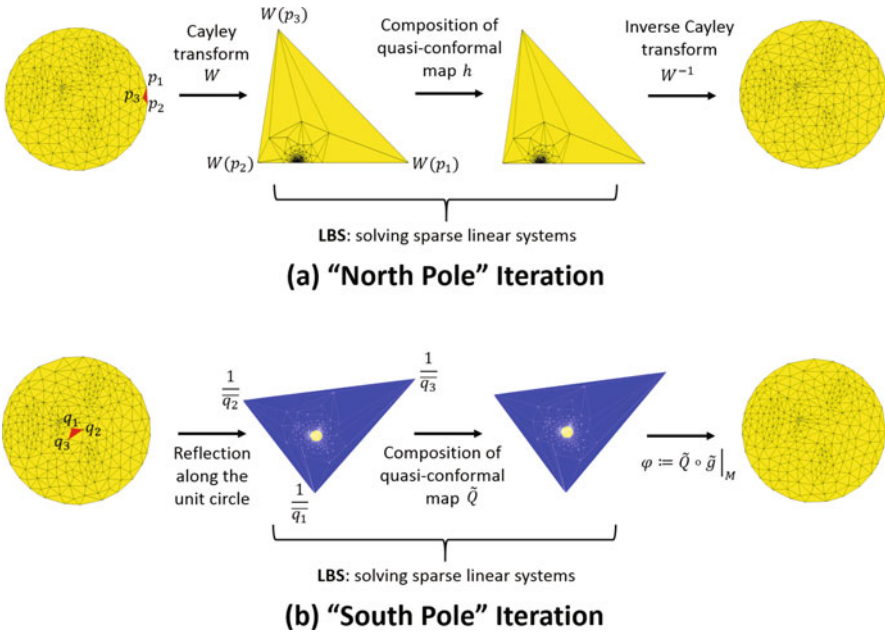


Fig. 11 An illustration of the fast disk conformal parameterization method for simply connected open triangle meshes. (Image adapted from Choi and Lui 2015). (a) "North pole" iteration. (b) "South pole" iteration

The above step helps reduce the conformal distortion at the innermost region of the disk, while the distortion at the region around $z = 1$ may still be large. Therefore, in the subsequent "south pole" step, the method uses a reflection mapping $z \mapsto \frac{1}{z}$ to reflect the disk along the unit circle, so that the outermost region of the new shape corresponds to the innermost region of the disk, which is with low conformal distortion due to the previous "north pole" step. One can then fix the outermost region and apply the idea of quasi-conformal composition again to compute a quasi-conformal map so that the conformal distortion at the region around $z = 1$ is reduced. By repeating the above procedure, one can eventually obtain a disk conformal parameterization.

In 2018, Choi and Lui proposed a linear formulation for disk conformal parameterization of simply connected open triangle meshes. The idea is to use a technique called *double covering* to turn any given simply connected open triangle mesh into a genus-0 mesh and then apply the fast spherical conformal parameterization method in Choi et al. (2015). More specifically, given a simply connected open triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, the method constructs a new mesh \mathcal{M}' by duplicating \mathcal{M} and reversing the orientation of every triangle in it. In other words, for each triangle $[v_i, v_j, v_k]$ in \mathcal{M} , the corresponding triangle in \mathcal{M}' is given by $[v'_i, v'_k, v'_j]$, where v'_i, v'_j, v'_k are copies of the vertices v_i, v_j, v_k . One can then glue \mathcal{M} and \mathcal{M}'

along their boundaries $\partial\mathcal{M}$ and $\partial\mathcal{M}'$ by identifying all the corresponding boundary vertices. The glued surface (denoted by $\tilde{\mathcal{M}}$) is then a genus-0 closed triangle mesh. Hence, one can apply the fast spherical conformal parameterization method in Choi et al. (2015) for parameterizing $\tilde{\mathcal{M}}$. By extracting the part corresponding to \mathcal{M} in the spherical parameterization and applying the stereographic projection in Equation (28), we obtain a conformal parameterization of \mathcal{M} onto a planar domain. As the planar domain may not be perfectly circular, the method further enforces the circularity of the boundary using a projection:

$$v \mapsto \frac{v}{|v|} \quad (38)$$

for all boundary vertices. Finally, to correct the conformal distortion caused by the projection, the method composes the parameterization map with another quasi-conformal map based on the composition formula in Equation (12), thereby yielding a disk conformal parameterization (see Fig. 12 for an example).

Note that the abovementioned methods compute the conformal parameterization of the input mesh globally. In case the density of the input mesh is very high or the mesh geometry is complicated, the computation of the global parameterization may be expensive and challenging. To resolve this issue, Choi et al. (2020a) proposed the parallelizable global conformal parameterization (PGCP) method (Choi et al. 2020a) (see Fig. 13 for an illustration). Specifically, the PGCP method considers partitioning the input mesh into different subdomains. For each subdomain, the discrete natural conformal parameterization (DNCP) method in Desbrun et al. (2002) is used for finding an initial free-boundary conformal flattening map. As the local parameterizations of different subdomains may not be consistent along their boundaries, the PGCP method looks for a series of conformal maps to deform



Fig. 12 The disk conformal parameterization of a simply connected open surface obtained using the linear disk map method. (Image adapted from Choi and Lui 2018)

Fig. 13 An illustration of the parallelizable global conformal parameterization (PGCP) method. (Image adapted from Choi et al. 2020a)

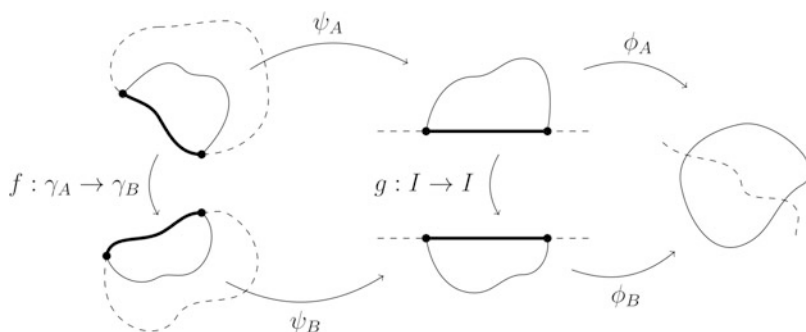
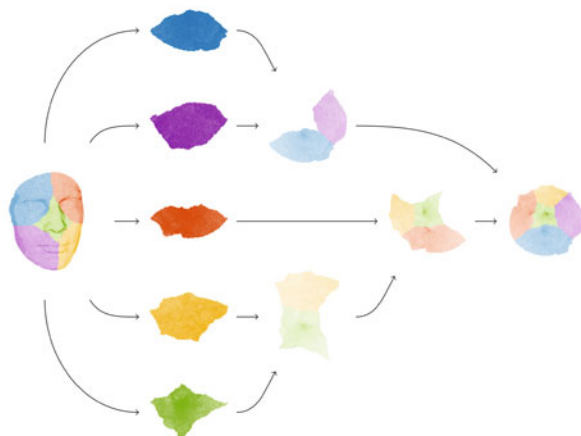


Fig. 14 An illustration of the partial welding procedure. (Image adapted from Choi et al. 2020a)

the boundaries to enforce the consistency between them. This is achieved using a variant of conformal welding called *partial welding*.

More specifically, given a diffeomorphism f from a closed curve (e.g., the unit circle) to itself, conformal welding aims to find two Jordan domains $D, \Omega \subset \bar{\mathbb{C}}$ and two conformal maps $\phi : D \rightarrow \Omega$ and $\phi^* : D^* \rightarrow \Omega^*$, where D^* and Ω^* are the exterior of D and Ω , respectively, such that $\phi = \phi^* \circ f$ on the closed curve. In other words, the two surfaces are stitched together seamlessly. By the sewing theorem (Lehto 1973), if f is a quasisymmetric function from the real axis to itself, then the upper and lower half-planes can be mapped conformally onto disjoint Jordan domains D, Ω by two maps ϕ, ϕ^* , with $\phi(x) = \phi^*(f(x))$ for all $x \in \mathbb{R}$. Partial welding is a variant of conformal welding in the sense that it does not assume the full correspondence between two boundary curves but only the correspondence between a portion of the two curves. As illustrated in Fig. 14, to enforce the consistency between two arcs of the boundaries of two Jordan regions A and B on the complex plane, one can apply a series of analytic functions to map A to the upper half plane and B to the lower half plane such that the two corresponding arcs are mapped to the same interval I on the real axis. Then, one can

find a conformal map that matches the corresponding points on the two arcs, thereby enforcing the consistency between them. After transforming all the boundaries of the flattened subdomains using this idea of partial welding, one can solve the Laplace equation subject to the welded boundary constraints for each subdomain. The final result is then a global free-boundary conformal parameterization of the input mesh. It is noteworthy that both the initial and final parameterizations of the subdomains are independent of those of the other subdomains, and hence one can exploit parallelization in the computational procedure. Some additional steps can be further incorporated for producing disk conformal parameterizations. It is also possible to further reduce the area distortion of the conformal parameterizations by finding an optimal Möbius transformation.

For some applications, it is more desirable to compute conformal parameterizations of the given surfaces onto a standardized planar domain different from a disk or a rectangle. For instance, 3D carotid artery surfaces are usually visualized with the aid of a nonconvex L-shaped parameter domain. In 2017, Choi et al. developed a conformal parameterization method for flattening carotid artery surface meshes. The method starts by computing an arclength scaling map onto a nonconvex L-shaped planar domain for the initialization. Next, it computes the Beltrami coefficient of the inverse of the arclength scaling map and then constructs a quasi-conformal map from the L-shaped domain onto itself with the same Beltrami coefficient using the LBS method (Lui et al. 2013), thereby yielding a conformal flattening map by the composition formula in Equation (12). However, since the L-shaped domain is nonconvex, the overall mapping is not guaranteed to be bijective especially near the nonconvex corner of the domain. To enforce the bijectivity, the method considers smoothing and chopping the Beltrami coefficient iteratively. More specifically, the smoothing step is done by solving the following energy minimization problem:

$$\tilde{\mu} = \operatorname{argmin}_{\mu} \int (|\nabla\mu|^2 + |\mu - \nu| + |\mu|^2), \quad (39)$$

where ν is the current Beltrami coefficient and $\tilde{\mu}$ is the smoothed Beltrami coefficient. The chopping step is done by changing the norm of the Beltrami coefficient from $|\tilde{\mu}|$ to $\min\{|\tilde{\mu}|, 1 - \epsilon\}$ where ϵ is a small positive number. One can then reconstruct a quasi-conformal map from $\tilde{\mu}$ using the LBS method (Lui et al. 2013) and repeat the above steps until the resulting map becomes bijective. Figure 15 shows an example of the conformal parameterization of a carotid artery surface obtained by Choi et al. (2017), from which it can be observed that the parameterization facilitates the visualization of the vessel-wall-plus-plaque thickness (VWT) measurement for the carotid model.

Quasi-conformal Parameterization

The LBS method (Lui et al. 2013) and the BHF method (Lui et al. 2012) can be naturally applied for computing quasi-conformal parameterizations of any given simply connected open triangle mesh. Specifically, after parameterizing the given mesh onto a planar domain using the abovementioned conformal parameterization methods, one can compute a quasi-conformal map with a prescribed Beltrami

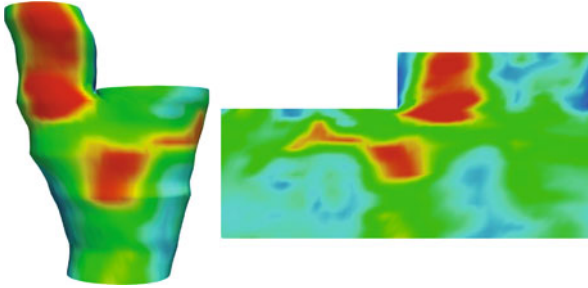


Fig. 15 The conformal parameterization of a carotid artery surface onto a standardized L-shaped planar domain. (Image adapted from Choi et al. 2017). Here, the color represents the vessel-wall-plus-plaque thickness (VWT) measurement for the carotid model

coefficient subject to some boundary constraints using either LBS or BHF. Similarly, the QC iteration method (Lui et al. 2014) can be used for computing landmark-matching Teichmüller parameterization of simply connected open triangle meshes. It is noteworthy that these approaches can only produce fixed-boundary quasi-conformal parameterizations.

More recently, Qiu et al. (2019) proposed a method for computing free-boundary quasi-conformal parameterization of simply connected open triangle meshes. Let $f(z) = f(x + iy) = u(x, y) + iv(x, y)$ and $\mu = \rho + i\tau$. The least squares quasi-conformal energy is defined as follows:

$$E_{\text{LSQC}}(u, v; \mu) = \frac{1}{2} \int_{\Omega} \|P\nabla u + JP\nabla v\|^2 dx dy, \tag{40}$$

where:

$$P = \frac{1}{\sqrt{1 - |\mu|^2}} \begin{pmatrix} 1 - \rho & -\tau \\ -\tau & 1 + \rho \end{pmatrix} \tag{41}$$

and:

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{42}$$

It has been shown in Qiu et al. (2019) that:

$$\begin{aligned} E_{\text{LSQC}}(u, v; \mu) &= \frac{1}{2} \int_{\Omega} \|A^{1/2}u\|^2 dx dy + \frac{1}{2} \int_{\Omega} \|A^{1/2}v\|^2 dx dy \\ &\quad - \int_{\Omega} (u_y v_x - u_x v_y) dx dy, \end{aligned} \tag{43}$$

where:

$$A = \begin{pmatrix} \frac{(\rho-1)^2 + \tau^2}{1-\rho^2-\tau^2} & -\frac{2\tau}{1-\rho^2-\tau^2} \\ -\frac{2\tau}{1-\rho^2-\tau^2} & \frac{1+2\rho+\rho^2+\tau^2}{1-\rho^2-\tau^2} \end{pmatrix} \tag{44}$$

Based on this observation, the computation of a free-boundary quasi-conformal parameterization can be done in a similar manner as in the least squares conformal mapping method (Lévy et al. 2002; Desbrun et al. 2002).

Multiply Connected Open Triangle Meshes

Conformal Parameterization

In Choi et al. (2021), Choi developed a method for the annulus conformal parameterization of multiply connected open triangle meshes with one hole and a method for the poly-annulus conformal parameterization of multiply connected open triangle meshes with $k > 1$ holes.

An illustration of the annulus conformal map (ACM) method is shown in Fig. 16. Given any multiply connected open triangle mesh, the ACM method starts by finding a path from a vertex at the inner boundary to a vertex at the outer boundary and slicing the mesh along the path. As the sliced mesh is simply connected, one can map it onto a rectangle using the rectangular conformal parameterization method in Meng et al. (2016) with a periodic boundary constraint at the top and bottom boundaries (the method will be explained in detail later in section “Point Cloud Parameterization Using Conformal and Quasi-conformal Geometry”). Now, denote the rectangular domain as $[0, L] \times [0, 1]$. One can apply the following exponential map η to map the rectangular domain to an annulus with inner radius $e^{-2\pi L}$ and outer radius 1:

$$\eta(z) = e^{2\pi(z-L)}. \tag{45}$$

Because of the periodic boundary constraint in the computation of the rectangular parameterization, the top and bottom boundaries of the rectangular domain will be

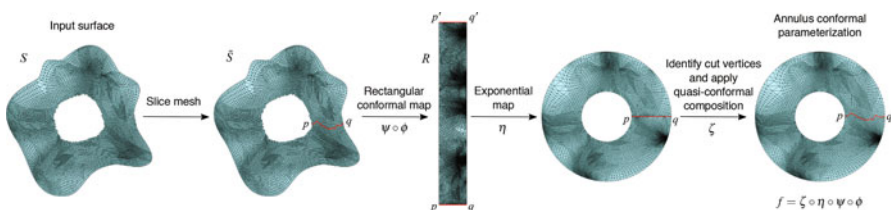


Fig. 16 An illustration of the annulus conformal map (ACM) method. (Image adapted from Choi et al. 2021)

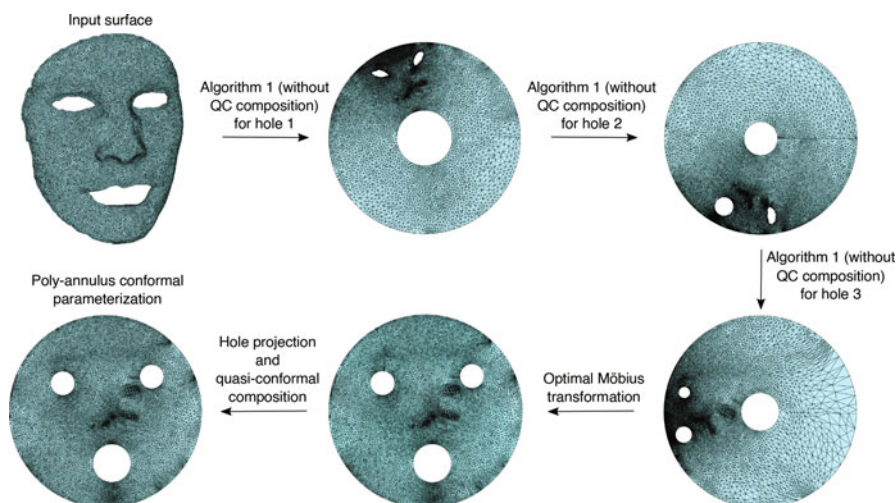


Fig. 17 An illustration of the poly-annulus conformal map (PACM) method. (Image adapted from Choi et al. 2021)

mapped to consistent positions in the annulus. Therefore, it is possible to identify the cut vertices to obtain a parameterization with annulus topology. Finally, one can apply the idea of quasi-conformal composition in Equation (12) to further reduce the conformal distortion of the parameterization caused by the cut and obtain the final annulus conformal parameterization.

Given any multiply connected open triangle mesh with $k > 1$ holes, the poly-annulus conformal map (PACM) method can be used for computing a conformal parameterization of it onto a circle domain with k circular holes (see Fig. 17 for an illustration). The PACM method starts by filling all but one holes of the input mesh and computing an initial parameterization onto an annulus, thereby making the unfilled hole circular. It then removes all filled regions and repeats the above procedure with another hole chosen to be unfilled. Under the series of annulus parameterizations, all holes eventually become highly circular in the parameter domain. Finally, the method performs a projection to further enforce the circularity of all holes and then applies the quasi-conformal composition as in Equation (12) to produce a poly-annulus conformal parameterization.

Quasi-conformal Parameterization

Given any multiply connected open surface and any target Beltrami coefficient, it is natural to ask whether one can compute a quasi-conformal parameterization of the surface onto a canonical circle domain with the Beltrami coefficient of the resulting mapping matching the input Beltrami coefficient. One major challenge in this problem is that the radii and centers of the inner circles on the circle domain

depend on the input multiply connected surface and hence cannot be set arbitrarily. As the LBS method (Lui et al. 2013) and the BHF method (Lui et al. 2012) require fixed (Dirichlet) boundary conditions, they cannot be used for computing the quasi-conformal parameterization with the desired Beltrami coefficient directly. To solve this problem, Ho and Lui (2016) proposed a variational approach called QCMC for computing the quasi-conformal parameterization of multiply connected open surfaces. More specifically, given any multiply connected open triangle mesh \mathcal{M} with $\partial M = \gamma_0 - \gamma_1 - \gamma_2 - \dots - \gamma_k$, i.e., γ_0 is the outer boundary and $\gamma_1, \dots, \gamma_k$ are the inner boundaries, and any Beltrami coefficient μ , the QCMC method treats the radii \mathbf{r} and centers \mathbf{c} of the inner circles on the circle domain as variables and minimizes the following energy to solve for an optimal quasi-conformal map f :

$$E(f, \mathbf{r}, \mathbf{c}) = \int_{\mathcal{M}} |f_{\bar{z}} - \mu f_z|^2, \quad (46)$$

subject to the constraints $f(\gamma_0) = \partial \mathbb{D}$, $f(\gamma_i) = \partial \mathcal{B}_{r_i}(c_i)$ for $i = 1, \dots, k$ and $\|\mu(f)\|_{\infty} = \|f_{\bar{z}}/f_z\|_{\infty} < 1$. Here, $\mathcal{B}_{r_i}(c_i)$ denotes the circle centered at a point $c_i \in \mathbb{Z}$ with radius $r_i > 0$. In other words, the QCMC method simultaneously searches for the optimal conformal module (\mathbf{r}, \mathbf{c}) for the boundary constraints and the optimal quasi-conformal map f that satisfies the boundary constraints and is associated with the prescribed Beltrami coefficient. Figure 18 shows an example of the quasi-conformal parameterization obtained by the QCMC method.

It is also possible to compute the Teichmüller parameterizations of multiply connected open triangle meshes. In 2014, Ng et al. developed a method for computing the extremal Teichmüller map between two multiply connected domains. The method iteratively updates the Beltrami coefficient of the mapping using BHF until the norm of the Beltrami coefficient becomes uniform (see Fig. 19 for an example). By combining the conformal parameterization methods for multiply connected open surfaces and the proposed extremal Teichmüller mapping method, the Teichmüller parameterization of any multiply connected open triangle mesh can be obtained.

Point Cloud Parameterization Using Conformal and Quasi-conformal Geometry

In recent years, several methods have been proposed for computing the conformal and quasi-conformal parameterization of point clouds. Many of these methods are motivated by prior mesh parameterization approaches, with some key modifications and extensions for handling point clouds. Table 2 gives an overview of the recent works. Below, we introduce the works for the parameterization of genus-0 point clouds and then the works for point clouds with disk topology.

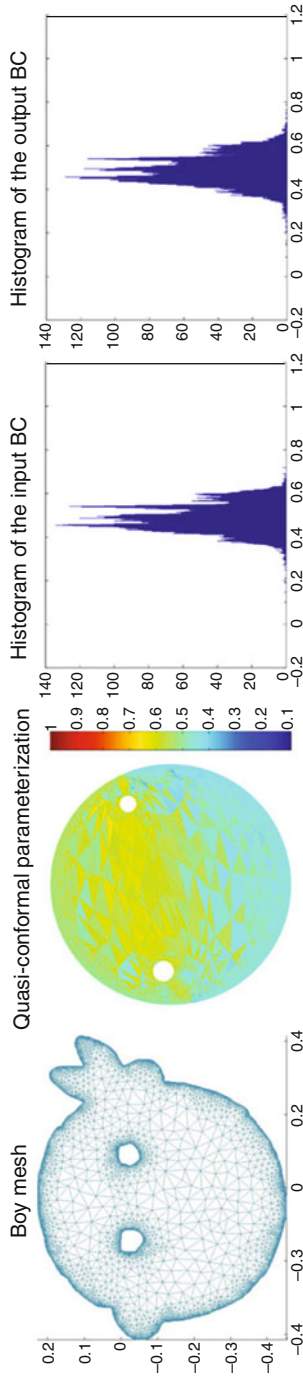


Fig. 18 An example of the QCMC method for the quasi-conformal parameterization of multiply connected open surfaces. (Image adapted from Ho and Lui 2016). Left: The input multiply connected open triangle mesh and the output quasi-conformal parameterization color-coded by the norm of the Beltrami coefficient of the output map. Right: The histograms of the norm of the prescribed Beltrami coefficient and that of the output map

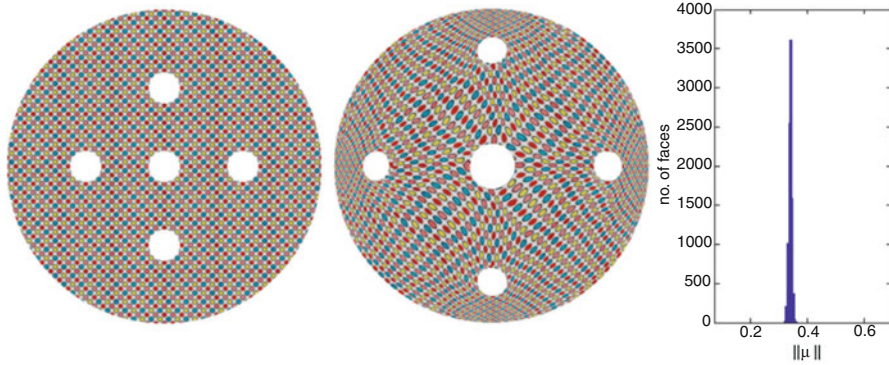


Fig. 19 An example of the extremal Teichmüller map between two multiply connected domains. (Image adapted from Ng et al. 2014). Left: A multiply connected domain with a circle packing texture. Middle: The extremal Teichmüller map onto another multiply connected domain. Note that the small circles are mapped to small ellipses with uniform eccentricity. Right: The histogram of the norm of the Beltrami coefficient of the resulting map

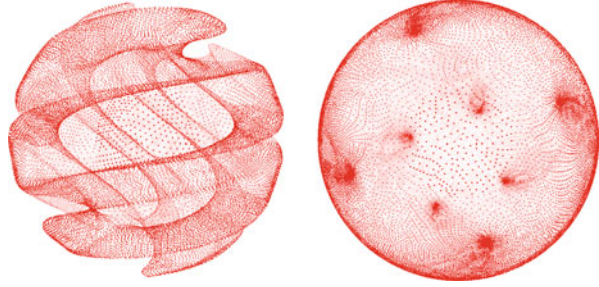
Table 2 A summary of recent conformal and quasi-conformal parameterization methods for point clouds

Method	Surface type	Target domain	Criterion
Spherical map (Choi et al. 2016)	Topological sphere	Sphere	Conformal
TEMPO (Meng et al. 2016)	Topological disk	Rectangle	Conformal/Teichmüller
PCQC (Meng and Lui 2018)	Topological disk	Rectangle	Quasi-conformal
Free-boundary map (Choi et al. 2022)	Topological disk	Free-boundary	Conformal

Genus-0 Point Clouds

For the parameterization of genus-0 point clouds, Choi *et al.* developed a spherical conformal parameterization method in Choi et al. (2016). Analogous to the spherical conformal mapping algorithm for triangle meshes in Choi et al. (2015), the point cloud spherical conformal parameterization method considers a “north pole” step and a “south pole” step. More specifically, the method starts by approximating the Laplacian operator on point clouds using the moving least squares (MLS) method with a Gaussian-type weight function. Using the point cloud Laplacian, one can compute a harmonic flattening map of a genus-0 point cloud and then map it to the sphere using the inverse stereographic projection in Equation (29). This forms the “north pole” step in the proposed method (Choi et al. 2016). As for the “south pole” step, instead of solving for a quasi-conformal map as described in Choi et al. (2015), here the method applies the south pole stereographic projection in Equation (30) and then solves another Laplace equation followed by the inverse south pole

Fig. 20 Spherical conformal parameterization of genus-0 point clouds. (Image adapted from Choi et al. 2016)



stereographic projection. It was shown in Choi et al. (2016) that by performing the “north pole” step and the “south pole” step iteratively, one can eventually obtain a spherical conformal parameterization of the point cloud. In other words, using the north-south reiteration scheme, one can achieve conformality without computing quasi-conformal maps as in the abovementioned mesh parameterization methods. Figure 20 shows an example of the spherical conformal parameterization obtained by Choi et al. (2016). More recently, a variation of the method has been proposed in Jarvis et al. (2021) for the spherical parameterization of sparse genus-0 point clouds.

Point Clouds with Disk Topology

In 2016, Meng et al. proposed a framework called TEMPO for computing Teichmüller extremal mappings of point clouds with disk topology. In particular, they developed methods for computing the rectangular conformal parameterizations and landmark-matching Teichmüller parameterizations of disk-type point clouds (see Fig. 21 for an illustration).

For the rectangular conformal parameterization, the method starts by computing a harmonic map $\phi_0 : \mathcal{P} \rightarrow \mathbb{D}$ of the input disk-type point cloud \mathcal{P} onto the unit disk by solving the Laplace equation:

$$\Delta\phi_0 = 0 \tag{47}$$

subject to a circular boundary constraint. It then computes a map $\phi_1 : \mathbb{D} \rightarrow [0, 1]^2$ from the unit disk to the unit square by solving the generalized Laplace equation (19). Now, let $\phi_1(x, y) = u(x, y) + iv(x, y)$. To achieve conformality, the method considers rescaling the height of the square by a factor h such that the Beltrami coefficient of the map $\phi_2(x, y) = u(x, y) + ihv(x, y)$ is the same as $\mu(\phi_0^{-1})$. The optimal h is obtained by solving the following minimization problem:

$$h = \operatorname{argmin}_{\mathbb{D}} \int_{\mathbb{D}} |\mu(\phi_2) - \mu(\phi_0^{-1})|^2. \tag{48}$$

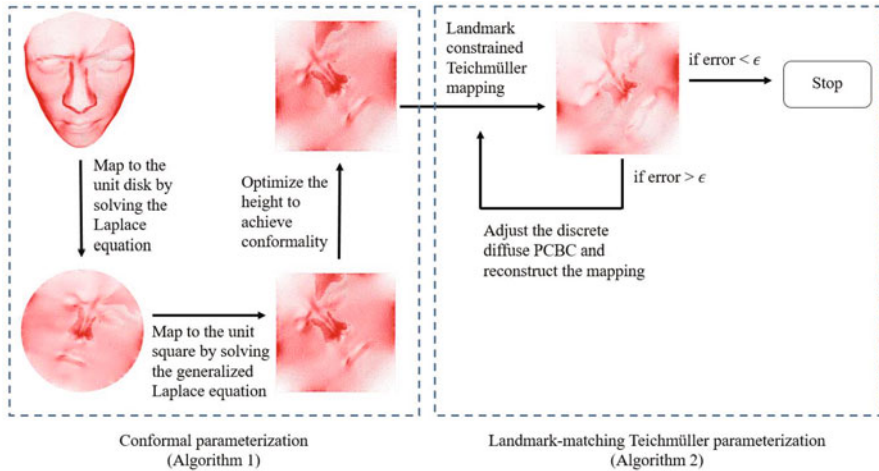


Fig. 21 The computation of rectangular conformal parameterization and landmark-matching Teichmüller parameterizations of point clouds with disk topology. (Image adapted from Meng et al. 2016)

By the composition formula (12), the composition map $\phi_2 \circ \phi_0$ with the optimal h gives a rectangular conformal parameterization of the input point cloud. After getting the rectangular conformal parameterization, the landmark-matching Teichmüller parameterization can be obtained by extending the QC iteration method (Lui et al. 2014) for point clouds. Using the TEMPO framework, it is possible to compute landmark-matching registrations of point cloud surfaces. Figure 22 shows an example of registering two facial point clouds with prescribed landmark constraints.

One important component in the above framework is the approximation of the Beltrami coefficient μ on point clouds. In 2018, Meng and Lui presented a rigorous treatment of the approximation of quasi-conformal maps and the relevant concepts on point clouds. In particular, they proposed a geometric quantity called the *point cloud Beltrami coefficient* (PCBC) and proved that it can effectively capture the local geometric distortion of a point cloud mapping. Using the PCBC, they developed the point cloud quasi-conformal (PCQC) parameterization method for the parameterization of point clouds with any prescribed PCBC (see Fig. 23 for an example).

More recently, Liu et al. developed a free-boundary conformal parameterization method for disk-type point clouds (Choi et al. 2022) by extending the mesh-based DNCP algorithm in Desbrun et al. (2002). The method approximates the Laplacian operator on disk-type point clouds using a modified local mesh method with some special treatments at the point cloud boundary. More specifically, let \mathcal{P} be the given point cloud with n vertices. For each vertex v_i , the method considers its k -nearest neighbors and computes the local Delaunay triangulation to obtain a one-ring neighborhood R_i . The angles in R_i are then used for constructing an $n \times n$ matrix $L_{k,i}^{PC}$:

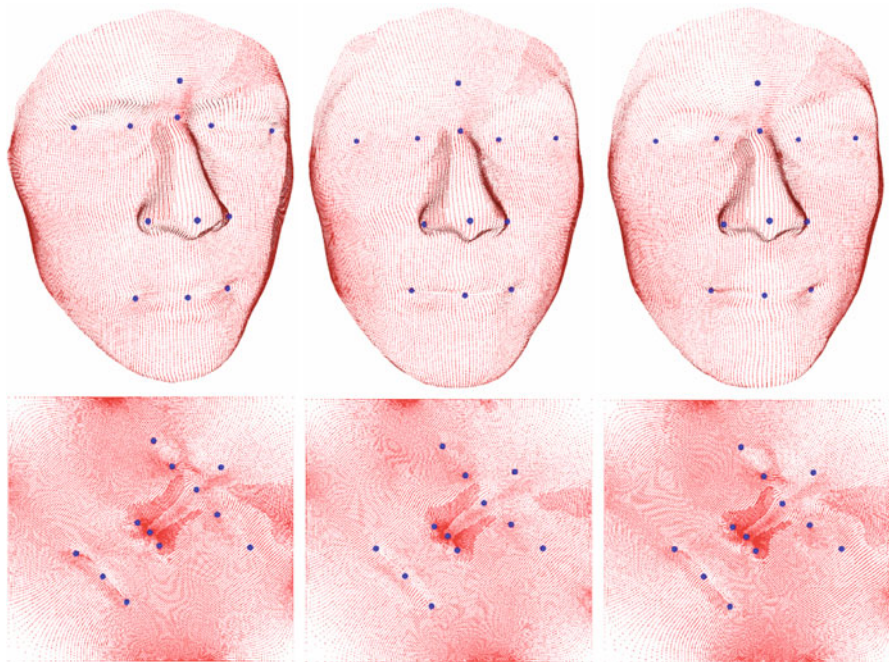


Fig. 22 An illustration of the TEMPO framework. (Image adapted from Meng et al. 2016). Left column: The source human facial point cloud and the rectangular conformal parameterization. Middle column: The target human facial point cloud and the rectangular conformal parameterization. Right column: The registration result and the corresponding landmark-matching Teichmüller mapping of the rectangular domain

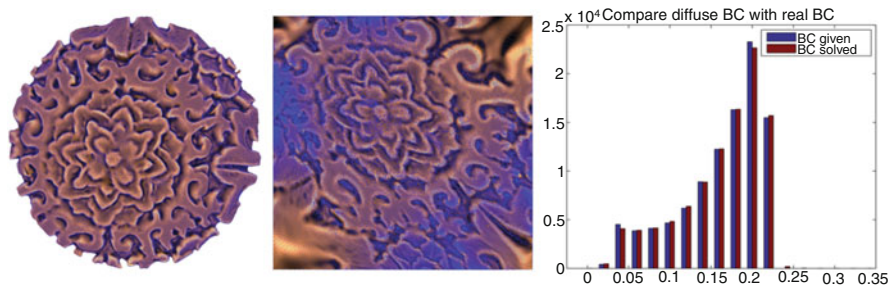


Fig. 23 An example of the point cloud quasi-conformal (PCQC) parameterization. (Image adapted from Meng and Lui 2018). Left: The input point cloud and its underlying surface. Middle column: The PCQC parameterization with the prescribed PCBC. Right: The histogram of the norm of the PCBC of parameterization result and that of the actual PCBC prescribed

$$\begin{cases} L_{k,i}^{pc}(i, j) = L_{k,i}^{pc}(j, i) = -\frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}) & \text{if } v_j \in R_i, \\ L_{k,i}^{pc}(i, i) = \frac{1}{2} \sum_{j: v_j \in R_i} (\cot \alpha_{ij} + \cot \beta_{ij}), \end{cases} \quad (49)$$

where α_{ij} and β_{ij} are the angles opposite to the edge $[v_i, v_j]$ in the local triangulation, and all other entries of $L_{k,i}^{pc}$ are set to be 0. Noticing that the above approximation may be inaccurate at the boundary vertices in case the point cloud boundary shape is nonconvex, the method further checks if every boundary angle θ in the local triangulation for boundary vertices satisfies the angle criterion $c_1 < \theta < c_2$, where (c_1, c_2) is a prescribed angle range. It then removes all triangles that violate this angle criterion and obtains the matrices $L_{k,i}^{pc}$ for the boundary vertices. The Laplacian operator L_k^{pc} for the entire point cloud can then be approximated by $L_k^{pc} = \frac{1}{3} \sum_{i=1}^n L_{k,i}^{pc}$. Finally, the point cloud parameterization $f = (f_x, f_y)$ can be obtained by solving the following linear system:

$$\left(\begin{pmatrix} L_k^{pc} & 0 \\ 0 & L_k^{pc} \end{pmatrix} - \begin{pmatrix} 0 & M_1 \\ M_2 & 0 \end{pmatrix} \right) \begin{pmatrix} f_x \\ f_y \end{pmatrix} = 0, \quad (50)$$

where $M_1(i, j) = M_2(j, i) = \frac{1}{2}$ and $M_1(j, i) = M_2(i, j) = -\frac{1}{2}$ if v_i, v_j are adjacent boundary points with positive orientation and 0 otherwise. As for the boundary conditions, the farthest two points in \mathcal{P} are mapped to $(0, 0)$ and $(1, 0)$ following the original DNCP formulation (Desbrun et al. 2002). Moreover, it has been shown in Choi et al. (2022) that the partial welding method for triangle meshes in Choi et al. (2020a) can be extended for point cloud parameterization (see Fig. 24). More specifically, the proposed point cloud parameterization method partitions the point cloud into several subdomains and flattens the boundary of each of them onto the plane. It then applies the partial welding method to enforce the consistency of the boundaries. Finally, the interior part of each subdomain can be mapped onto the plane by solving the Laplace equation with the welded boundary constraints.

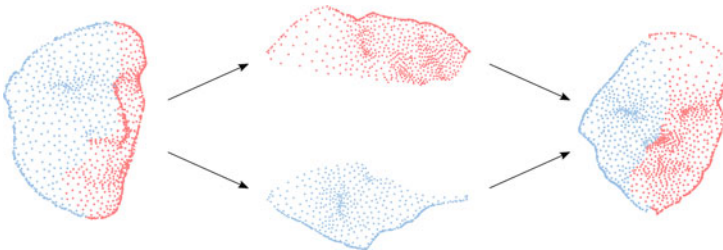


Fig. 24 An illustration of the free-boundary conformal parameterization method for disk-type point clouds via partial welding. (Image adapted from Choi et al. 2022)

Applications

The surface mapping and parameterization methods developed based on quasi-conformal geometry have been found useful in many practical applications in recent years.

For instance, the mapping methods have been applied to biological and medical shape analysis. In Zeng et al. (2010) and Zeng and Yang (2014), Zeng et al. applied quasi-conformal mappings for supine and prone colon registration. In 2015, Wen et al. used landmark-matching quasi-conformal mappings for analyzing vestibular systems. In 2015, Lam et al. used Teichmüller mappings for skull registration. In 2015, Choi et al. used the FLASH method for registering brain cortical surfaces (see Fig. 25). In Chan et al. (2016, 2020), Chan et al. utilized conformal and quasi-conformal mappings for the shape analysis of hippocampal surfaces. The spherical

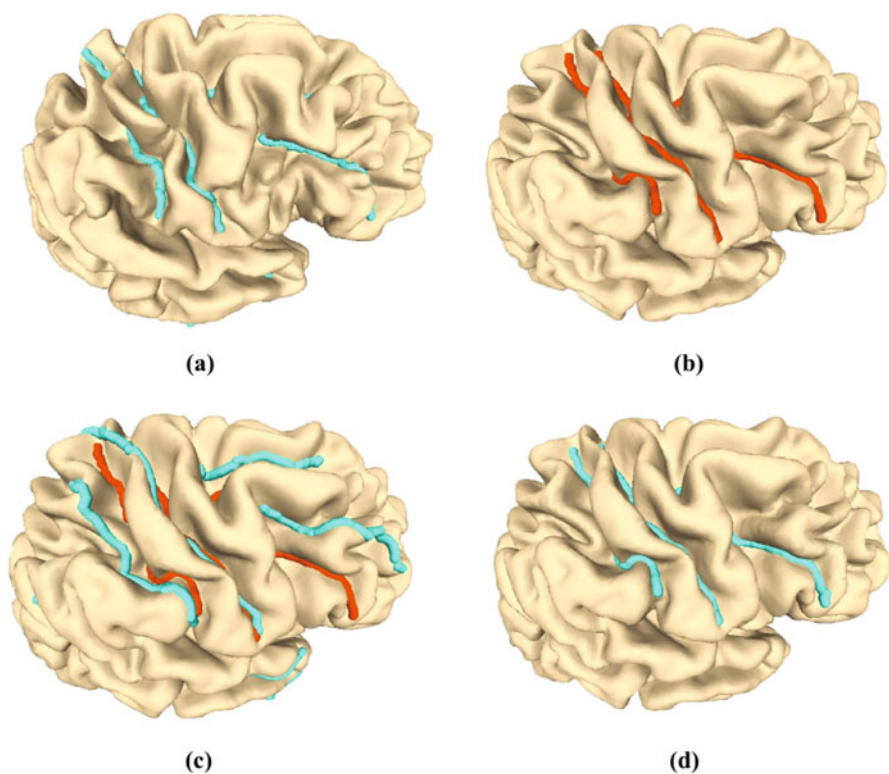


Fig. 25 Registering brain cortical surfaces using the FLASH method. (Image adapted from Choi et al. 2015). (a) The source brain with sulcal landmarks. (b) The target brain with sulcal landmarks. (c) The registration obtained using conformal parameterization without landmark constraints. (d) The registration obtained using landmark-constrained optimized conformal parameterization. It can be observed that the landmark-constrained parameterization gives a more accurate registration result

conformal parameterization method developed in Choi et al. (2015) has been applied to optical mapping for cardiac electrophysiology (Christoph et al. 2017) and cardiac radiofrequency catheter ablation (Zhou et al. 2016). In 2018, Choi and Mahadevan utilized Teichmüller mappings for insect wing morphometry (see Fig. 26). In Choi et al. (2020c,d), Choi et al. utilized conformal parameterizations and Teichmüller mappings for analyzing human and other mammalian tooth shape (see Fig. 27).

The mapping methods have also been applied to different engineering problems. For instance, the spherical conformal parameterization method in Choi et al. (2015) has been applied to collaborative robotics (Popov and Klimchik 2019). The disk conformal parameterization method in Choi and Lui (2015) has been applied to

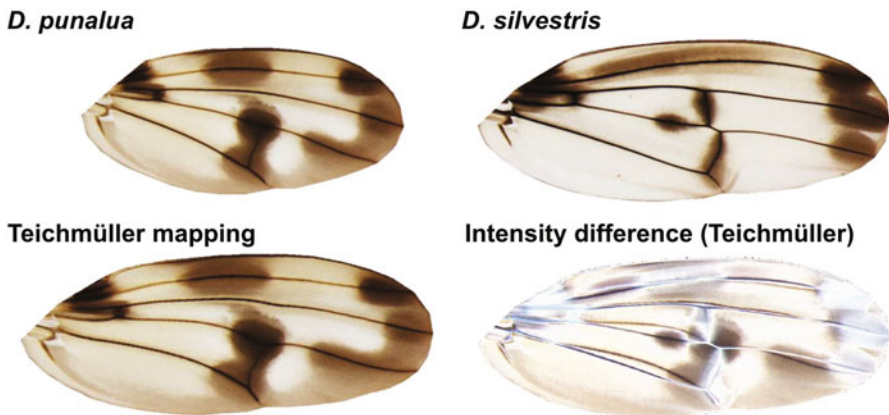


Fig. 26 Insect wing morphometry using landmark-matching Teichmüller mappings. (Image adapted from Choi and Mahadevan 2018). To quantify the difference between two different *Drosophila* wing shapes (top row), one can compute a landmark-matching Teichmüller mapping (bottom left) from the first wing to the second wing that matches the prominent structural features of the two wings such as the intersections of the veins. It is then possible to compare the Teichmüller mapping result and the second wing by considering their intensity difference

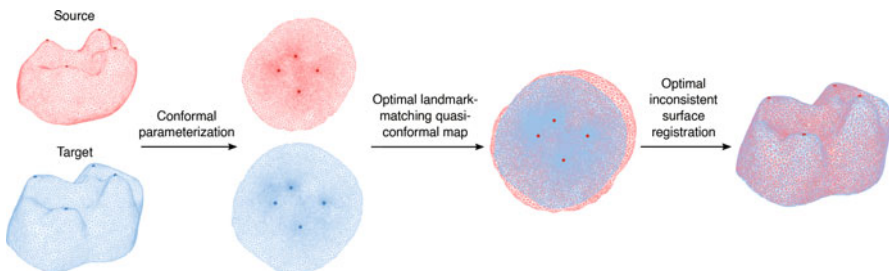


Fig. 27 Mammalian tooth morphometry using quasi-conformal mappings. (Image adapted from Choi et al. 2020c). Given two tooth surfaces, the method first computes a free-boundary conformal parameterization of each surface. It then finds an optimal landmark-matching quasi-conformal map on the plane, which finally gives an optimal inconsistent surface registration

structural optimization (Kusssmaul et al. 2019) and robot navigation (Notomista and Saveriano 2021). The rectangular parameterization method in Meng et al. (2016) has been applied to T-spline surface reconstruction (Wang 2021) and nanotechnology (Guralnik 2021). In 2017, Choi et al. developed a method for subdivision connectivity surface remeshing via Teichmüller mappings. In 2018, Yung et al. developed an efficient image registration method using coarse triangulations and landmark-matching quasi-conformal mappings. In (2019, 2021), Choi et al. utilized conformal and quasi-conformal mapping methods (Meng et al. 2016; Choi and Lui 2018) in developing constrained optimization frameworks for kirigami metamaterial design. In 2021, Shaqfa et al. extended the disk conformal parameterization method (Choi and Lui 2015) for spherical cap parameterization and utilized it for analyzing stone microstructures. Recently, Jarvis et al. (2021) developed a method for reconstructing 3D asteroid and comet shapes from sparse feature point sets via spherical parameterizations based on the method in Choi et al. (2016).

Conclusion

With the theoretical guarantee and computational efficiency of quasi-conformal maps, many conformal and quasi-conformal parameterization methods have been developed for triangle meshes and point clouds. The methods have been successfully applied to various science and engineering problems.

More recently, there is an increasing interest in volumetric mapping methods for the deformations of 3D solid shapes (Lee et al. 2016; Yueh et al. 2019; Choi and Rycroft 2021; Zhang et al. 2022). Therefore, a natural future research direction is the development of higher-dimensional parameterization methods using higher-dimensional quasi-conformal theory.

Acknowledgments This work was supported in part by the National Science Foundation under Grant No. DMS-2002103 (to Gary P. T. Choi) and HKRGC GRF under project ID 2130549 (to Lok Ming Lui).

References

- Ahlfors, L.V.: Lectures on Quasiconformal Mappings, vol. 38. American Mathematical Society, Providence (2006)
- Belkin, M., Sun, J., Wang, Y.: Constructing Laplace operator from point clouds in \mathbb{R}^d . In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1031–1040 (2009)
- Ben-Chen, M., Gotsman, C., Bunin, G.: Conformal flattening by curvature prescription and metric scaling. *Comput. Graph. Forum* **27**(2), 449–458 (2008)
- Chan, H.L., Li, H., Lui, L.M.: Quasi-conformal statistical shape analysis of hippocampal surfaces for Alzheimer’s disease analysis. *Neurocomputing* **175**, 177–187 (2016)
- Chan, H.L., Yam, T.C., Lui, L.M.: Automatic characteristic-calibrated registration (ACC-REG): hippocampal surface registration using eigen-graphs. *Pattern Recogn.* **103**, 107142 (2020)

- Chien, E., Levi, Z., Weber, O.: Bounded distortion parametrization in the space of metrics. *ACM Trans. Graph.* **35**(6), 1–16 (2016)
- Choi, P.T.: Surface conformal/quasi-conformal parameterization with applications. In: CUHK Electronic Theses and Dissertations Collection, The Chinese University of Hong Kong (2016)
- Choi, P.T., Lui, L.M.: Fast disk conformal parameterization of simply-connected open surfaces. *J. Sci. Comput.* **65**(3), 1065–1090 (2015)
- Choi, G.P.-T., Lui, L.M.: A linear formulation for disk conformal parameterization of simply-connected open surfaces. *Adv. Comput. Math.* **44**(1), 87–114 (2018)
- Choi, G.P.T., Mahadevan, L.: Planar morphometrics using Teichmüller maps. *Proc. R. Soc. A* **474**(2217), 20170905 (2018)
- Choi, G.P.T., Rycroft, C.H.: Density-equalizing maps for simply connected open surfaces. *SIAM J. Imaging Sci.* **11**(2), 1134–1178 (2018)
- Choi, G.P.T., Rycroft, C.H.: Volumetric density-equalizing reference map with applications. *J. Sci. Comput.* **86**(3), 41 (2021)
- Choi, P.T., Lam, K.C., Lui, L.M.: FLASH: fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM J. Imaging Sci.* **8**(1), 67–94 (2015)
- Choi, G.P.-T., Ho, K.T., Lui, L.M.: Spherical conformal parameterization of genus-0 point clouds for meshing. *SIAM J. Imaging Sci.* **9**(4), 1582–1618 (2016)
- Choi, G.P.-T., Man, M.H.-Y., Lui, L.M.: Fast spherical quasiconformal parameterization of genus-0 closed surfaces with application to adaptive remeshing. *Geom. Imaging Comput.* **3**(1–2), 1–29 (2016)
- Choi, G.P.T., Chen, Y., Lui, L.M., Chiu, B.: Conformal mapping of carotid vessel wall and plaque thickness measured from 3D ultrasound images. *Med. Biol. Eng. Comput.* **55**(12), 2183–2195 (2017)
- Choi, C.P., Gu, X., Lui, L.M.: Subdivision connectivity remeshing via Teichmüller extremal map. *Inverse Probl. Imaging* **11**(5), 825–855 (2017)
- Choi, G.P.T., Dudte, L.H., Mahadevan, L.: Programming shape using kirigami tessellations. *Nat. Mater.* **18**(9), 999–1004 (2019)
- Choi, G.P.T., Leung-Liu, Y., Gu, X., Lui, L.M.: Parallelizable global conformal parameterization of simply-connected surfaces via partial welding. *SIAM J. Imaging Sci.* **13**(3), 1049–1083 (2020a)
- Choi, G.P.T., Chiu, B., Rycroft, C.H.: Area-preserving mapping of 3D carotid ultrasound images using density-equalizing reference map. *IEEE Trans. Biomed. Eng.* **67**(9), 1507–1517 (2020b)
- Choi, G.P.T., Qiu, D., Lui, L.M.: Shape analysis via inconsistent surface registration. *Proc. R. Soc. A* **476**(2242), 20200147 (2020c)
- Choi, G.P.T., Chan, H.L., Yong, R., Ranjitkar, S., Brook, A., Townsend, G., Chen, K., Lui, L.M.: Tooth morphometry using quasi-conformal theory. *Pattern Recogn.* **99**, 107064 (2020d)
- Choi, G.P.T.: Efficient conformal parameterization of multiply-connected surfaces using quasi-conformal theory. *J. Sci. Comput.* **87**(3), 70 (2021)
- Choi, G.P.T., Giri, A., Kumar, L.: Adaptive area-preserving parameterization of open and closed anatomical surfaces. *Comput. Biol. Med.*, **148**, 105715 (2022)
- Choi, G.P.T., Dudte, L.H., Mahadevan, L.: Compact reconfigurable kirigami. *Phys. Rev. Res.* **3**(4), 043030 (2021)
- Choi, G.P.T., Liu, Y., Lui, L.M.: Free-boundary conformal parameterization of point clouds. *J. Sci. Comput.* **90**(1), 14 (2022)
- Christoph, J., Schröder-Schetelig, J., Luther, S.: Electromechanical optical mapping. *Prog. Biophys. Mol. Biol.* **130**, 150–169 (2017)
- Claici, S., Bessmeltsev, M., Schaefer, S., Solomon, J.: Isometry-aware preconditioning for mesh parameterization. *Comput. Graph. Forum* **36**(5), 37–47 (2017)
- Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum* **21**(3), 209–218 (2002)
- Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: *Advances in Multiresolution for Geometric Modelling*, pp. 157–186. Springer, Berlin/New York (2005)
- Fu, X.-M., Liu, Y., Guo, B.: Computing locally injective mappings by advanced MIPS. *ACM Trans. Graph.* **34**(4), 1–12 (2015)

- Gardiner, F.P., Lakic, N.: *Quasiconformal Teichmüller Theory*, vol. 76. American Mathematical Society, Providence (2000)
- Giri, A., Choi, G.P.T., Kumar, L.: Open and closed anatomical surface description via hemispherical area-preserving map. *Sig. Process.* **180**, 107867 (2021)
- Gu, X.D., Yau, S.-T.: *Computational Conformal Geometry*, vol. 1. International Press, Somerville (2008)
- Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* **23**(8), 949–958 (2004)
- Gu, X., Luo, F., Yau, S.T.: Computational conformal geometry behind modern technologies. *Not. Am. Math. Soc.* **67**(10), 1509–1525 (2020)
- Guralnik, B., Hansen, O., Henrichsen, H.H., Caridad, J.M., Wei, W., Hansen, M.F., Nielsen, P.F., Petersen, D.H.: Effective electrical resistivity in a square array of oriented square inclusions. *Nanotechnology* **32**(18), 185706 (2021)
- Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M.: Conformal surface parameterization for texture mapping. *IEEE Trans. Vis. Comput. Graph.* **6**(2), 181–189 (2000)
- Ho, K.T., Lui, L.M.: QCMC: quasi-conformal parameterizations for multiply-connected domains. *Adv. Comput. Math.* **42**(2), 279–312 (2016)
- Hormann, K., Lévy, B., Sheffer, A.: Mesh parameterization: theory and practice. In: *ACM SIGGRAPH 2007 Courses* (2007)
- Jarvis, B., Choi, G.P.T., Hockman, B., Morrell, B., Bandopadhyay, S., Lubey, D., Villa, J., Bhaskaran, S., Bayard, D., Nesnas, I.A.: 3D shape reconstruction of small bodies from sparse features. *IEEE Robot. Autom. Lett.* **6**(4), 7089–7096 (2021)
- Jin, M., Kim, J., Luo, F., Gu, X.: Discrete surface Ricci flow. *IEEE Trans. Vis. Comput. Graph.* **14**(5), 1030–1043 (2008)
- Wang, J., Leach, R., Chen, R., Xu, J., Jiang, X.J.: Distortion-free intelligent sampling of sparse surfaces via locally refined T-spline metamodelling. *Int. J. Precis. Eng. Manuf. – Green Technol.* **8**(5), 1471–1486 (2021)
- Kussmaul, R., Jónasson, J.G., Zogg, M., Ermanni, P.: A novel computational framework for structural optimization with patched laminates. *Struct. Multidiscipl. Optim.* **60**(5), 2073–2091 (2019)
- Lai, R., Liang, J., Zhao, H.-K.: A local mesh method for solving PDEs on point clouds. *Inverse Probl. Imaging* **7**(3), 737–755 (2013)
- Lai, R., Wen, Z., Yin, W., Gu, X., Lui, L.M.: Folding-free global conformal mapping for genus-0 surfaces by harmonic energy minimization. *J. Sci. Comput.* **58**(3), 705–725 (2014)
- Lam, K.C., Lui, L.M.: Landmark-and intensity-based registration with large deformations via quasi-conformal maps. *SIAM J. Imaging Sci.* **7**(4), 2364–2392 (2014)
- Lam, K.C., Gu, X., Lui, L.M.: Landmark constrained genus-one surface Teichmüller map applied to surface registration in medical imaging. *Med. Image Anal.* **25**(1), 45–55 (2015)
- Lee, Y.T., Lam, K.C., Lui, L.M.: Landmark-matching transformation with large deformation via n -dimensional quasi-conformal maps. *J. Sci. Comput.* **67**(3), 926–954 (2016)
- Lehto, O.: *Quasiconformal Mappings in the Plane*, vol. 126. Springer, Berlin/Heidelberg (1973)
- Lei, N., Gu, X.: FFT-OT: a fast algorithm for optimal transportation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6280–6289 (2021)
- Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* **21**(3), 362–371 (2002)
- Liang, J., Zhao, H.: Solving partial differential equations on point clouds. *SIAM J. Sci. Comput.* **35**(3), A1461–A1486 (2013)
- Liang, J., Lai, R., Wong, T.W., Zhao, H.: Geometric understanding of point clouds using Laplace-Beltrami operator. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 214–221 (2012)
- Li, S., Zeng, W., Zhou, D., Gu, X., Gao, J.: Compact conformal map for greedy routing in wireless mobile sensor networks. *IEEE Trans. Mobile Comput.* **15**(7), 1632–1646 (2015)
- Lipman, Y.: Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph.* **31**(4), 1–13 (2012)

- Lipman, Y., Kim, V.G., Funkhouser, T.A.: Simple formulas for quasiconformal plane deformations. *ACM Trans. Graph.* **31**(5), 1–13 (2012)
- Liu, L., Ye, C., Ni, R., Fu, X.-M.: Progressive parameterizations. *ACM Trans. Graph.* **37**(4), 1–12 (2018)
- Lui, L.M., Wen, C.: Geometric registration of high-genus surfaces. *SIAM J. Imaging Sci.* **7**(1), 337–365 (2014)
- Lui, L.M., Wong, T.W., Thompson, P., Chan, T., Gu, X., Yau, S.-T.: Shape-based diffeomorphic registration on hippocampal surfaces using Beltrami holomorphic flow. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 323–330. Springer, (2010)
- Lui, L.M., Wong, T.W., Zeng, W., Gu, X., Thompson, P.M., Chan, T.F., Yau, S.-T.: Optimization of surface registrations using Beltrami holomorphic flow. *J. Sci. Comput.* **50**(3), 557–585 (2012)
- Lui, L.M., Lam, K.C., Wong, T.W., Gu, X.: Texture map and video compression using Beltrami representation. *SIAM J. Imaging Sci.* **6**(4), 1880–1902 (2013)
- Lui, L.M., Lam, K.C., Yau, S.-T., Gu, X.: Teichmüller mapping (T-map) and its applications to landmark matching registration. *SIAM J. Imaging Sci.* **7**(1), 391–426 (2014)
- Lui, L.M., Gu, X., Yau, S.-T.: Convergence of an iterative algorithm for Teichmüller maps via harmonic energy optimization. *Math. Comput.* **84**(296), 2823–2842 (2015)
- Meng, T., Lui, L.M.: PCBC: quasiconformality of point cloud mappings. *J. Sci. Comput.* **77**(1), 597–633 (2018)
- Meng, Q., Li, B., Holstein, H., Liu, Y.: Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBF networks. *Pattern Recogn.* **46**(8), 2361–2375 (2013)
- Meng, T.W., Choi, G.P.-T., Lui, L.M.: TEMPO: feature-endowed Teichmüller extremal mappings of point clouds. *SIAM J. Imaging Sci.* **9**(4), 1922–1962 (2016)
- Nadeem, S., Su, Z., Zeng, W., Kaufman, A., Gu, X.: Spherical parameterization balancing angle and area distortions. *IEEE Trans. Vis. Comput. Graph.* **23**(6), 1663–1676 (2016)
- Ng, T.C., Gu, X., Lui, L.M.: Computing extremal Teichmüller map of multiply-connected domains via Beltrami holomorphic flow. *J. Sci. Comput.* **60**(2), 249–275 (2014)
- Notomista, G., Saveriano, M.: Safety of dynamical systems with multiple non-convex unsafe sets using control barrier functions. *IEEE Control Syst. Lett.* **6**, 1136–1141 (2021)
- Popov, D., Klimchik, A.: Real-time external contact force estimation and localization for collaborative robot. In: *2019 IEEE International Conference on Mechatronics*, vol. 1, pp. 646–651. IEEE (2019)
- Pumarola, A., Sanchez-Riera, J., Choi, G.P.T., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: modeling the geometry of dressed humans. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2242–2251 (2019)
- Qiu, D., Lam, K.-C., Lui, L.-M.: Computing quasi-conformal folds. *SIAM J. Imaging Sci.* **12**(3), 1392–1424 (2019)
- Rabinovich, M., Poranne, R., Panozzo, D., Sorkine-Hornung, O.: Scalable locally injective mappings. *ACM Trans. Graph.* **36**(4), 1 (2017)
- Sawhney, R., Crane, K.: Boundary first flattening. *ACM Trans. Graph.* **37**(1), 1–14 (2017)
- Shaqfa, M., Choi, G.P.T., Beyer, K.: Spherical cap harmonic analysis (SCHA) for characterising the morphology of rough surface patches. *Powder Technol.* **393**, 837–856 (2021)
- Sharp, N., Crane, K.: A Laplacian for nonmanifold triangle meshes. *Comput. Graph. Forum* **39**(5), 69–80 (2020)
- Sheffer, A., Praun, E., Rose, K.: Mesh parameterization methods and their applications. *Found. Trends® Comput. Graph. Vis.* **2**(2), 105–171 (2006)
- Smith, J., Schaefer, S.: Bijective parameterization with free boundaries. *ACM Trans. Graph.* **34**(4), 1–9 (2015)
- Su, K., Cui, L., Qian, K., Lei, N., Zhang, J., Zhang, M., Gu, X.D.: Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. *Comput. Aided Geom. Des.* **46**, 76–91 (2016)
- Su, J.-P., Ye, C., Liu, L., Fu, X.-M.: Efficient bijective parameterizations. *ACM Trans. Graph.* **39**(4), 111–1 (2020)

- Ta, D., Tu, Y., Lu, Z.-L., Wang, Y.: Quantitative characterization of the human retinotopic map based on quasiconformal mapping. *Med. Image Anal.* **75**, 102230 (2021)
- Tewari, G., Gotsman, C., Gortler, S.J.: Meshing genus-1 point clouds using discrete one-forms. *Comput. Graph.* **30**(6), 917–926 (2006)
- Tu, Y., Ta, D., Gu, X.D., Lu, Z.-L., Wang, Y.: Diffeomorphic registration for retinotopic mapping via quasiconformal mapping. In: 2020 IEEE 17th International Symposium on Biomedical Imaging, pp. 687–691. IEEE (2020)
- Vogiatzis, P., Ma, M., Chen, S., Gu, X.D.: Computational design and additive manufacturing of periodic conformal metasurfaces by synthesizing topology optimization with conformal mapping. *Comput. Methods Appl. Mech. Eng.* **328**, 477–497 (2018)
- Weber, O., Myles, A., Zorin, D.: Computing extremal quasiconformal maps. *Comput. Graph. Forum* **31**(5), 1679–1689 (2012)
- Wen, C., Wang, D., Shi, L., Chu, W.C.W., Cheng, J.C.Y., Lui, L.M.: Landmark constrained registration of high-genus surfaces applied to vestibular system morphometry. *Comput. Med. Imaging Graph.* **44**, 1–12 (2015)
- Wong, T.W., Zhao, H.-K.: Computation of quasi-conformal surface maps using discrete Beltrami flow. *SIAM J. Imaging Sci.* **7**(4), 2675–2699 (2014)
- Wong, T.W., Zhao, H.-K.: Computing surface uniformization using discrete Beltrami flow. *SIAM J. Sci. Comput.* **37**(3), A1342–A1364 (2015)
- Yang, Y.-J., Zeng, W.: Quasiconformal rectilinear map. *Graph. Models* **107**, 101057 (2020)
- Yang, Y.-L., Guo, R., Luo, F., Hu, S.-M., Gu, X.: Generalized discrete Ricci flow. *Comput. Graph. Forum* **28**(7), 2005–2014 (2009)
- Yin, X., Dai, J., Yau, S.-T., Gu, X.: Slit map: conformal parameterization for multiply connected surfaces. In: International Conference on Geometric Modeling and Processing, pp. 410–422. Springer (2008)
- Yueh, M.-H., Lin, W.-W., Wu, C.-T., Yau, S.-T.: An efficient energy minimization for conformal parameterizations. *J. Sci. Comput.* **73**(1), 203–227 (2017)
- Yueh, M.-H., Lin, W.-W., Wu, C.-T., Yau, S.-T.: A novel stretch energy minimization algorithm for equiareal parameterizations. *J. Sci. Comput.* **78**(3), 1353–1386 (2019)
- Yueh, M.-H., Li, T., Lin, W.-W., Yau, S.-T.: A novel algorithm for volume-preserving parameterizations of 3-manifolds. *SIAM J. Imaging Sci.* **12**(2), 1071–1098 (2019)
- Yueh, M.-H., Huang, H.-H., Li, T., Lin, W.-W., Yau, S.-T.: Optimized surface parameterizations with applications to chinese virtual broadcasting. *Electron. Trans. Numer. Anal.* **53**, 383–405 (2020)
- Yung, C.P., Choi, G.P.T., Chen, K., Lui, L.M.: Efficient feature-based image registration by mapping sparsified surfaces. *J. Vis. Commun. Image Represent.* **55**, 561–571 (2018)
- Zeng, W., Gu, X.D.: Registration for 3D surfaces with large deformations using quasi-conformal curvature flow. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2457–2464. IEEE (2011)
- Zeng, W., Yang, Y.-J.: Colon flattening by landmark-driven optimal quasiconformal mapping. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 244–251. Springer (2014)
- Zeng, W., Yin, X., Zhang, M., Luo, F., Gu, X.: Generalized Koebe’s method for conformal mapping multiply connected domains. In: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, pp. 89–100 (2009)
- Zeng, W., Marino, J., Gurijala, K.C., Gu, X., Kaufman, A.: Supine and prone colon registration using quasi-conformal mapping. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1348–1357 (2010)
- Zeng, W., Lui, L.M., Luo, F., Chan, T.F.-C., Yau, S.-T., Gu, D.X.: Computing quasiconformal maps using an auxiliary metric and discrete curvature flow. *Numer. Math.* **121**(4), 671–703 (2012)
- Zhang, L., Liu, L., Gotsman, C., Huang, H.: Mesh reconstruction by meshless denoising and parameterization. *Comput. Graph.* **34**(3), 198–208 (2010)
- Zhang, M., Guo, R., Zeng, W., Luo, F., Yau, S.-T., Gu, X.: The unified discrete surface Ricci flow. *Graph. Models* **76**(5), 321–339 (2014)

- Zhang, M., Zeng, W., Guo, R., Luo, F., Gu, X.D.: Survey on discrete surface Ricci flow. *J. Comput. Sci. Technol.* **30**(3), 598–613 (2015)
- Zhang, D., Choi, G.P.T., Zhang, J., Lui, L.M.: A unifying framework for n -dimensional quasi-conformal mappings. *SIAM J. Imaging Sci.* **15**(2), 960–988 (2022)
- Zhao, X., Su, Z., Gu, X.D., Kaufman, A., Sun, J., Gao, J., Luo, F.: Area-preservation mapping using optimal mass transport. *IEEE Trans. Vis. Comput. Graph.* **19**(12), 2838–2847 (2013)
- Zhao, J., Qi, X., Wen, C., Lei, N., Gu, X.: Automatic and robust skull registration based on discrete uniformization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 431–440 (2019)
- Zhou, X.-Y., Ernst, S., Lee, S.-L.: Path planning for robot-enhanced cardiac radiofrequency catheter ablation. In: *2016 IEEE International Conference on Robotics and Automation*, pp. 4172–4177. IEEE (2016)
- Zou, G., Hu, J., Gu, X., Hua, J.: Authalic parameterization of general surfaces using Lie advection. *IEEE Trans. Vis. Comput. Graph.* **17**(12), 2005–2014 (2011)
- Zwicker, M., Gotsman, C.: Meshing point clouds using spherical parameterization. In: *Proceedings of the Eurographics Symposium on Point-Based Graphics*, pp. 173–180 (2004)



Recent Geometric Flows in Multi-orientation Image Processing via a Cartan Connection

44

R. Duits, B. M. N. Smets, A. J. Wemmenhove, J. W. Portegies, and
E. J. Bekkers

Contents

Introduction	1527
Scores on Lie Groups $G = \mathbb{R}^d \times T$ and the Motivation for Left-Invariant Processing and a Left-Invariant Connection on $T(G)$	1530
Motivation: Choosing a Cartan Connection for Geometric (PDE-Based) Image Processing via Scores	1533
Structure and Contributions of the Article	1534
A Parameterized Class of Cartan Connections and Their Duals	1536
Expressing the Lie-Cartan Connection (and Its Dual) in Left-Invariant Coordinates	1541
(Partial) Lie-Cartan Connections for (Sub)-Riemannian Geometry	1543
The Special Case of Interest $\nu = 1$ and Hamiltonian Flows for the Riemannian Geodesic Problem on G	1544
The Homogeneous Space \mathbb{M}_d of Positions and Orientations	1549
The Metric Models on \mathbb{M}_d : Shortest Curves and Spheres	1550
Straight Curve Fits	1554
Exponential Curve Fits of the Second Order Are Found by SVD of the Hessian	1557
Overview of Image Analysis Applications for $G = SE(d)$	1559
Shortest Curve Application: Tracking of Blood Vessels	1560
Straight Curve Application: Biomarkers for Diabetes	1563

R. Duits (✉) · B. Smets · J. Wemmenhove
Applied Differential Geometry, Department of Mathematics and Computer Science, Eindhoven
University of Technology, Eindhoven, Netherlands
e-mail: R.Duits@tue.nl; B.M.N.Smets@tue.nl; a.j.wemmenhove@tue.nl

E. Bekkers
Amsterdam Machine Learning Lab, University of Amsterdam, Amsterdam, Netherlands
e-mail: E.J.Bekkers@uva.nl

J. Portegies
Center for Analysis, Scientific Computing and Applications, Department of Mathematics and
Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands
e-mail: J.W.Portegies@tue.nl

Straight Curve Application: PDEs on \mathbb{M}_2 for Denoising	1565
Conclusion	1571
Appendix A: Hamiltonian Flow of the Left-Invariant (Sub-)Riemannian Geodesic Problem on Lie Group G	1573
Appendix B: Left-Invariant Vector Fields on $SE(3)$ via Two Charts	1576
Appendix C: Proofs of Results on Lie-Cartan Connections	1577
Proof of Lemma 2	1577
Proof of Lemma 3	1578
References	1579

Abstract

Applications of geometric flows to multi-orientation image processing require the choice of an (affine) connection on the Lie group G of roto-translations. Typical choices of such connections are called the $(-)$, (0) and $(+)$ connection. As the construction of these connections in standard references is quite involved, we provide an overview. We show that these connections are members of a larger, one-parameter class of connections, and we motivate that the $(+)$ connection is most suited for our image analysis applications. The class $\nabla^{[\nu]}$, with $\nu \in \mathbb{R}$, is given by $\nabla_X^{[\nu]} Y = \nu[X, Y]$ for all left-invariant vector fields X, Y on G . Their auto-parallel curves are the exponential curves. Their torsion is $T[X, Y] = (2\nu - 1)[X, Y]$, and the $(-)$, (0) and $(+)$ connections arise for $\nu = 0, \frac{1}{2}, 1$.

We propose the case $\nu = 1$, as then the Hamiltonian flows on $T^*(G)$ for Riemannian distance minimizers on G (induced by left-invariant metric tensor field \mathcal{G}) reduce to $\nabla_{\dot{\gamma}}^{[1]} \lambda = 0$ and $\dot{\gamma} = \mathcal{G}_{|\dot{\gamma}}^{-1} \lambda$, where $\dot{\gamma}$ is velocity and λ is momentum. So now ‘shortest curves’ have parallel momentum, whereas ‘straight curves’ have auto-parallel velocity. We also extend this idea to sub-Riemannian geometry via a partial connection.

The connection underlies PDE flows for crossing-preserving geodesic wave-front propagation and denoising in multi-orientation image processing, where we use:

1. The ‘shortest curves’ for tracking in multi-orientation image representations,
2. The ‘straight curve fits’ for locally adaptive frames in PDEs for crossing-preserving image denoising and enhancement.

Keywords

Cartan connections · Multi-orientation image processing · Riemannian geometry · sub-Riemannian geometry · Geometric control · Geodesic tracking · Medical image processing

Introduction

The synergy between the mathematical fields of partial differential equations, geometric control, Lie group analysis, harmonic analysis, variational methods and the applied fields of image analysis, numerical analysis, neurogeometry and neuroimaging is increasing rapidly and has attracted many researchers. An emerging field for interaction between these fields is multi-orientation image analysis, where image data is lifted to the space of positions and orientations. Typically such lifted data is concentrated around lifted curves; see Fig. 1.

There exist many ways to construct such orientation lifts of image data. For example, it can be done linearly by means of convolving the image by rotated and translated Gabor wavelets (where image reconstruction requires integration over scale and orientation) (Citti and Sarti 2006; Baspinar 2018) or by proper

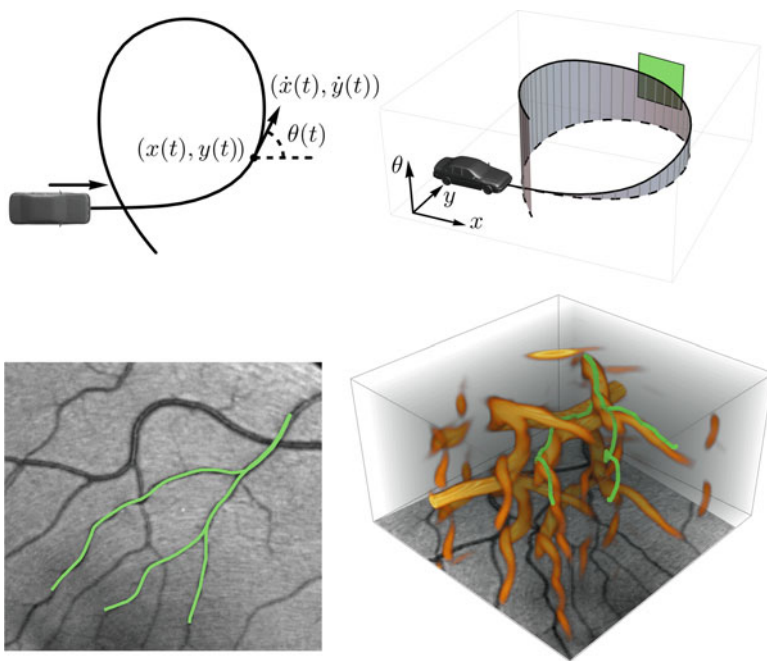


Fig. 1 Top: *Lifted paths* $\gamma(t) = (x(t), y(t), \theta(t))$ in $\mathbb{R}^2 \times S^1$ (left) where the tangent $\dot{\gamma}(t)$ is restricted to the span of $(\cos \theta(t), \sin \theta(t), 0)$ and $(0, 0, 1)$, of which the green plane on the right is an example. Bottom: *Lifted image data* depicted by an orange volume rendering. The meaning of shortest path between points in an image is determined by a combination of a cost computed from the lifted data, the restriction above and a curvature penalization. The path optimization problem is formulated on the position-orientation domain such as in the image on the right. The cost for moving through the orange parts is lower than elsewhere

wavelets, including cake wavelets (where inversion requires integration over angles only) (Duits et al. 2007; Bekkers 2017), or nonlinearly via orientation channel representations (Forssen 2004; Felsberg et al. 2006). In the differential geometry article, we constrain ourselves to invertible orientation scores (Duits and Franken 2010a) constructed by cake wavelets following standard settings as explained in Bekkers (2017).

In multi-orientation processing on orientation scores (Janssen et al. 2018; Duits et al. 2007, 2019; Zhang et al. 2016) (or on other orientation lifts Felsberg 2012; Citti and Sarti 2006; Duits and Franken 2011; Citti et al. 2016; Momayyez-Siahkhal and Siddiqi 2009), differential geometry plays a fundamental role in PDE- and ODE-based techniques for pattern recognition, cortical modelling and image analysis. Image processing applications are then provided with fundamental differential geometrical tools such as Cartan connections (Piuze et al. 2015; Duits et al. 2016) that ‘literally connect’ all tangent spaces in the tangent bundle $T(\mathbb{M}_d)$ above the space \mathbb{M}_d of positions and orientations. Such a connection underlies flows Duits and Franken (2011), segmentations (Zhang et al. 2016), detection (Bekkers et al. 2015) and tracking (Duits et al. 2018) on \mathbb{M}_d . In all of these PDE-based processing techniques on \mathbb{M}_d , one has the major benefit (over related algorithms acting directly in the image domain \mathbb{R}^d) that the processing generically deals with complex structures (such as crossings, bifurcations, etc.). In this article, we will highlight some applications in the experimental section, to illustrate how our preferred Cartan connection enters image analysis applications.

Here the key idea is that elongated structures that are involved in crossings are manifestly disentangled in orientation lifts of image data; see Fig. 2. This allows for crossing-preserving enhancements and tracking via such orientation lifts as shown in Fig. 3.

Furthermore, in the space of positions and orientations it is possible to check for alignment of local orientations in the image data. Filtering well-aligned local features in multi-orientation distributions (e.g. orientation scores) of image data is sometimes called ‘contextual image processing’ (Prčkovska et al. 2015; Bekkers 2017; Franken 2008). It relates to cortical models for line perception in human vision (Petitot 2003; Bosking et al. 1997; Citti and Sarti 2006) and is highly beneficial for data enhancement and denoising in image analysis applications; see, for example, (Duits et al. 2019; Chambolle and Pock 2018; Citti et al. 2016; Duits and Franken 2011; Momayyez-Siahkhal and Siddiqi 2009; Franken and Duits 2009; Portegies et al. 2015), prior to geometric tracking (Meesters et al. 2017; Portegies et al. 2015; Chen and Cohen 2018; Duits et al. 2018) in the homogeneous space of positions and orientations.

The homogeneous space of positions is formally defined as a Lie group quotient:

$$\mathbb{M}_d = G/H = SE(d)/(\{\mathbf{0}\} \times SO(d-1)) \quad (1)$$

in the Lie group $SE(d)$ of roto-translations on \mathbb{R}^d . We shall be concerned with applications of crossing-preserving denoising, analysis and tracking of line structures (blood vessels) via orientation scores, as depicted in Fig. 3. In the application

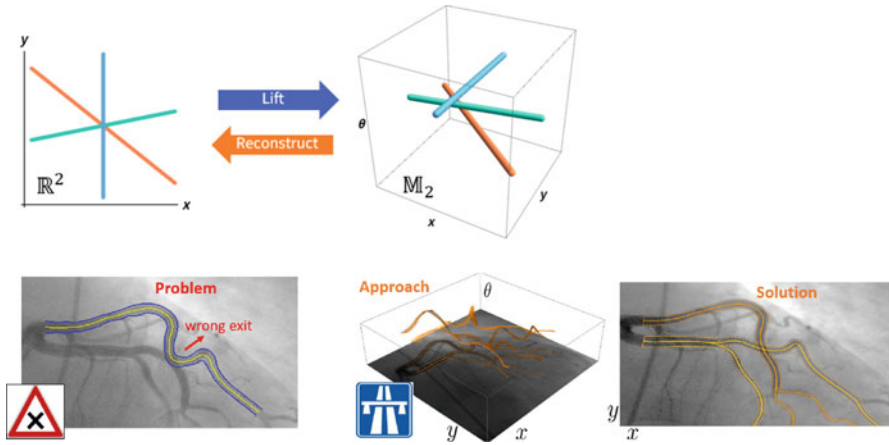


Fig. 2 Current tracking algorithms on images often fail (left); therefore, we first extend the image domain to the space of positions and orientations (where no such crossings occur) and then apply geodesic tracking (right), enhancement and learning to automatically deal with complex structures

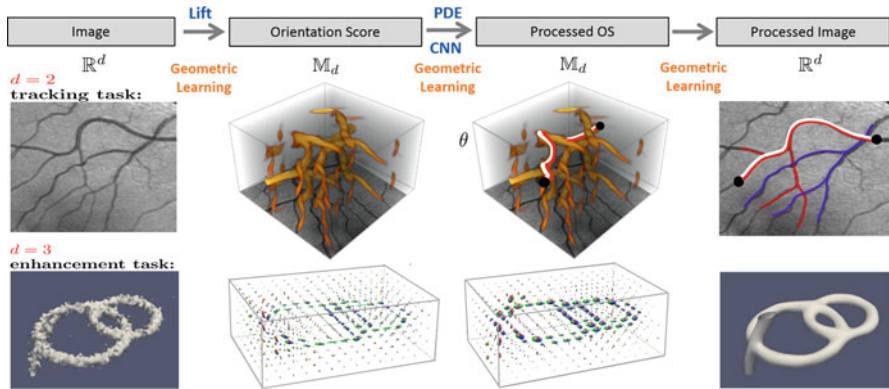


Fig. 3 Top: instead of direct processing of an image, we process via an invertible orientation score, obtained by convolving the image with a set of rotated wavelets (Duits et al. 2007; Bekkers 2017; Janssen et al. 2014). Second row: vessel tracking in a 2D image via orientation scores (Bekkers 2017; Bekkers et al. 2015; Duits et al. 2018). Third row: crossing-preserving diffusion via the orientation score of a 3D image (Janssen et al. 2014; Duits et al. 2016). For automation, one can integrate geometric deep learning via PDE-based G-CNNs (Smets et al. 2020) and G-CNNs (Bekkers et al. 2018; Cohen and Welling 2016). Here we will not elaborate on such machine learning techniques but rather focus on the underlying PDEs and Cartan connection

section of this work, we mainly focus on the case $d = 2$, but we also highlight related works and applications where the case $d = 3$ is tackled.

Remark 1. The multi-orientation analysis of images is much simpler for $d = 2$ as then subgroup the H consists only of the unity element and therefore the

Lie $SE(2)$ group of rotations and translations in the plane is isomorphic to the three-dimensional homogeneous space \mathbb{M}_2 of positions and orientations. In case $d = 3$, the subgroup $H \equiv SO(2)$ and therefore the homogeneous space \mathbb{M}_3 of positions and orientations is five dimensional. In that case, a multi-orientation distribution $U : \mathbb{M}_3 \rightarrow \mathbb{C}$ can be visualized by a field of angular profiles on a grid: $\{\mathbf{x} + \frac{|U(\mathbf{x}, \mathbf{n})|}{2\|U\|_{\infty}(\mathbb{M}_3)} \mathbf{n} \mid \mathbf{n} \in S^2, \mathbf{x} \in \mathbb{Z}^3\}$ with colour-coded orientations. For such a visualization, see the bottom row of Fig. 3.

Remark 2. The idea of crossing-preserving denoising and tracking via multi-feature representations of images also generalizes to other ‘scores’ (multi-feature representations) on other Lie groups G (and Lie group quotients G/H):

- Image processing of multi-frequency representations (Gabor transforms) (Duits et al. 2013) defined on the Heisenberg group $H(2d + 1)$
- Image processing of multi-velocity distributions (velocity scores) (Barbieri et al. 2014) defined on the Heisenberg group $H(2d + 1)$
- Image processing of spherical image data (Mashtakov et al. 2017) defined on a quotient $S^2 \equiv SO(3)/SO(2)$ the rotation group $SO(3)$
- Image processing of multi-orientation and scale scores (continuous wavelet transforms) (Sharma and Duits 2015) on the similitude group $SIM(d)$

In this article, we shall not be concerned with applications of the other Lie group cases mentioned in the remark above, but in order to keep generality of our theoretical results, we will initially study Cartan connections on Lie groups G in general, so that our results also apply to the general Lie group setting.

Furthermore, we deliberately avoid technical issues (Duits et al. 2013, 2019; Smets et al. 2019, 2020) that come along with taking Lie group quotients like in (1). The differential geometrical results in this article are easier to grasp if one just considers the whole Lie group G . For integration of the appropriate symmetries that come along with taking Lie group quotients, with in particular the one of primary interest (1), see Duits et al. (2013, 2019), Smets et al. (2019, 2020).

Scores on Lie Groups $G = \mathbb{R}^d \rtimes T$ and the Motivation for Left-Invariant Processing and a Left-Invariant Connection on $T(G)$

In the general Lie group setting, we consider Lie groups $G = \mathbb{R}^d \rtimes T$ that are the semi-direct product of \mathbb{R}^d with another Lie group T (reflecting the feature of interest, e.g. orientations, velocities, frequencies, scales, etc.). Then one uses a unitary representation $g \mapsto \mathcal{U}_g$ of such a Lie group onto the space of images modelled by $\mathbb{L}_2(\mathbb{R}^d)$ to construct the ‘score’ (or ‘lifted image’) by probing image f by a family of group coherent wavelets constructed from a wavelet $\psi \in \mathbb{L}_2(\mathbb{R}^d) \cap \mathbb{L}_1(\mathbb{R}^d)$

$$\mathcal{W}_\psi f(g) = (\mathcal{U}_g \psi, f)_{\mathbb{L}_2(\mathbb{R}^d)}.$$

Clearly, not every (square integrable) function on the Lie group is the orientation score of an image. It turns out that such a transform $\mathcal{W}_\psi : \mathbb{L}_2(\mathbb{R}^d) \rightarrow \mathbb{C}_K^G$ is a unitary map onto its range which is the unique reproducing kernel Hilbert space \mathbb{C}_K^G consisting of functions on the Lie group G with reproducing kernel $K(g, h) = (\mathcal{U}_g \psi, \mathcal{U}_h \psi)_{\mathbb{L}_2(\mathbb{R}^d)}$. For details, see Duits (2005), Ali et al. (1999) and Fuehr (2005).

Remark 3. In our special case of interest where the score is an ‘orientation score’, we set Lie group $G = SE(d) = \mathbb{R}^d \rtimes SO(d)$, for $d \in \{2, 3\}$, with group product

$$g_1 g_2 = (\mathbf{x}_1, \mathbf{R}_1)(\mathbf{x}_2, \mathbf{R}_2) = (\mathbf{R}_1 \mathbf{x}_2 + \mathbf{x}_1, \mathbf{R}_1 \mathbf{R}_2), \quad g_i = (\mathbf{x}_i, \mathbf{R}_i) \in SE(d), \tag{2}$$

for $i = 1, 2$. Furthermore, we obtain the group coherent wavelets via the action

$$\mathcal{U}_g \psi(\mathbf{x}) = \psi(\mathbf{R}^{-1}(\mathbf{x} - \mathbf{b})), \tag{3}$$

for all $g = (\mathbf{b}, \mathbf{R}) \in SE(d)$, $\mathbf{x} \in \mathbb{R}^d$. In this case, the family of group coherent wavelets are rotated and translated versions of ψ . For $d = 3$, one must assume that ψ is rotationally symmetric around the reference axis in order to ensure that the orientation score $\mathcal{W}_\psi f$ is well defined on \mathbb{M}_3 . For details, see Janssen et al. (2018).

The reproducing kernel norm coincides with a (constrained) \mathbb{L}_2 -norm if \mathcal{U} is irreducible (Grossmann et al. 1985). This essentially follows by a generalization of Schur’s lemma (The overall idea is that $\tilde{\mathcal{W}}_\psi^* \circ \tilde{\mathcal{W}}_\psi$ commutes with the unitary irreducible representation and is therefore a multiple of the identity. Subtleties arise as it is not obvious that operator $\tilde{\mathcal{W}}_\psi$ as defined below, is bounded, cf. (Grossmann et al. 1985; Dieudonné 1977).

Remark 4. If \mathcal{U} is reducible, which is the case for the representation given by (3), one can apply a decomposition into irreducible subspaces (Duits and Franken 2010a, App.A). Then one either must restrict the space of images (e.g. to the space of ball-limited images Fuehr 2005, ch.5.2, Duits 2005, ch.4.5) or one must rely on distributional wavelet transforms (Bekkers et al. 2014, App.B). In both cases, one must take care that all coherently transformed wavelets $\mathcal{U}_g \psi$ together ‘cover all the frequencies in the Fourier domain’; see Duits (2005), Fuehr (2005) and Duits and Bekkers (2020).

Let us define $\tilde{\mathcal{W}}_\psi : \mathbb{L}_2(\mathbb{R}^d) \rightarrow \mathbb{L}_2(G)$ by $\tilde{\mathcal{W}}_\psi f = \mathcal{W}_\psi f$. We can rely on the following commutative diagram to design operators in the enlarged image domain $G = \mathbb{R}^d \rtimes T$. As a consequence of the following Lemma, processing on scores must be left invariant and not right invariant.

Remark 5. In this article, we will not address the issue of choosing a proper wavelet ψ . For the setting of $G = SE(3)$ or more precisely for $\mathbb{M}_3 = G/H$, we prefer to use so-called cake wavelets to construct invertible orientation scores (Duits 2005). For

quick practical explanations on 2D cake wavelets, see Bekkers et al. (2014); for the same on 3D cake wavelets, see Duits et al. (2016). All experiments in this chapter use cake wavelets ψ with standard parameter settings (Martin and Duits 2017). For detailed educational background on invertible orientation scores, proper wavelets, and cake wavelets, see Duits and Bekkers (2020).

Definition 1. An operator $\Phi : \mathbb{L}_2(G) \rightarrow \mathbb{L}_2(G)$ is *left invariant* iff

$$\Phi[\mathcal{L}_g V] = \mathcal{L}_g[\Phi V], \text{ for all } g \in G, V \in \mathbb{L}_2(G), \tag{4}$$

where the left-regular action \mathcal{L}_g of $g \in G$ onto $\mathbb{L}_2(G)$ is given by

$$\mathcal{L}_g V(q) = V(g^{-1}q) \text{ for almost every } q \in G. \tag{5}$$

Similarly, the right-regular action is given by

$$\mathcal{R}_g V(q) = V(qg), \text{ for all } g, q \in G, V \in \mathbb{L}_2(G),$$

and an operator Φ is right invariant $\Phi[\mathcal{R}_g V] = \mathcal{R}_g[\Phi V]$, for all $g \in G$, and for all $V \in \mathbb{L}_2(G)$.

Lemma 1. Let $\mathcal{U} : G \rightarrow B(\mathbb{L}_2(\mathbb{R}^d))$ be a unitary representation. Let $\Phi : \mathbb{C}_K^G \rightarrow \mathbb{L}_2(G)$ be a bounded operator. Then the corresponding operator Υ_ψ on $\mathbb{L}_2(\mathbb{R}^d)$ given by $\Upsilon_\psi[f] = (\tilde{W}_\psi)^* \circ \Phi \circ \tilde{W}_\psi[f]$ on the images $f \in \mathbb{L}_2(\mathbb{R}^d)$ satisfies

$$\mathcal{U}_g \circ \Upsilon = \Upsilon \circ \mathcal{U}_g \text{ for all } g \in G$$

if and only if the effective operator on the score $\mathbb{P}_\psi \circ \Phi$ is left-invariant, i.e.

$$\mathcal{L}_g(\mathbb{P}_\psi \circ \Phi) = (\mathbb{P}_\psi \circ \Phi)\mathcal{L}_g, \text{ for all } g \in G,$$

which shows that score processing must be left invariant. Moreover, we have

$$\Phi \circ \mathcal{R}_g = \mathcal{R}_g \circ \Phi \Rightarrow \Upsilon_\psi = \Upsilon_{\mathcal{U}_g \psi} \text{ for all } g \in G,$$

which shows that right invariance is a highly undesirable property for score processing.

See Fig. 5 to get a visual impression what the above theorem means for the group of roto-translations in the plane $G = SE(2) \equiv \mathbb{M}_2$; recall Remark 3.

Proof. This Lemma essentially gathers earlier results of the first author Duits (2005, Thm. 21) and Duits et al. (2013, Thm. 1) where the proof can be found.

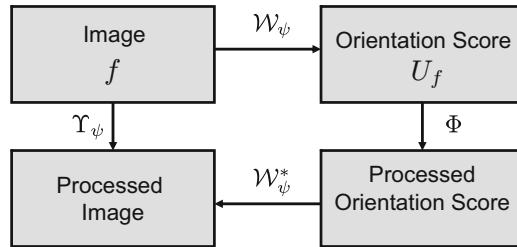


Fig. 4 A schematic view of image processing via scores. According to Lemma 1, Φ must be left invariant and not right invariant. The same applies to the other Lie group cases mentioned in Remark 2, where the score is not an ‘orientation score’ but, for example, a ‘frequency score’ (Duits et al. 2013)

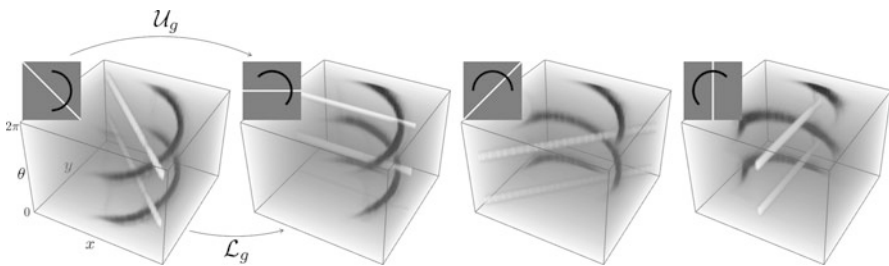


Fig. 5 A roto-translation of the image corresponds to a shift twist of the orientation score, both defined via group representations of $G = SE(2)$ on the image and the orientation score. Shift twist of images and orientation scores are denoted, respectively, by the *left-regular* representations \mathcal{U}_g (3) and \mathcal{L}_g (5). In this illustration of $\mathcal{W}_\psi \circ \mathcal{U}_g = \mathcal{L}_g \circ \mathcal{W}_\psi$, we have set $g = (\mathbf{0}, \theta)$, with θ increasing from left to right

Corollary 1. *We want to apply (second-order) PDE-based operators Φ that involve covariant derivatives based on a connection. Both Φ and the connection should be left invariant, and not right invariant.*

In the sequel, we will therefore study a specific family of left-invariant connections: a parameterized family ($\nu \in \mathbb{R}$) of Lie-Cartan connection, which, as we will see, contains one very special member: the case $\nu = 1$.

Motivation: Choosing a Cartan Connection for Geometric (PDE-Based) Image Processing via Scores

Geometric image processing via scores on Lie Groups requires a choice of underlying Cartan connection on $T(G)$. For geometric image processing, we literally need to ‘connect’ tangent spaces $T_g(G)$ at different base points $g \in G$ in the domain of a score. Such a connection gives rise to (coordinate free) covariant derivatives that

we need in PDE-based image processing via scores on Lie groups. Next, we will illustrate this on two geometric (PDE-based) image processing techniques:

1. Crossing-preserving image enhancement and denoising via scores, via geometric PDEs expressed in left invariant/covariant derivatives
2. Shortest paths (and optimal control) in orientation scores

Crossing-preserving image enhancement and denoising via scores require left-invariant PDEs and data-driven locally adaptive frames ('gauge frames') on G . The PDEs expressed in left-invariant frames include geometric flow along *straight curves* (with *parallel velocity*), which are exponential curves in Cartan connections. The PDEs expressed in gauge frames include geometric flow along 'straight curve fits' that solve a local curve optimization problem where straight-curve fits follow the data at each base point $g \in G$ in a locally optimal way.

Shortest curves are paths that minimize a distance in G . We will show such shortest paths have *parallel momentum* with respect to a specific choice of Cartan connection. This will be the Lie-Cartan connection with $\nu = 1$.

The geometric notions of 'short' and 'straight' for curves will depend on the connection. Recall that such a connection must be left invariant by Corollary 1; moreover, it must account for torsion visible in the score (see Fig. 1).

Structure and Contributions of the Article

In this section, we have so far provided an overview of geometric image processing via scores on Lie groups, with a particular focus on the case where the score is an orientation score defined on the homogeneous space of positions and orientations \mathbb{M}_d as a Lie group quotient in $SE(d)$; recall (1). We also motivated the quest for choosing an effective Cartan connection as we need to 'connect' tangent spaces in PDE flows for (crossing-preserving) enhancement and geodesic tracking in scores.

In section "A Parameterized Class of Cartan Connections and Their Duals", we will study a parameterized class (parameterized $\nu \in \mathbb{R}$) of Cartan connections that we call 'Lie-Cartan' connections that we will employ later on Riemannian geometry and Riemannian geometrical methods in later sections. We also consider partial Lie-Cartan connections to deal with the sub-Riemannian geometry setting. In *sub-Riemannian* geometry, motions on $T(G)$ are constrained to a sub-bundle as other directions carry an 'infinite cost'; this amounts to 'nonholonomic systems' in mechanics (cf. the green tangent plane restriction in Fig. 1).

In section "The Special Case of Interest $\nu = 1$ and Hamiltonian Flows for the Riemannian Geodesic Problem on G ", we show that the Lie-Cartan connection with $\nu = 1$ is the best choice for geometric image processing on scores. We motivate this mainly with our new general result: Theorem 1. Roughly speaking, we show that shortest curves have parallel momentum as straight curves have parallel velocity.

In the remaining sections, we drop the generality and focus on the case where the score is an orientation score defined on the homogeneous space \mathbb{M}_d of positions

and orientations; recall (1). We start in section “[The Homogeneous Space \$\mathbb{M}_d\$ of Positions and Orientations](#)” to outline the details regarding this homogeneous space.

In section “[The Metric Models on \$\mathbb{M}_d\$: Shortest Curves and Spheres](#)”, we study the shortest curves and the induced spheres on \mathbb{M}_d where we put emphasis on the sub-Riemannian setting, with pointers to the literature.

In section “[Straight Curve Fits](#)”, we study the straight curves and data-driven straight curve fits in $SE(d)$ and their projections in \mathbb{M}_d .

Finally, in section “[Overview of Image Analysis Applications for \$G = SE\(d\)\$](#) ”, we consider several image analysis applications: three applications where the shortest curves in \mathbb{M}_d play a central role and four applications where the straight-curve fits in \mathbb{M}_d play a central role:

- In section “[Shortest Curve Application: Tracking of Blood Vessels](#)”, we use shortest curves (geodesics) in \mathbb{M}_d to show that geodesic vessel tracking in \mathbb{M}_d outperforms geodesic tracking in \mathbb{R}^d and that sub-Riemannian geometric tracking outperforms isotropic Riemannian geometric tracking in \mathbb{M}_d . Initially, we consider $d = 2$, but then in section “[Shortest Curve Applications: Geodesic Vessel and Fibre Tracking in \$\mathbb{M}_3\$](#) ” we also address applications (Duits et al. 2018) for $d = 3$ and a new 3D vessel tracking experiment.
- In section “[Straight Curve Application: Biomarkers for Diabetes](#)”, we use straight curve fits in $\mathbb{M}_2 \equiv SE(2)$ for biomarkers of diabetes in retinal images.
- In section “[Straight Curve Application: PDEs on \$\mathbb{M}_2\$ for Denoising](#)”, we use straight curve fits in $\mathbb{M}_2 \equiv SE(2)$ for image denoising. We also address extensions to \mathbb{M}_3 :
 - In section “[Straight Curve Application: PDEs on \$\mathbb{M}_3\$ for Denoising FODFs in DW-MRI](#)”, we briefly highlight applications of the \mathbb{M}_3 -case in enhancing fibre bundles in diffusion-weighted MRI (DW-MRI). For details, see St Onge et al. (2019) and Smets et al. (2019).
 - In section “[Straight Curve Application: PDEs on \$\mathbb{M}_3\$ for Denoising 3D X-Ray Data](#)”, we briefly highlight applications of the \mathbb{M}_3 -case in denoising of 3D X-ray data. For details, see Janssen et al. (2018).

Contributions: This article summarizes results and image analysis applications from previous works on PDE-based image processing via (orientation) scores (Duits and Franken 2010a; Bekkers et al. 2015, 2017; Bekkers 2017; Bekkers et al. 2018; Duits et al. 2013; Duits et al. 2018, 2019; Smets et al. 2019), and more importantly, it puts them in a single novel geometrical perspective via a specific Lie-Cartan connection. Theorem 1 and Theorem 2 contain new general results. Lemma 2, Lemma 3 and Corollary 2 gather (standard) differential geometrical computations that are relevant in our quest of choosing an appropriate Cartan connection on Lie groups for geometric image processing via scores.

On the experimental side, we provide new experiments (e.g. Fig. 14) and illustrations (e.g. Fig. 18) and Tables (Tables 1,2), in addition to our previously published work. However, this is only with the intention of providing a general

overview of the possibilities and impact of the differential geometric theory on many medical image analysis applications.

A Parameterized Class of Cartan Connections and Their Duals

In this section, we will address a parameterized class of Cartan connections on Lie groups that we will call ‘Lie-Cartan’ connections as they are induced by the Lie bracket on the Lie group. We will adhere to references and conventions in the book by Kobayashi and Nomizu (1963) and the recent review article by Cogliati and Mastrolia (2018).

Let G be a Lie group of dimension n . Let $\mathbb{L}_2(G)$ denote the space of square integrable functions on G endowed with the left-invariant Haar measure. Let $T_e(G)$ be the tangent space at unity element e . Let G be a Lie group such that the exponential map $\exp : T_e(G) \rightarrow G$ is surjective. Then $T_e(G)$ is a Lie algebra with Lie bracket

$$\begin{aligned}
 [A, B] &= - \left. \frac{d}{dt} \right|_{t=0} (\gamma^{-B}(\sqrt{t}) \gamma^{-A}(\sqrt{t}) \gamma^B(\sqrt{t}) \gamma^A(\sqrt{t})) \in T_e(G), \\
 &= -\frac{1}{2} \left. \frac{d^2}{dt^2} \right|_{t=0} (\gamma^{-B}(t) \gamma^{-A}(t) \gamma^B(t) \gamma^A(t)),
 \end{aligned}
 \tag{6}$$

where $t \mapsto \gamma^X(t) = e^{tX}$ is a differentiable curve in G with $\gamma^X(0) = e$ and $(\gamma^X)'(0) = X$ for $X = A, B$. For details on Lie brackets, see Kolar et al. (1999). Let the right-regular representation $\mathcal{R} : G \rightarrow BL(\mathbb{L}_2(G))$ be given by $\mathcal{R}_g V(h) = V(hg)$. Then $d\mathcal{R}$ is given by

$$(d\mathcal{R}(A))V(g) = \lim_{t \downarrow 0} \frac{(\mathcal{R}_{e^{tA}} - I)V(g)}{t}, \text{ for } V \in \mathcal{D}(d\mathcal{R}(A))$$

and the domain $\mathcal{D}(d\mathcal{R}(A))$ of this unbounded operator $\mathcal{R}(A)$ is the subset of $\mathbb{L}_2(G)$ for which the above limit exists in \mathbb{L}_2 -sense.

Let $L_g : G \rightarrow G$ denote the left multiplication given by $L_g h = gh$. Let us choose a basis $\{A_1, \dots, A_n\}$ in $T_e(G)$, and let us define the corresponding vector fields

$$\mathcal{A}_i|_g = (L_g)_* A_i, \text{ for } i = 1, \dots, n.$$

Let us define the corresponding dual basis (‘left-invariant co-frame’) in $T_g^*(G)$ by

$$\left\langle \omega^i \Big|_g, \mathcal{A}_j \Big|_g \right\rangle = \delta_j^i
 \tag{7}$$

with δ_j^i denoting the usual Kronecker delta. Then one has $\mathcal{A}_i = d\mathcal{R}(A_i)$, and the structure constants c_{ij}^k of the Lie algebra relate via

$$[A_i, A_j] = \sum_{k=1}^n c_{ij}^k A_k \Leftrightarrow [\mathcal{A}_i, \mathcal{A}_j] = \mathcal{A}_i \circ \mathcal{A}_j - \mathcal{A}_j \circ \mathcal{A}_i = \sum_{k=1}^n c_{ij}^k \mathcal{A}_k. \quad (8)$$

If one imposes a left-invariant metric tensor field $g \mapsto \mathcal{G}_g(\cdot, \cdot) : T_g(G) \times T_g(G) \rightarrow \mathbb{R}$ to form a Riemannian manifold (M, \mathcal{G}) , then there exists a unique *constant* matrix $[g_{ij}] \in \mathbb{R}^{n \times n}$ such that

$$\mathcal{G}_g = \sum_{i,j=1}^n g_{ij} \omega^i \Big|_g \otimes \omega^j \Big|_g.$$

for all $g \in G$. We restrict ourselves to the diagonal case

$$g_{ij} = \xi_i \delta_{ij} \quad (9)$$

with $\xi_i > 0$ for $i = 1, \dots, n$ and the Kronecker δ_{ij} . As a result for all $g \in G$, the mapping $(L_{g^{-1}})_* : T_g(G) \rightarrow T_e(G)$ is unitary. The mapping is known as the Cartan-Maurer form and ‘connects’ tangent spaces in a left-invariant way. See Fig. 6 where the Maurer-Cartan form is illustrated for the group $SE(2)$ of roto-translations in the plane with group product (2). The associated ‘Cartan – connection’ (Kobayashi and Nomizu 1963) is given by

$$\nabla^- := \sum_{i,k=1}^n \omega^i \otimes (\mathcal{A}_i \circ \omega^k(\cdot)) \mathcal{A}_k,$$

inducing a covariant derivative:

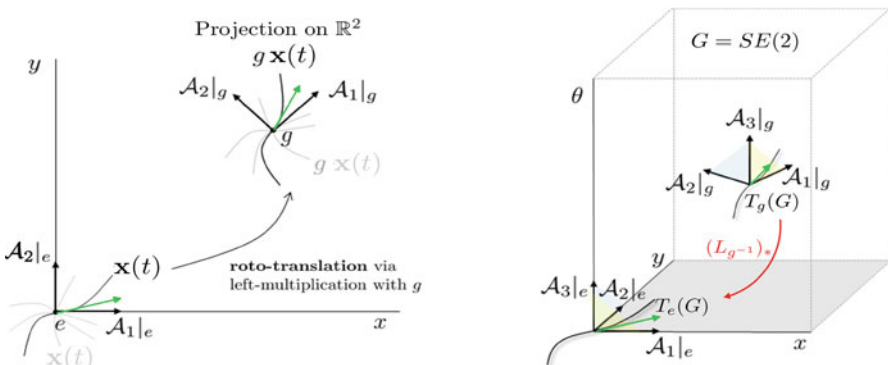


Fig. 6 The Maurer-Cartan form (in red) ‘connects’ tangent space $T_g(G)$ to $T_e(G)$ in a left-invariant way. It underlies the Lie-Cartan connection with $\nu = 0$ as can be seen in Lemma 2. Right we depict the Lie group case $SE(2) = \mathbb{R}^2 \times S^1$ and left we show spatial projections $\mathbf{x}(t)$ of the curves $\gamma(t) = (\mathbf{x}(t), \theta(t)) \in SE(2)$

$$\nabla_X^- Y := \sum_{i,k=1}^n \omega^i(X) \left(\mathcal{A}_i(\omega^k(Y)) \right) \mathcal{A}_k.$$

More precisely, for two arbitrary vector fields $X = \sum_{i=1}^n x^i \mathcal{A}_i$ and $Y = \sum_{j=1}^n y^j \mathcal{A}_j$, possibly non-left invariant (i.e. x^i and y^j need not be constant), one has

$$\nabla_X^- Y = \sum_{k=1}^n \left(\sum_{i=1}^n x^i \mathcal{A}_i y^k \right) \mathcal{A}_k.$$

This connection ∇^- has vanishing Christoffel symbols $\Gamma_{ij}^k = 0$ relative to the left-invariant frame (and co-frame) of reference, since

$$\Gamma_{ij}^k = \left\langle \omega^k, \nabla_{\mathcal{A}_i} \mathcal{A}_j \right\rangle. \tag{10}$$

This has big limitations and is not always the right choice for a connection on a Lie group G . Therefore, we consider a more general class of connections on the Lie group G , the so-called Lie-Cartan connections, as we define next. Then in particular we consider a 1-parameter class of Cartan connections. We will call these connections ‘Lie-Cartan connections’ as they are directly induced by the Lie bracket.

Definition 2. Per Cogliati and Mastrolia (2018, section 5.2), Cartan (1926), a Cartan (or canonical) connection on a Lie group is a vector bundle connection with the following additional properties:

1. Left invariance:

$$X, Y \text{ are left-invariant vector fields} \Rightarrow \nabla_X Y \text{ is a left-invariant vector field.} \tag{11}$$

2. For any $\mathbf{a} \in T_e(G)$, the exponential curve and auto-parallel curve coincide:

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0 \quad \text{where} \quad \gamma(t) = \gamma(0) \exp(t\mathbf{a}). \tag{12}$$

We now look at a specific set of Cartan connections that relate to the Lie bracket.

Definition 3 (Lie-Cartan Connection). Consider a Lie group with Lie brackets $[\cdot, \cdot]$ and structure constants $c_{ij}^k \in \mathbb{R}$ s.t. $[\mathcal{A}_i, \mathcal{A}_j] = \sum_k c_{ij}^k \mathcal{A}_k$. Then the Lie-Cartan connection indexed with $\nu \in \mathbb{R}$ equals:

$$\nabla^{[\nu]} := \sum_{i,k=1}^n \omega^i \otimes (\mathcal{A}_i \circ \omega^k(\cdot)) \mathcal{A}_k + \sum_{i,j,k=1}^n \omega^i \otimes \omega^j \nu c_{ij}^k \mathcal{A}_k \tag{13}$$

Remark 6. Left-invariant vector field X can be written as $X = \sum_{i=1}^d x^i \mathcal{A}_i$ with constant coefficients $x^i \in \mathbb{R}$. As a result, we have that for left-invariant vector fields X, Y the first term vanishes in (13), and we have that $\nabla_X^{[\nu]} Y = \nu[X, Y]$.

Remark 7. The Christoffel symbols Γ_{ij}^k (10) relative to the left-invariant moving frame of reference equal $\Gamma_{ij}^k = \nu c_{ij}^k$ and vanish iff $\nu = 0$, and indeed one has for the classical ‘minus’ Cartan connection $\nabla^- = \nabla^{[0]}$. It is common (Cogliati and Mastrolia 2018; Kobayashi and Nomizu 1963; Cartan 1926; Lee et al. 2009) to index the Cartan connections in terms of their Torsion $T_{\nabla^{[\nu]}}$ given by

$$T_{\nabla^{[\nu]}}(X, Y) := \nabla_X^{[\nu]} Y - \nabla_Y^{[\nu]} X - [X, Y] = (2\nu - 1)[X, Y],$$

for left-invariant vector fields X, Y , but we prefer to index the Lie-Cartan connections $\nabla^{[\nu]}$ with the parameter ν arising in the commutator rather than with the parameter $2\nu - 1$ in the torsion of the connection:

$$\nabla^{[\nu]} = \nabla^{2\nu-1} \text{ and thus } \nabla^{[0]} = \nabla^-, \nabla^{[1]} = \nabla^+.$$

Remark 8. Lie-Cartan connections $\nabla^{[\nu]}$ are clearly connections on a vector bundle (satisfying the standard 4 requirements for Koszul connections). Furthermore, they are indeed Cartan connections (Definition 2). The first item follows by Remark 6. The second item follows by anti-symmetry of the Christoffel symbols relative to the left-invariant frame of vector fields. We will show this later in (41).

Lemma 2. *For arbitrary smooth vector fields X, Y on G , we have*

$$(\nabla_{\dot{\gamma}}^{[0]} Y)(g) = \lim_{t \rightarrow 0} \frac{(L_{g(\gamma(t))^{-1}})_* Y(\gamma(t)) - Y(g)}{t}. \tag{14}$$

For left-invariant vector fields X, Y on G , we have

$$\begin{aligned} \nu = 0 : \quad \nabla^{[0]} Y &= 0, \\ \nu \in \mathbb{R} : \quad (\nabla_{\dot{\gamma}}^{[\nu]} Y)(g) &= \lim_{t \rightarrow 0} \frac{(\widetilde{Ad}(\gamma(vt))Y)(g) - Y(g)}{t} \text{ i.e.} \\ \nabla_X^{[\nu]} Y &= \nu[X, Y], \end{aligned} \tag{15}$$

with $\gamma(t)$ as an integral curve of left-invariant vector field X with $\gamma(0) = g \in G$, and

$$\widetilde{Ad}(q) = (L_g)_* \circ Ad(q) \circ (L_{g^{-1}})_*, \tag{16}$$

with $Ad(g) = (L_g \circ R_{g^{-1}})_* : T_e(G) \rightarrow T_e(G)$, where $(\cdot)_*$ denotes the push-forward, so that $(Ad)_* = ad$ with $ad(X_e)(Y_e) = [X_e, Y_e]$, and the transferred adjoint representation given by $\widetilde{Ad}(g) = (L_g \circ R_{g^{-1}})_* : T_g(G) \rightarrow T_g(G)$ that satisfies

$$(\widetilde{Ad})_*(X_g)(Y_g) = [X_g, Y_g] \text{ for all } g \in G. \tag{17}$$

Proof. The proof follows by direct computations; see Appendix C.

Lemma 3 (Properties of the Lie-Cartan connections). *Let X, Y, Z be left-invariant vector fields.*

The torsion tensor gives

$$T_{\nabla^{[v]}}(X, Y) = (2\nu - 1)[X, Y]. \tag{18}$$

The curvature tensor gives

$$R_{\nabla^{[v]}}(X, Y)Z = \nu(1 - \nu)[Z, [X, Y]]. \tag{19}$$

Relative to the left-invariant frame on G , we have the following components:

$$T_{jk}^i = (2\nu - 1)c_{jk}^i \text{ and } R_{k,ij}^l = \nu(1 - \nu) \sum_{q=1}^n c_{kq}^l c_{ij}^q.$$

The Lie-Cartan connections satisfy the following identity:

$$(\nabla^{[v]}\mathcal{G})(X, Y, Z) = -\nu (\mathcal{G}([X, Y], Z) + \mathcal{G}([X, Z], Y)). \tag{20}$$

Proof. The proof can be found in Appendix C. □

Remark 9 (from left-invariant vector fields to general vector fields). The formulas above in Lemma 3 only hold for left-invariant vector fields. For example, the general formula for the torsion is

$$T_{\nabla^{[v]}} = (2\nu - 1) \sum_{i,j,k=1}^n \omega^i \otimes \omega^j c_{ij}^k \mathcal{A}_k, \tag{21}$$

so only for left-invariant vector fields do we have $T_{\nabla^{[v]}}(X, Y) = (2\nu - 1)[X, Y]$. It is not a coincidence that vanishing torsion for arbitrary non-commuting vector fields gives $\nu = \frac{1}{2}$, whereas the same conclusion can be drawn from left-invariant non-commuting vector fields. In general, the torsion T_{∇} and curvature R_{∇} of a

connection ∇ , and the covariant derivative $\nabla\mathcal{G}$ of the metric tensor fields, are *tensor fields*. Therefore one has for example:

$$\begin{aligned}
 T_{\nabla^{[\nu]}}(f_1X_1 + f_2X_2, g_1Y_1 + g_2Y_2) = \\
 f_1g_1T_{\nabla^{[\nu]}}(X_1, Y_1) + f_2g_1T_{\nabla^{[\nu]}}(X_2, Y_1) + f_1g_2T_{\nabla^{[\nu]}}(X_1, Y_2) + f_2g_2T_{\nabla^{[\nu]}}(X_2, Y_2)
 \end{aligned}
 \tag{22}$$

for all $f_i, g_i \in C^\infty(G)$ and all vector fields X_i, Y_i on $G, i = 1, 2$.

Corollary 2. *Let G be a non-commutative Lie group and assume G is not two-step nilpotent. The Lie-Cartan connection $\nabla^{[\nu]}$ is*

1. *Torsion-free iff $\nu = \frac{1}{2}$,*
2. *Curvature-free iff $\nu \in \{0, 1\}$,*
3. *Metric compatible w.r.t. left-invariant metric \mathcal{G} if $\nu = 0$.*

Proof. By Remark 9, we may as well restrict our Lie-Cartan connection $\nabla^{[\nu]}$ to left-invariant vector fields, since T_∇, R_∇ and $\nabla\mathcal{G}$ are all tensor fields. Therefore, they have C^∞ -linearity (such as in (22)) in all of their entries. This C^∞ linearity allows us to turn arbitrary vector fields into left-invariant vector fields by linear combinations.

The first item now follows by (18) and G being non-commutative (i.e. there exist left-invariant vector fields X, Y s.t. $[X, Y] \neq 0$ as $2\nu - 1 = 0 \Leftrightarrow \nu = \frac{1}{2}$). Note that it also follows by (21). The second item follows by (19), and by the assumptions on G , there exist (left-invariant) X, Y, Z s.t. $[Z, [X, Y]] \neq 0$, and therefore $\nu(1 - \nu) \Leftrightarrow \nu \in \{0, 1\}$. The third item follows by (20) as for metric compatibility the covariant derivative of the metric tensor should vanish.

The above properties explain why the choices $\nu \in \{0, \frac{1}{2}, 1\}$ are the most common choices for Cartan connections. Our application (recall Fig. 1) will require torsion and metric incompatibility of connections on G . Metric incompatibility allows us to distinguish between ‘straight curves’ (auto-parallel curves with parallel velocity) and ‘shortest curves’ (distance minimizing geodesics with parallel momentum), as we will see in Theorem 1.

Expressing the Lie-Cartan Connection (and Its Dual) in Left-Invariant Coordinates

Now that we defined the Lie-Cartan connections and that we addressed their fundamental geometric properties, we express them explicitly in left-invariant coordinates.

The covariant derivative of a field $Y = \sum_{k=1}^n y^k \mathcal{A}_k$, along a smooth vector field $X = \sum_{i=1}^n x^i \mathcal{A}_i$, is given by (for details, see Remark 10)

$$\nabla_X^{[v]} Y = \sum_{k=1}^n \left(\dot{y}^k + \sum_{i,j=1}^n v c_{ij}^k x^i y^j \right) \mathcal{A}_k, \tag{23}$$

where we use common short notation (Jost 2011, (3.1.6)) $\dot{y}^k(t) = \frac{d}{dt} y^k(\gamma(t))$ which equals $\dot{y}^k(t) = (Xy^k)(\gamma(t))$ and $x^i = \dot{\gamma}^i(t)$ where $x^i \Big|_{\gamma(t)} := \gamma^i(t) = \langle \omega^i \Big|_{\gamma(t)}, \dot{\gamma}(t) \rangle$ along all flowlines γ of smooth vector field X . A ‘flowline’ is a smooth curve γ satisfying $\dot{\gamma}(t) = X_{\gamma(t)}$. With slight abuse of notation, we write

$$\nabla_{\dot{\gamma}}^{[v]} Y = \sum_{k=1}^n \left(\dot{\lambda}^k + \sum_{i,j=1}^n v c_{ij}^k \dot{\gamma}^i y^j \right) \mathcal{A}_k. \tag{24}$$

The corresponding dual connection on the co-tangent bundle is given by

$$\nabla_{\dot{\gamma}}^{[v],*} \lambda = \sum_{i=1}^n \left(\dot{\lambda}_i + \sum_{k,j=1}^n v c_{ij}^k \lambda_k \dot{\gamma}^j \right) \omega^i, \tag{25}$$

where $\lambda = \sum_{i=1}^n \lambda_i \omega^i \in T^*(G)$. Note that $\langle \nabla_X^{[v],*} \lambda, Y \rangle = X \langle \lambda, Y \rangle - \langle \lambda, \nabla_X^{[v]} Y \rangle$, and from this formula we see how (25) follows from (24). The fact that both formulas involve a plus sign for the summation reflects that the Christoffel symbols (Jost 2011) of the connection and dual connection (in the left-invariant frame) are each other’s inverse:

$$0 = v(c_{ji}^k + c_{ij}^k) = \langle \nabla_{\mathcal{A}_i}^{[v],*} \omega^k, \mathcal{A}_j \rangle + \langle \omega^k, \nabla_{\mathcal{A}_i}^{[v]} \mathcal{A}_j \rangle.$$

Remark 10. Next, we explain how (23) follows by the corresponding (previously addressed) coordinate free formulation (13):

$$\begin{aligned} \nabla_X^{[v]}(Y) &:= \nabla^{[v]}(X, Y) = \sum_{i,j,k=1}^n \left(\left(\omega^i \otimes (\mathcal{A}_i \circ \omega^k(\cdot)) \right) (X, Y) + \omega^i(X) \omega^j(Y) v c_{ij}^k \right) \mathcal{A}_k \\ &= \sum_{i,k=1}^n x^i (\mathcal{A}_i y^k) \mathcal{A}_k + \sum_{i,j,k=1}^n v c_{ij}^k x^i y^j \mathcal{A}_k, \end{aligned}$$

with $X|_{\gamma} = \sum_{i=1}^n x^i \mathcal{A}_i|_{\gamma(\cdot)} = \dot{\gamma} = \sum_{i=1}^n \dot{\gamma}^i \mathcal{A}_i|_{\gamma(\cdot)}$ and $Y = \sum_{k=1}^n y^k \mathcal{A}_k$, and

$\dot{y}^k(t) = \frac{d}{dt} y^k(\gamma(t)) = \sum_{i=1}^n x^i (\mathcal{A}_i y^k)(\gamma(t)) = X(y^k)(\gamma(t))$ via the chain-law.

(Partial) Lie-Cartan Connections for (Sub)-Riemannian Geometry

The Lie-Cartan connections introduced will be in support of understanding Riemannian geometry when the Lie group is considered as a Riemannian manifold (G, \mathcal{G}) with a left-invariant Riemannian metric tensor field given by

$$\mathcal{G} = \sum_{i,j=1}^n g_{ij} \omega^i \otimes \omega^j, \quad (26)$$

where g_{ij} constant relative to the left-invariant co-frame ω^i given by (7) s.t. matrix $[g_{ij}] \in \mathbb{R}^{n \times n}$ is symmetric positive definite. Recall we restricted ourselves to the diagonal case (9). The linear map associated with metric tensor field \mathcal{G} is written as

$$\tilde{\mathcal{G}}(X) = \mathcal{G}(X, \cdot) \quad (27)$$

In many applications (robotics (Chirikjian and Kyatkin 2001; Saccon et al. 2012), image analysis (Bekkers et al. 2015), cortical vision (Citti and Sarti 2015; Petitot 2003)), it is useful to rely on sub-Riemannian geometry (Agrachev and Sachkov 2004) where certain direction in the tangent bundle is forbidden as they go with infinite cost. This means that tangents of connecting curves are prescribed to be in a sub-bundle Δ (also known as ‘distribution’) of the tangent bundle $T(G)$, i.e.

$$\dot{\gamma}(t) \in \Delta_{\gamma(t)} \subset T_{\gamma(t)}(G) \text{ for all } t \in \text{Dom}(\gamma) \subset \mathbb{R}.$$

Typically for a controllable system, Δ and its commutators should fill the full tangent space, in view of Hörmander’s theorem (Hörmander 1968). Here we will constrain ourselves to the case that the Lie algebra is two-bracket generating

$$\Delta + [\Delta, \Delta] = T(G). \quad (28)$$

Remark 11. For instance, let us consider the car in Fig. 1 that needs to move in Lie group $SE(2)$. As the car can proceed forward (by giving gas) and change its orientation (by turning the wheel), it cannot move sideward. Optimal paths for the car boil down to sub-Riemannian geodesic problems in which the partial Cartan connection $\bar{\nabla}^{[1]}$ will play a major role, as we will show in the next subsection.

Now let us assume that we label the Lie algebra in such a way that

$$\Delta = \text{Span}\{\mathcal{A}_i\}_{i \in I} \quad (29)$$

for some index set $I \subset \{1, \dots, n\}$ and recall that we assumed (28) to hold.

This allows us to consider *partial Cartan connections* on G that will play a major role on sub-Riemannian problems on sub-Riemannian manifolds $(G, \Delta, \mathcal{G}_0)$ with

$$\mathcal{G}_0 = \sum_{i,j \in I} g_{ij} \omega^i \otimes \omega^j, \tag{30}$$

as we will see later. Again we restrict ourselves to the diagonal case $g_{ij} = \xi_i \delta_{ij}$.

Definition 4 (Partial Lie-Cartan Connection). Consider a Lie group with Lie brackets $[\cdot, \cdot]$ and structure constants $c_{ij}^k \in \mathbb{R}$ so that

$$[\mathcal{A}_i, \mathcal{A}_j] = \sum_{k=1}^n c_{ij}^k \mathcal{A}_k.$$

Consider the distribution given by (29). Then the partial Lie-Cartan connection with parameter $\nu \in \mathbb{R}$ (defined only on vector fields which map into the distribution) equals

$$\overline{\nabla}^{[\nu]} := \sum_{i,k \in I} \omega^i \otimes (\mathcal{A}_i \circ \omega^k(\cdot)) \mathcal{A}_k + \sum_{i,j,k \in I} \omega^i \otimes \omega^j \nu c_{ij}^k \mathcal{A}_k. \tag{31}$$

So from this definition, we deduce that

$$\begin{aligned} \overline{\nabla}_Y^{[\nu]} X &= \sum_{i,j,k \in I} \left(\dot{y}^k + \nu c_{ij}^k \dot{y}^i y^j \right) \mathcal{A}_k, \\ \overline{\nabla}_X^{[\nu],*} \lambda &= \sum_{i=1}^n \left(\dot{\lambda}_i + \nu \sum_{k=1}^n \sum_{j \in I} c_{ij}^k \lambda_k \dot{y}^j \right) \omega^i, \end{aligned} \tag{32}$$

where we highlighted the difference with the full Lie-Cartan connection in red, compared to Definition 3, (24), (25). Again X, Y are vector fields and λ a dual vector field and γ is an integral curve of X , and $\dot{y}^k(t) := \frac{d}{dt} y^k(\gamma(t)), \dot{\lambda}_k(t) := \frac{d}{dt} \lambda_k(\gamma(t))$.

The Special Case of Interest $\nu = 1$ and Hamiltonian Flows for the Riemannian Geodesic Problem on G

Let $C : G \rightarrow \mathbb{R}^+$ be an a priori smooth cost (or mobility) for moving through q Lie group G that is bounded from below. For the moment, it can be considered as constant, but later on in the application sections, it will play an important role.

Then the Riemannian metric tensor field \mathcal{G} induces a Riemannian metric on G :

$$d_{\mathcal{G}}(g_0, g_1) := \min_{\gamma \in \text{Lip}([0, 1], G)} \int_0^1 C(\gamma(t)) \sqrt{\mathcal{G}_{|\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt. \tag{33}$$

$\gamma(0) = g_0,$
 $\gamma(1) = g_1,$

for all $g_0, g_1 \in G$. The sub-Riemannian metric tensor field \mathcal{G}_0 induces a **sub-Riemannian** metric on G by $d_{\mathcal{G}_0} : G \times G \rightarrow \mathbb{R}^+$ on G :

$$d_{\mathcal{G}_0}(g_0, g_1) := \min_{\substack{\gamma \in \text{Lip}([0, 1], G) \\ \gamma(0) = g_0, \\ \gamma(1) = g_1, \\ \forall t \in [0, 1] : \dot{\gamma}(t) \in \Delta|_{\gamma(t)}}} \int_0^1 C(\gamma(t)) \sqrt{\mathcal{G}_0|_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt. \quad (34)$$

for all $g_0, g_1 \in G$. The next theorem motivates the choice $\nu = 1$ for the Lie-Cartan connection $\nabla^{[1]}$, that is, underlying the Hamiltonian flow associated with (33).

Recall from geometric control theory (Agrachev and Sachkov 2004; Agrachev et al. 2020) that the Pontryagin maximum principle describes the Hamiltonian flow. It allows us to simultaneously analyse all lifted geodesics $(\gamma(\cdot), \lambda(\cdot))$ in the co-tangent bundle $T^*(G)$, where $\lambda(\cdot)$ denotes the momentum along the geodesic. This is important, as a single (analytic) description of a geodesic typically does not say that much. It is rather the continuum of geodesics and how their lifted versions in $T^*(G)$ are organized that help us understanding their behaviour. This is well known for classical problems like the ‘mathematical pendulum’, but it is also crucial in understanding the cut locus (Sachkov 2011) of the ‘sub-Riemannian geodesic’ problem or the ‘elastica problem’ (Sachkov 2008; Bryant and Griffiths 1986; Mumford 1994) in $SE(2)$ as shown by Sachkov. For cortical contour perception models (Petitot 2017; Citti and Sarti 2006), this is equally important.

However, the underlying deep role of Cartan connections is often not mentioned, despite its use in deriving simple solutions to cusp-free sub-Riemannian geodesics in $SE(2)$ solving association field models (Duits et al. 2016) and for new solutions (Duits et al. 2016) of sub-Riemannian geodesics in $SE(3)$. The power of such Cartan connections is also stressed in the Lagrangian geometric viewpoint on optimal curves by Bryant et al. (2003), Bryant and Griffiths (1986).

In this work, we take the venture point of the geometric Hamiltonian viewpoint on (sub-)Riemannian geometry (Agrachev et al. 2020; Agrachev and Sachkov 2004) and include a key element coming from Bryant’s Lagrangian viewpoint on contact manifolds (and his analysis of ‘elastica’ Bryant and Griffiths 1986): That is (partial) Cartan connections that carry torsion. They will allow us to distinguish between ‘shortest’ and ‘straight’ curves in Lie groups. For multi-orientation image processing, this is very useful and intuitive as we show in Theorem 1, Fig. 7 and section “Overview of Image Analysis Applications for $G = SE(d)$ ”.

Theorem 1. *In a Riemannian manifold $(G, T(G), \mathcal{G})$, with the tangent bundle $T(G)$ and metric tensor field \mathcal{G} defined in (26), the induced metric $d_{\mathcal{G}}$ defined in (33) and the Lie-Cartan connection $\nabla^{[v]}$ for $\nu = 1$ defined in (13), we have the following relations for ‘straight’ curves:*

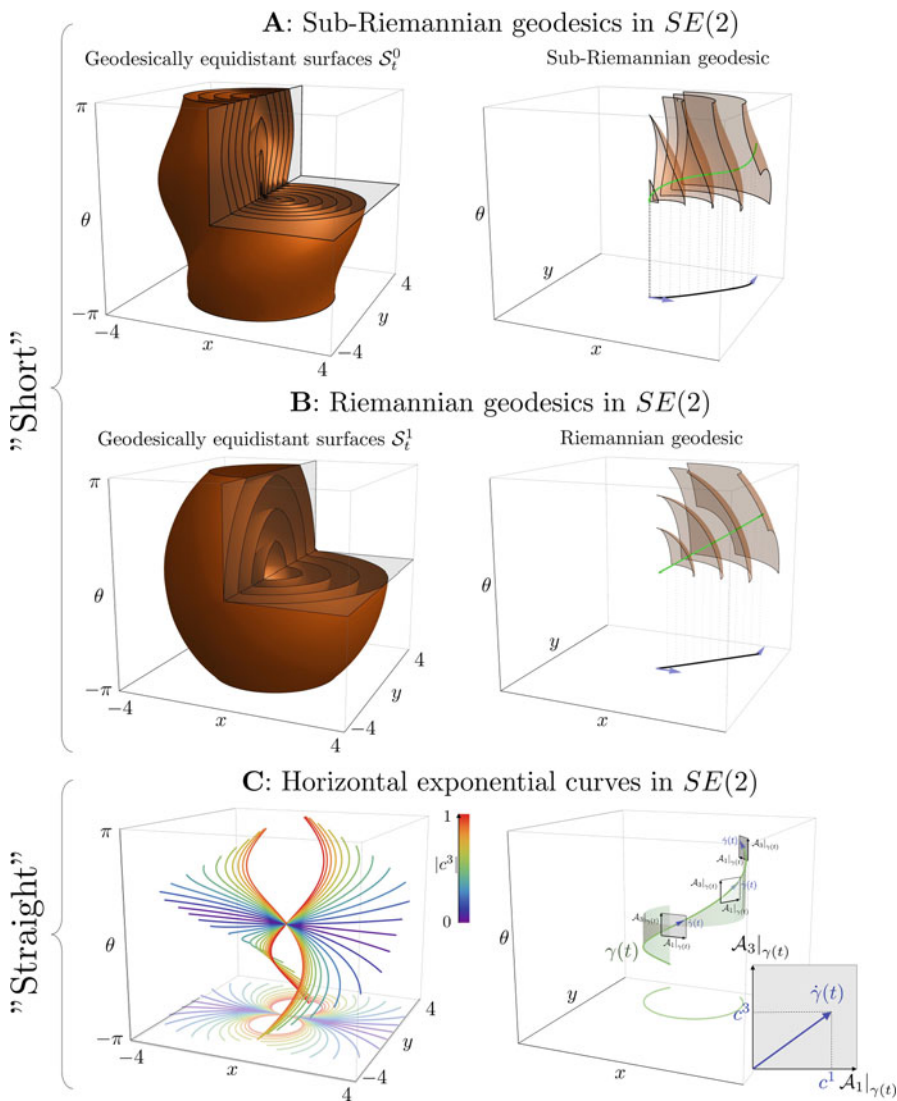


Fig. 7 (a) Geodesically equidistant surfaces $S_t^\epsilon = \{g \in SE(2) | d_\epsilon(0, g) = t\}$ and geodesic (in green) for the sub-Riemannian case: $\epsilon = 0$ and $C = 1$. (b) Geodesically equidistant surfaces S_t^ϵ and geodesic for the isotropic Riemannian case: $\epsilon = 1$ and $C = 1$. Now the geodesics are straight lines. (c) A set of horizontal exponential curves for which $\dot{\gamma}(\tau) = c^1 \mathcal{A}_1|_{\gamma(\tau)} + c^3 \mathcal{A}_3|_{\gamma(\tau)} \in \Delta$, with constant tangent vector components c^1 and c^3 . Such curves are auto-parallel (‘straight curves’ in torqued and curved geometry modelled by Lie-Cartan connection $\nabla^{[1]}$)

$$\begin{aligned} \gamma \text{ is a } \nabla^{[1]}\text{-straight curve} &\Leftrightarrow \gamma \text{ is an exponential curve} \\ \Leftrightarrow \nabla_{\dot{\gamma}}^{[1]}\dot{\gamma} = 0 &\Leftrightarrow \gamma \text{ has } \nabla^{[1]}\text{-auto-parallel velocity,} \end{aligned} \tag{35}$$

and the following for ‘shortest’ curves (minimizers in (33)); recall also (27):

$$\begin{aligned} \gamma \text{ is a shortest curve} &\Leftrightarrow \gamma \text{ is a minimizing curve in } d_{\mathcal{G}} \\ \Rightarrow \begin{cases} \nabla_{\dot{\gamma}}^{[1],*}\lambda = 0 \\ \dot{\gamma} = \tilde{\mathcal{G}}^{-1}\lambda \end{cases} &\Leftrightarrow \gamma \text{ has } \nabla^{[1],*}\text{-parallel momentum.} \end{aligned} \tag{36}$$

In a sub-Riemannian (SR) manifold $(SE(2), \Delta, \mathcal{G}_0)$ with sub-bundle Δ defined in (31), the sub-Riemannian metric tensor \mathcal{G}_0 (30) and distance (34) and partial Cartan connection (31), we have the following relations for ‘straight’ curves:

$$\begin{aligned} \gamma \text{ is a } \bar{\nabla}^{[1]}\text{-straight curve} &\Leftrightarrow \gamma \text{ is a horizontal exponential curve} \\ \Leftrightarrow \bar{\nabla}_{\dot{\gamma}}^{[1]}\dot{\gamma} = 0 &\Leftrightarrow \gamma \text{ has } \bar{\nabla}^{[1]}\text{-auto-parallel velocity,} \end{aligned} \tag{37}$$

and the following for ‘shortest’ curves (minimizers in (34)):

$$\begin{aligned} \gamma_0 \text{ is a shortest curve} &\Leftrightarrow \gamma_0 \text{ is a minimizing curve in } d_{\mathcal{G}_0} \\ \Rightarrow \begin{cases} \bar{\nabla}_{\dot{\gamma}_0}^{[1],*}\lambda = 0 \\ \dot{\gamma}_0 = \tilde{\mathcal{G}}_0^{-1}\mathbb{P}_{\Delta}^*\lambda \end{cases} &\Leftrightarrow \gamma_0 \text{ has } \bar{\nabla}^{[1],*}\text{-parallel momentum,} \end{aligned} \tag{38}$$

in which \mathbb{P}_{Δ}^* is the projection $\mathbb{P}_{\Delta}^* \left(\sum_{i=1}^n \lambda_i \omega^i \right) = \sum_{i \in I} \lambda_i \omega^i$. For the reverse in (36) and (38) and for a minimizing curve between $g_1 = \gamma(0)$ and $g_2 = \gamma(t)$, one must have $0 \leq t \leq t_{cut} = \min\{t_{conj}(\lambda(0)), t_{Max,1}(\lambda(0))\}$, cf. Agrachev and Sachkov (2004); Sachkov (2011) for details.

They are found by steepest descent:

$$\gamma(t) = \gamma(0) + \int_0^t \text{grad}_{\mathcal{G}} W(\gamma(s)) \, ds, \tag{39}$$

on distance maps $W(g) = d_{\mathcal{G}}(g, e)$ that are viscosity solutions of eikonal PDE:

$$\begin{cases} \|\text{grad}_{\mathcal{G}} W(g)\| = \sqrt{\mathcal{G}|_g(\text{grad}_{\mathcal{G}} W(g), \text{grad}_{\mathcal{G}} W(g))} = 1, \\ W(e) = 0, \end{cases} \tag{40}$$

with (metric-intrinsic) gradient $\text{grad}_{\mathcal{G}} W(g) = \tilde{\mathcal{G}}^{-1} dW(g)$, as this only gives global minimizing curves, even in the SR setting $\mathcal{G} \rightarrow \mathcal{G}_0$.

Proof. First, we address the ‘shortest curves’ part of the theorem. The items (36) and (38) follow by the Pontryagin maximum principle Agrachev and Sachkov (2004) and Theorem 2 in Appendix A. Theorem 2 proves the actual fundamental relation between the (partial) Lie-Cartan connection $\nabla^{[1]}$ to the Hamiltonian flow, for the (sub)-Riemannian setting. Here we stress that PMP provides only *local* optimality of geodesics.

The geodesics are found by the exponential map that integrates the Hamiltonian flow $(\lambda(0), t) \mapsto (\gamma(t), \lambda(t)) = e^{t\mathfrak{h}}(\lambda_0)$.

Optimality of $t \mapsto \gamma(t)$ requires t to be less than the cut time. Such a cut time is the minimum of the conjugate time $t_{conj}(\lambda(0)) \in \mathbb{R} \cup \{\infty\}$ where local optimality is lost and the first Maxwell time $t_{Max,1}$, where two equidistant geodesics meet for the first time and where global optimality is lost. Now $t \leq t_{cut}(\lambda(0))$ is guaranteed by steepest descent (39) on the distance maps W which are obtained as viscosity solutions (Crandall and Lions 1983; Evans 2010) to the eikonal PDE. This is well known for the Riemannian case (Mantegazza and Menonucci 2002; Crandall and Lions 1983) but also applies to the sub-Riemannian (For an intuitive illustration inside, the viscosity solutions of the PDEs non-optimal wavefronts are cut (at the first Maxwell set) in the sub-Riemannian setting (42); see (Bekkers et al. 2015, Fig.3).) case (Monti and Cassano 2001; Bekkers et al. 2015) and holds even in more general Finsler geometrical settings (Duits et al. 2018).

Secondly, regarding the ‘straight curves’ (35), one has (by (24)) and anti-symmetry of the structure constants (8) that

$$\begin{aligned} \nabla_{\dot{\gamma}}^{[1]}\dot{\gamma} = 0 &\Leftrightarrow \forall_{k \in \{1, \dots, n\}} : \ddot{\gamma}^k - \sum_{i,j=1}^n c_{ij}^k \dot{\gamma}^i \dot{\gamma}^j = \ddot{\gamma}^k = 0 \\ &\Leftrightarrow \forall_{k \in \{1, \dots, n\}} : \langle \omega^k \Big|_{\gamma}, \dot{\gamma} \rangle =: \dot{\gamma}^k = c^k = \text{constant} \quad (41) \\ &\Leftrightarrow \gamma(t) = \gamma(0) e^{t \sum_{k=1}^n c^k A_k}, \end{aligned}$$

with $\ddot{\gamma}^k(t) = \frac{d}{dt} \dot{\gamma}^k(t)$. Note that the first, third and fourth statements in (35) are just tautological, so that (41) proves the remaining second equivalence. The SR-case (37) follows similarly by (32) taking into account the restriction to (29) via projection P_{Δ}^* which means we constrain the summations to index set I and set $\dot{\gamma}^i = 0$ if $i \notin I$. □

For 2D image processing via orientation scores (Duits and Franken 2010a,b; Bekkers 2017; Duits et al. 2007), we must consider the special case:

$$\begin{aligned}
 G &= SE(2), \\
 \mathcal{A}_1 &= \cos \theta \partial_x + \sin \theta \partial_y, \quad \mathcal{A}_2 = -\sin \theta \partial_x + \cos \theta \partial_y, \quad \mathcal{A}_3 = \partial_\theta, \\
 I = \{1, 3\} &\Rightarrow \Delta = \text{span}\{\mathcal{A}_1, \mathcal{A}_3\}, \\
 \mathcal{G}_0 &= \xi^2 \omega^1 \otimes \omega^1 + \omega^3 \otimes \omega^3 \\
 &= \xi^2 (\cos \theta dx + \sin \theta dy) \otimes (\cos \theta dx + \sin \theta dy) + d\theta \otimes d\theta, \\
 \mathcal{G} &= \mathcal{G}_0 + \xi^2 \zeta^{-2} (-\sin \theta dx + \cos \theta dy) \otimes (-\sin \theta dx + \cos \theta dy),
 \end{aligned}
 \tag{42}$$

with curve stiffness parameter $\xi > 0$ and with anisotropy parameter $0 < \zeta \ll 1$. The basic idea here is that one considers a path optimization via a Reeds-Shepp car moving in the orientation score ($\xi > 0$ puts relative costs on moving forward to turning the wheel of the car); see Fig. 1. In Fig. 1 the green plane indicates $\Delta_g \subset T_g(G)$ for some $g = (x, y, \theta) \in SE(2)$. This 2D subspace is the subspace to which local velocities are constrained in the sub-Riemannian setting, i.e. $\dot{\gamma}(0) \in \Delta_g$ for all smooth ‘horizontal’ curves γ in the sub-Riemannian manifold with $\gamma(0) = g$.

The geometric control problem (34) is then concerned with finding the shortest path for the car in the orientation score. See Fig. 7 for an intuitive illustration of Theorem 1 in the $SE(2)$ setting (42). In order to generalize this special case from $d = 2$ to $d = 3$, we must distinguish between the homogeneous space $\mathbb{R}^d \times S^{d-1}$ of positions and orientations on which the rigid body motion group $SE(d)$ acts and the Lie group itself. This will be the topic of the next section.

The Homogeneous Space \mathbb{M}_d of Positions and Orientations

We consider geometric image processing on the homogeneous space of positions and orientations which equals the partition of left cosets given by

$$\mathbb{M}_d := \mathbb{R}^d \times S^{d-1} := G/H \tag{43}$$

for $d \in \{2, 3\}$, with roto-translation group $G = SE(d) = \mathbb{R}^d \times SO(d)$ and with subgroup $H = \{\mathbf{0}\} \times \text{Stab}_{SO(d)}(\mathbf{a})$. Here $\text{Stab}_{SO(d)}(\mathbf{a}) = \{\mathbf{R} \in SO(d) \mid \mathbf{R}\mathbf{a} = \mathbf{a}\}$ denotes the subgroup of $SO(d)$ that stabilizes an a priori reference axis $\mathbf{a} \in S^{d-1}$.

In case $d = 2$, H consist only of the unity element and $\mathbb{R}^2 \times S^1 \equiv SE(2)$.

Therefore, let us explain the remaining case $d = 3$, where we set $\mathbf{a} = (0, 0, 1)^T$. Then the subgroup H can be parameterized as follows:

$$H = \{h_\alpha := (\mathbf{0}, \mathbf{R}_{\mathbf{a},\alpha}) \mid \alpha \in [0, 2\pi)\}, \tag{44}$$

where we recall that $\mathbf{R}_{\mathbf{a},\alpha}$ denotes a (counterclockwise) rotation around the reference axis \mathbf{a} . The reason behind this construction is that the group $SE(3)$ acts transitively on $\mathbb{R}^3 \times S^2$ by $(\mathbf{x}', \mathbf{n}') \mapsto g \odot (\mathbf{x}', \mathbf{n}')$ given by

$$g \odot (\mathbf{x}', \mathbf{n}') = (\mathbf{R}\mathbf{x}' + \mathbf{x}, \mathbf{R}\mathbf{n}'), \quad \text{for all } g = (\mathbf{x}, \mathbf{R}) \in SE(3), (\mathbf{x}', \mathbf{n}') \in \mathbb{R}^3 \times S^2.$$

Recall that by the definition of the left cosets, one has $g_1 \sim g_2 \Leftrightarrow g_1^{-1}g_2 \in H$. The latter equivalence simply means that for $g_1 = (\mathbf{x}_1, \mathbf{R}_1)$ and $g_2 = (\mathbf{x}_2, \mathbf{R}_2)$, one has

$$g_1 \sim g_2 \Leftrightarrow \mathbf{x}_1 = \mathbf{x}_2 \text{ and } \exists_{\alpha \in [0, 2\pi)} : \mathbf{R}_1 = \mathbf{R}_2 \mathbf{R}_{\mathbf{a}, \alpha}.$$

The equivalence classes $[g] = \{g' \in SE(3) \mid g' \sim g\}$ are often just denoted by

$$(\mathbf{x}, \mathbf{n}) \in \mathbb{M}_3.$$

They consist of all $g = (\mathbf{x}, \mathbf{R}_{\mathbf{n}}) \in SE(3)$ that map reference point $(\mathbf{0}, \mathbf{a})$ onto $(\mathbf{x}, \mathbf{n}) \in \mathbb{R}^3 \times S^2 : g \odot (\mathbf{0}, \mathbf{a}) = (\mathbf{x}, \mathbf{n})$, where $\mathbf{R}_{\mathbf{n}}$ is *any* rotation that maps $\mathbf{a} \in S^2$ onto $\mathbf{n} \in S^2$.

The Metric Models on \mathbb{M}_d : Shortest Curves and Spheres

The shortest curves (distance minimizers) are computed by steepest descent on the distance maps; recall Theorem 1 and Fig. 1. For a visualization of a steepest descent (according to Theorem 1) in the lifted image data defined on \mathbb{M}_d , see Fig. 8.

For uniform cost, the non-data-driven uniform cost case (i.e. $C = 1$ in (33)), they can often be computed analytically, and also the cut locus $t_{cut}(\lambda(0))$ can be computed analytically (Sachkov 2011) for $(G = SE(2), \Delta = \text{span}\{\mathcal{A}_1, \mathcal{A}_3\}, \mathcal{G}_0)$.

For the higher dimensional case

$$(G = SE(3), \Delta = \text{span}\{\mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5\}, \mathcal{G}_0 = \xi^2 \omega^3 \otimes \omega^3 + \omega^4 \otimes \omega^4 + \omega^5 \otimes \omega^5),$$

the curves can be computed analytically (Duits et al. 2016), and the cut locus mainly numerically (Duits et al. 2016, 2018). For an explicit definition of the left-invariant vector fields on $SE(3)$, see Appendix B. The corresponding distance on \mathbb{M}_3 is then given by

$$d_{\mathbb{M}_3}(\mathbf{p}_1, \mathbf{p}_2) = \min_{h_1, h_2 \in H} d_{SE(3)}((\mathbf{x}_1, \mathbf{R}_{\mathbf{n}_1})h_1, (\mathbf{x}_1, \mathbf{R}_{\mathbf{n}_2})h_2), \tag{45}$$

where $\mathbf{R}_{\mathbf{n}_i}$ are any rotations mapping a priori reference axis \mathbf{a} onto \mathbf{n}_i .

Numerical implementations to compute the shortest distance curves in $d_{\mathbb{M}_d}$ can be done by accurate, relatively slow, PDE iterations (Bekkers et al. 2015) or better by more efficient anisotropic fast-marching algorithms (Mirebeau 2018) that are sufficiently accurate (Sanguinetti et al. 2015). For state-of-the-art fast-marching approaches, we refer to work of Jean-Marie Mirebeau (2018) and several variants including a semi-Lagrangian fast-marching approach (where acuteness of stencils guarantees a single pass algorithm with convergence results) (Mirebeau 2014). See

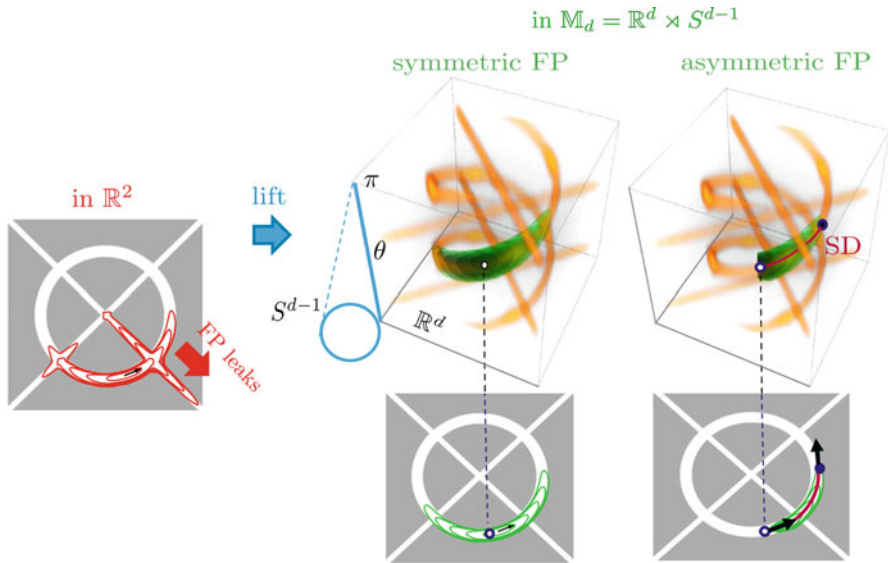


Fig. 8 Geodesic front propagation directly in the image domain leaks at crossings (left). To overcome this complication, we lift the data to $\mathbb{M}_d = \mathbb{R}^d \times S^{d-1}$ (here $d = 2$). This gives a mobility/cost C in the lifted space (Duits et al. 2018; Bekkers et al. 2015). This determines the distance on (34), and we apply geodesic front propagation (FP) in \mathbb{M}_d via the eikonal equation (40), as depicted by the growing opaque spheres in green. We depict FP in symmetric (sub)-Riemannian models and in asymmetric improvements (Duits et al. 2018). In purple, we indicate the steepest descent (SD) backtracking (39)

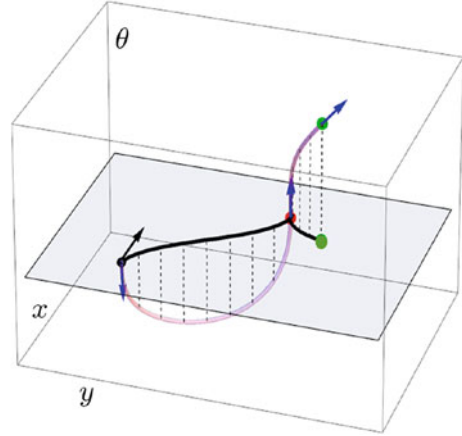
also Duits et al. (2018). For a more recent Hamiltonian fast-marching approach, see Mirebeau and Portegies (2019). The Hamiltonian approach directly relates to the PDE approach in Bekkers et al. (2015) also discretizing the eikonal equations, with the main difference that at each step one updates only the relevant voxels (instead of a full volume) in a single pass algorithm which leads to a tremendous speedup (Sanguinetti et al. 2015).

In practice, it does not make a big difference if one relies on highly anisotropic Riemannian geodesic models (with an anisotropy of, say, about 10) to simplify sub-Riemannian geodesic models (with infinite anisotropy). See Duits et al. (2018, Thm.2) and Sanguinetti et al. (2015) for theoretical and practical underpinning of this statement.

For the homogeneous space of positions and orientations, both the highly anisotropic Riemannian and the sub-Riemannian models have major benefits over isotropic Riemannian models in vessel tracking applications (Bekkers et al. 2017, 2018); see section “Overview of Image Analysis Applications for $G = SE(d)$ ”.

Furthermore, there exist several extensions of the highly anisotropic Riemannian or sub-Riemannian models (Duits et al. 2018). There the most relevant extended models are:

Fig. 9 An example of a smooth sub-Riemannian geodesic $\gamma = (x(\cdot), y(\cdot), \theta(\cdot))$ (in purple), whose spatial projection (in black) shows a cusp (red point). A cusp point is a point (x, y, θ) on γ such that the velocity (black arrow) $\dot{\mathbf{x}}$ of the projected curve $\mathbf{x}(\cdot) = (x(\cdot), y(\cdot))$ switches sign at (x, y)



- The **anisotropic, asymmetric, positive control variant** where one forces positive spatial control (see Fig. 8) to avoid the problem of cusps (see Fig. 9). Essentially, it means that the metric tensor fields in (42) are replaced by the following Finsler functions on $T(\mathbb{M}_d)$:

$$\mathcal{F}_0^+(\mathbf{p}, \dot{\mathbf{p}})^2 := \begin{cases} \xi^2 |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \|\dot{\mathbf{n}}\|^2 & \text{if } \dot{\mathbf{x}} \times \mathbf{n} \text{ and } \dot{\mathbf{x}} \cdot \mathbf{n} \geq 0, \\ +\infty & \text{otherwise,} \end{cases} \quad (46)$$

with $\mathbf{p} = (\mathbf{x}, \mathbf{n})$ and $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}})$, and while including a highly anisotropic Riemannian approximation and the mobility/cost C (recall (33) and (34)) into the Finsler function, one obtains altogether

$$\mathcal{F}_\zeta^+(\mathbf{p}, \dot{\mathbf{p}})^2 := (C(\mathbf{p}))^2 \left(\xi^2 |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \frac{\xi^2}{\zeta^2} \|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 + (\frac{1}{\zeta^2} - 1)(\dot{\mathbf{x}} \cdot \mathbf{n})_-^2 + \|\dot{\mathbf{n}}\|^2 \right) \quad (47)$$

with $0 < \zeta \ll 1$. For details and illustrations, see Duits et al. (2018).

- **The projective line bundle variant** (where anti-podal points are identified) that partly resolves the cusp problem (Bekkers et al. 2017, ch.4) and that better relates to cortical sub-Riemannian models (Petitot 2017). It can be shown that it boils down to taking the minimum distance over the four cases that arise by flipping (i.e. $\mathbf{n}_i \mapsto -\mathbf{n}_i$) or not flipping the two boundary conditions. For details, see Bekkers et al. (2017).

In Fig. 10, we depict growing spheres of several models. It can be observed in the sub-Riemannian setting such spheres reveal folds which are the closure of the first Maxwell set where two geodesics with equal length meet. This is easily understood as geodesic back propagation via steepest descent (with the same speed) can be done along two directions orthogonal to each of the orthogonal wavefronts that meet at

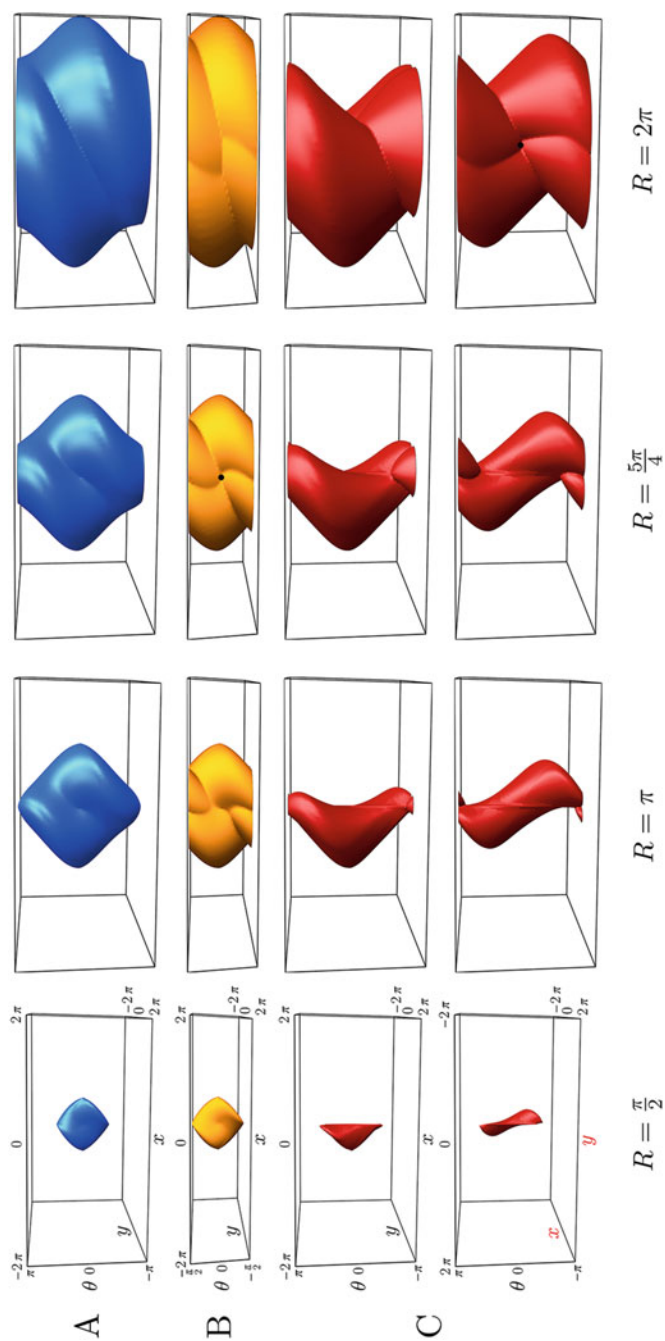


Fig. 10 The development of spheres centred around $\mathbf{e} = (0, 0, 0)$ with increasing radius R . **A:** The normal SR spheres on \mathbb{M}_2 given by $\{\mathbf{p} \in \mathbb{M} \mid d_{\mathcal{F}_0}(\mathbf{p}, \mathbf{e}) = R\}$ where the folds reflect the first Maxwell sets (Bekkers et al. 2015; Sachkov 2011). **B:** The SR spheres with identification of antipodal points with additional folds (first Maxwell sets) due to π -symmetry. **C:** The asymmetric Finsler norm spheres given by $\{\mathbf{p} \in \mathbb{M} \mid d_{\mathcal{F}_0}(\mathbf{p}, \mathbf{e}) = R\}$ visualized from two perspectives with extra folds (first Maxwell sets) at the back $(-\mu, 0, 0)$. The black dots indicate points with two folds. For details, see Duits et al. (2018)

folds on the spheres. In Fig. 11, we depict the cut locus where geodesic fronts lose their optimality for the projective line bundle case.

Straight Curve Fits

Let G be the roto-translation Lie group $G = SE(d) = \mathbb{R}^d \rtimes SO(d)$. Given differentiable data $f : G \rightarrow \mathbb{R}$ and a point $g \in G$, we consider the exponential curve $t \mapsto \gamma_{g,\mathbf{c}}(t)$ passing through g at time $t = 0$ with tangent $\gamma'_{g,\mathbf{c}}(0) = \mathbf{c} \in T_g(G)$.

Definition 5. Let $g \in G$. Let $t \geq 0$ and let $\mathbf{c} \in T_g(G)$:

$$\gamma_{g,\mathbf{c}}(t) := g \exp_G \left((L_{g^{-1}})_* \mathbf{c} t \right) \tag{48}$$

Such an exponential curve (recall Fig. 7) is determined by $\mathbf{c} = \sum_{i=1}^n c^i \mathcal{A}_i|_g \in T_g(G)$. Expressed in the left-invariant moving frame of reference, we have

$$\begin{cases} \frac{d}{dt} \gamma_{g,\mathbf{c}}(t) = \sum_{i=1}^n c^i \mathcal{A}_i|_{\gamma_{g,\mathbf{c}}(t)}, & t \in \mathbb{R}, \\ \gamma'_{g,\mathbf{c}}(0) = \mathbf{c} \in T_g(G). \end{cases}$$

The tangent to the locally best fitting exponential curve will be the first vector of our locally adaptive frame (henceforth referred to as ‘gauge frame’). The mathematical details on the fitting procedure on how to compute the best fitting exponential curve and the local optimization problem that defines such a best exponential curve fit will follow in section “[Exponential Curve Fits of the Second Order Are Found by SVD of the Hessian](#)”. For now, to get a geometrical intuition, see Fig. 12. Inclusion of such a gauge frame has the following benefits:

- It allows for curvature adaption in crossing-preserving PDE enhancements (Smets 2019) and curvature estimation in 2D (Franken 2008) that can be employed for biomarkers of diabetes (see section “[Straight Curve Application: Biomarkers for Diabetes](#)”) and 3D (Janssen et al. 2017).
- It allows for a reduction of orientation samples (Franken 2008) (even to $N = 4$) in $SE(2, N) = \mathbb{R}^2 \rtimes \mathbb{T}_N$, where \mathbb{T}_N is the finite subgroup of $\mathbb{T} \equiv SO(2)$ consisting of N equidistant samples on the circle/torus. The reason for this is that it can remove bias toward sampled orientations, as it takes into account deviation from horizontality (Franken and Duits 2009). Similar considerations apply to the $SE(3)$ setting (Janssen et al. 2018).
- More effective geometric vessel segmentation algorithms (Zhang et al. 2016).

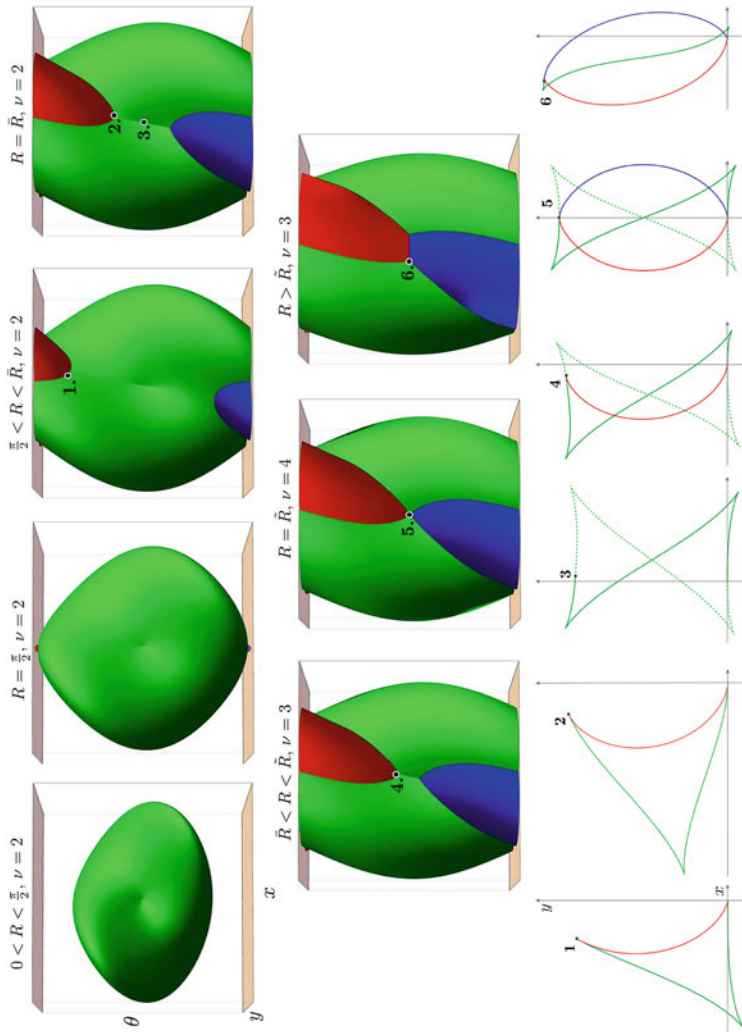


Fig. 11 Top two rows: Evolution of the first Maxwell set as the radius R of the SR spheres (with uniform cost $C = 1$) increases. The first Maxwell sets are visible via folds on the spheres, as steepest descent (39) has more than one equal length options. Bottom: Equal length SR length minimizers (shortest curves) in the projective line bundle case ending at the points indicated in the top two rows. The multiplicity of the Maxwell points is indicated by ν and the characteristic radii \tilde{R} , $\tilde{\tilde{R}}$, where the multiplicity changes from 2 to 3 and from 3 to 4 can be computed analytically; see Bekkers et al. (2017)

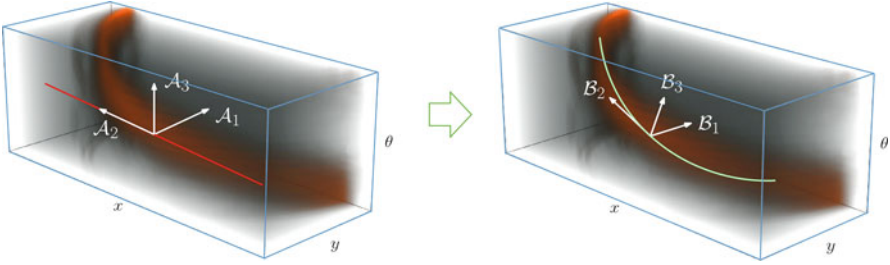


Fig. 12 Illustrating gauge frame fitting at a fixed point $g \in SE(2)$. Top: left invariant frame where $\mathcal{A}_d = \mathbf{n} \cdot \nabla_{\mathbb{R}^2}$, as indicated by the red line. Bottom: we choose a frame with $\mathcal{B}_d = \mathbf{c}$ given by (51) or (55) that takes into account the local curvature. In green, we see the corresponding exponential curve fit $\gamma_{g,\mathbf{c}}$ to the data

Next, we revise the technical considerations in Duits et al. (2016) in a coordinate-free way. The considerations in Duits et al. (2016) are more general, as it discusses exponential curve fits of the first order (solved by spectral decomposition of a structure tensor of f) and exponential curve fits of the second order. The latter are either solved by spectral decomposition of a symmetric sum of the nonsymmetric Hessian of f , or they are solved by spectral decomposition of a symmetric product of the Hessian of f .

Here we shall only be concerned with second-order exponential curve fits solved by spectral decomposition of the symmetric product of the Hessian, i.e. by a singular value decomposition (SVD) of the Hessian Hf of f .

Let us remark upfront that a Hessian depends on the choice of connection ∇ on $T(G)$ (inducing a dual connection ∇^* on $T^*(G)$), since by definition (Jost 2011) one has

$$Hf = \nabla^* df \tag{49}$$

for all $f \in C^2(G, \mathbb{R})$. It will turn out in section “Exponential Curve Fits of the Second Order Are Found by SVD of the Hessian” that the theory of best exponential curve fits of second order will boil down to an SVD of $\nabla^* df$, where one can either choose $\nabla = \nabla^{[0]}$ or $\nabla = \nabla^{[1]}$ as the corresponding linear maps associated to the Hessian are each other’s adjoints. Indeed, a brief computation in the frame $\{\mathcal{A}_i\}_{i=1}^n$ of left-invariant vector fields gives us

$$\begin{aligned} ((\nabla^{[0],*})_{\mathcal{A}_i} df) (\mathcal{A}_j) &= \mathcal{A}_i \mathcal{A}_j f, \\ ((\nabla^{[1],*})_{\mathcal{A}_i} df) (\mathcal{A}_j) &= \mathcal{A}_i \mathcal{A}_j f - \sum_{k=1}^n c_{ij}^k \mathcal{A}_k f = \mathcal{A}_i \mathcal{A}_j f - (\mathcal{A}_i \mathcal{A}_j - \mathcal{A}_j \mathcal{A}_i) f \\ &= \mathcal{A}_j \mathcal{A}_i f, \end{aligned} \tag{50}$$

where i is the row index and j is the column index.

Exponential Curve Fits of the Second Order Are Found by SVD of the Hessian

In this section, we will show that exponential curve fits (like the white line in Fig. 12) are computed by singular value decomposition of the Hessian. This technique is well known on the Lie group $G = \mathbb{R}^2$ and widely used in image processing to compute locally adaptive frames (or ‘gauge frames’) (Haar Romenij 2003), but generalizing this to a Lie group like $G = SE(2)$ requires the Lie-Cartan connections (for $\nu = 0$ or $\nu = 1$ as we will see).

We start by defining the main gauge vector (by means of the Lie-Cartan connection) as

$$\mathcal{B}_d|_g := \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| \left(\nabla_{\mathbf{c}}^{[0]} \operatorname{grad} f \right) (g) \right\|, \tag{51}$$

where the (metric-intrinsic) gradient is given by the vector field $\operatorname{grad} f = \sum_{i=1}^n \xi_i^{-2} (\mathcal{A}_i f) \mathcal{A}_i$. Note that by direct computations, one has

$$\begin{aligned} \mathcal{B}_d|_g &= \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| \left(\nabla_{\mathbf{c}}^{[0]} \operatorname{grad} f \right) (g) \right\| \\ &\stackrel{\text{Lemma 2}}{=} \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| \lim_{t \rightarrow 0} \frac{\left(L_{g\gamma_{g,\mathbf{c}}(t)^{-1}} \right)_* \operatorname{grad} f \left(\gamma_{g,\mathbf{c}}(t) \right) - \operatorname{grad} f(g)}{t} \right\| \\ &\stackrel{(9)}{=} \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| \sum_{i,j=1}^n (\xi_j)^{-2} c^j \mathcal{A}_i \mathcal{A}_j f(g) \mathcal{A}_i|_g \right\|. \end{aligned}$$

Above the vectors in the purple parts belong to $T_g(G)$, whereas the vectors in the green part belongs to $T_{\gamma_{g,\mathbf{c}}(t)}(G)$. Next, we write (51) as an SVD problem that involves the Hessian of f at g :

$$\begin{aligned} \mathcal{B}_d|_g &= \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| \nabla_{\mathbf{c}}^{[0]} \operatorname{d}f(g) \right\|_* \\ &= \operatorname{argmin}_{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}} \left\| (H^{[0]} f(g))(\mathbf{c}, \cdot) \right\|_*. \end{aligned} \tag{52}$$

Now identify the Hessian $H^{[0]}f(g)$ in the natural way to the associated linear map $A_g : T_g(G) \rightarrow T_g^*(G)$ by

$$A_g \mathbf{c} := (H^{[0]}f(g))(\mathbf{c}, \cdot).$$

Then by Euler-Lagrange, we have

$$A_g^* A_g \mathbf{c} = \lambda_{\min} \mathbf{c} \in T_g(G). \tag{53}$$

so we arrive at an SVD of A .

Remark 12. As the SVD of A^* coincides with the SVD of A , we may as well replace the choice $\nu = 0$ in (52) and (51) with our special choice $\nu = 1$. Recall also (19).

Remark 13. The matrix representation for (53) relative to the basis of left-invariant vector fields gets a bit involved if expressed in the left-invariant frame since the adjoint A^* depends on the choice of left-invariant metric. In our special case of interest (42), we set $\zeta = 1$ and get

$$M_\xi^2 \mathbf{H}^T M_\xi^2 \mathbf{H} \mathbf{c} = \lambda_{\min} \mathbf{c}$$

with $M_\xi = \text{diag}\{\xi^{-1}, \xi^{-1}, 1\}$ and with \mathbf{H} the matrix whose element H_j^i with row index i and column index j equals $H_j^i = \mathcal{A}^j \mathcal{A}_i f(g)$.

Inclusion of External Regularization

It is also possible to include (external) regularization. For this, we need to define neighbouring exponential curves.

Definition 6. Let $g, h \in G$. Let $t \geq 0$ and let $\mathbf{c} \in T_g(G)$:

$$\gamma_{g,\mathbf{c}}^h(t) := hg^{-1} \gamma_{g,\mathbf{c}}(t) = h \exp_G \left((L_{g^{-1}})_* \mathbf{c} t \right) \tag{54}$$

$$\mathcal{B}_d|_g := \underset{\substack{\mathbf{c} \in T_g(G) \\ \|\mathbf{c}\| = 1}}{\text{argmin}} \left\| \int_G K_\rho \left(h^{-1} g \right) \cdot (L_{gh^{-1}})_* \left(\nabla_{\bar{c}}^{[0]} \text{grad } f \right) (h) \, d\mu(h) \right\|. \tag{55}$$

where μ is the left-invariant Haar measure on $G = SE(3) = \mathbb{R}^3 \rtimes SO(3)$, with $\bar{c}(h) = (L_{hg^{-1}})_* \mathbf{c} \in T_h(G)$, and where K is an external regularization kernel with two regularization parameters $\rho = (\rho_S, \rho_A) \in (\mathbb{R}^+)^2$. Typically, it is the direct

product of an isotropic spatial Gaussian on \mathbb{R}^3 with spatial scale $\rho_S > 0$ and a heat kernel on $SO(3)$ with angular scale ρ_A ; for details and motivation, see Duits et al. (2016, ch:2.7).

Such a regularization will stabilize the best exponential curve fits, so that they become more adjacent with neighbouring exponential curve fits. Again the regularized problem is solved with an SVD with A^ρ :

$$(A_g^\rho)^* A_g^\rho \mathbf{c} = \lambda_{\min} \mathbf{c} \in T_g(G), \text{ with } A_g^\rho = (K_\rho * A.)(g),$$

which is the regularized version of A (with $A^\rho \rightarrow A$ as $\rho \downarrow 0$).

A Single Exponential Curve Fit Gives Rise to a Gauge Frame

Each local exponential curve fit at $g \in SE(d)$ to lifted data (orientation score) gives rise to a basis of local derivatives ('gauge frame'). This can be seen for $d = 2$ in Fig. 12. The general mathematical construction is explained in Duits et al. (2016, App.A, Thm. 7) and is highly beneficial in medical imaging applications (such as in vessel segmentation, see Zhang et al. (2016) for extensive comparisons to many other geometric and machine learning methods). For documented implementations of Gauge frames in $SE(d)$, for $d = 2, 3$, in *Mathematica*, see Martin and Duits (2017).

Overview of Image Analysis Applications for $G = SE(d)$

The analysis and computation of intensity variations in images plays a fundamental role in image processing. Here it is particularly useful to employ orientation lifts such as orientations scores (Duits et al. 2007; Bekkers et al. 2014; Bertalmío et al. 2019), continuous wavelet transforms (Citti and Sarti 2006; Sharma and Duits 2015; Siffre 2014), or orientation channel representations (Forssen 2004; Felsberg et al. 2006), to take advantage of the manifest disentanglement of local orientations in images to deal with complex structures such as crossings; recall Fig. 1.

For example, the crossing-preserving geometric analysis could be in solving PDE flows for enhancement (Janssen et al. 2018; Momayyez-Siahkal and Siddiqi 2009; Citti et al. 2016; Duits et al. 2013), denoising (Duits et al. 2019), regularization (Chambolle and Pock 2018), perception (Citti and Sarti 2006; Bertalmío et al. 2019) or segmentation (Zhang et al. 2016); for determining principal directions (Duits et al. 2016) (e.g. to steer PDEs or filters (Hannink et al. 2014)); or for defining geometric regularization priors in machine learning (Bekkers et al. 2018; Smets et al. 2020).

In the upcoming subsections, we go through some of the applications which find a direct application of the theory described in this chapter. Some algorithms based on the described differential geometric toolset on \mathbb{M}_d can be regarded as the natural generalization of classical geometric tools on \mathbb{R}^d . For example, the widely used Frangi vesselness filter (Frangi et al. 1998) is based on the analysis of the Hessian, which in our framework of lifted representations via orientation scores (Hannink

et al. 2014) is computed via an SVD of the Hessian induced by the Lie-Cartan connection with $\nu = 1$; recall section “[Straight Curve Fits](#)”.

More important, however, is that the proposed toolset enables the design of a completely new range of algorithms that enables analyses that are simply not possible by holding on to data representations on \mathbb{R}^d . These include globally optimal path optimization with an intrinsic (curvature penalizing) smoothness constraint via sub-Riemannian geometry (Bekkers et al. 2015; Duits et al. 2018; Chen 2016; Mirebeau and Portegies 2019; Franceschiello et al. 2019), which is the topic of Sect. “[Shortest Curve Application: Tracking of Blood Vessels](#)”; the direct computation of curvature and torsion of blood vessels for biomarker research without having to explicitly track/model the vessel trajectories (Bekkers et al. 2015), which is the topic of section “[Straight Curve Application: Biomarkers for Diabetes](#)”; and crossing-preserving, curvature-adaptive denoising schemes (Franken and Duits 2009; Duits et al. 2019; Smets et al. 2019), which is the topic of section “[Straight Curve Application: PDEs on \$\mathbb{M}_2\$ for Denoising](#)”.

What all of these applications have in common is that they either rely on ‘straight curves’ which are auto-parallel w.r.t. the Lie-Cartan connections or on ‘shortest curves’ which have parallel momentum (for $\nu = 1$) according to our main theorem, Theorem 1.

The differential-geometrical toolset described in this chapter can directly be translated to numerical schemes by working with discrete grids and finite difference stencils (Creusen et al. 2011) or via basis expansion methods such as spherical harmonics (Janssen et al. 2017; Reisert and Kiselev 2011; Skibbe and Reisert 2017) and B-splines (Bekkers et al. 2018) that allow for the computation of exact derivatives or via a mix of numerical and analytical schemes (Bekkers 2017; Zhang et al. 2016). Examples of the latter include the use of analytic approximations of sub-Riemannian distances (Sachkov 2011; Bekkers et al. 2015) in a clustering algorithm (Bekkers et al. 2017) or analytic solution approximations to left-invariant diffusion equations (Portegies et al. 2015) for smoothing or uncertainty analysis (Meesters et al. 2017). The interested reader is referred to Franken and Duits (2009), Janssen et al. (2017), Duits et al. (2016), Creusen et al. (2011), and Bekkers (2017) for algorithmic implementation details of the left-invariant derivatives for processing of orientation scores.

Shortest Curve Application: Tracking of Blood Vessels

Shortest path algorithms provide a robust way of extracting trajectories of blood vessels in medical images in a semi-automatic way. They rely on the specification of start and end points of the curves by a user, after which the algorithm computes the globally optimal geodesic connecting these points given a pre-computed metric. A fundamental problem in such algorithms is, however, that they have difficulties in tracking blood vessels through complex geometries and that they suffer from so-called short cuts in which the computed geodesics snap to parallel vessels or other interfering structures; see, e.g. Fig. 13. Via the computation of shortest paths in the

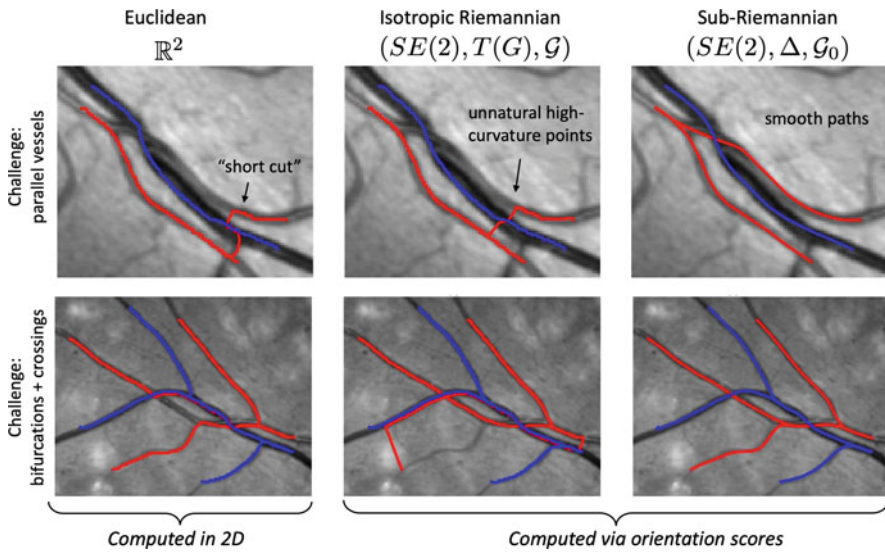


Fig. 13 Results of globally optimal data-adaptive geodesics computed in different metric tensor settings. Left column: Conventionally such shortest paths are computed based on 2D isotropic metrics. Such models suffer from short cuts (geodesics snap to other, typically parallel, dominant vessels) and often fail at crossings. Middle column: Shortest path computations using an isotropic metric in a lifted position-orientation space $\mathbb{M}_2 \equiv SE(2)$ reduce problems with crossings due to a disentanglement of local orientations, but the issue of short cuts remains as unnatural curves with high curvature points are still allowed. Right column: Both problem are solved by working with a sub-Riemannian metric on $SE(2)$ by which only natural curves are allowed in the lifted space (cf. Fig. 1). The right two columns show the 2D projections of geodesics in $SE(2)$. For further experiments on large datasets, see Bekkers et al. (2015, 2017)

lifted space \mathbb{M}_d using a sub-Riemannian geometry (cf. Fig. 1), we are able to solve such limitations of classical vessel tracking on \mathbb{R}^d .

In our approach for computing globally optimal sub-Riemannian distance minimizers between two points $g_0, g_1 \in SE(2)$, we consider the metric $d_{\mathcal{G}_0}$ of Eq. (34), which is defined using the SR metric tensor \mathcal{G}_0 given in (30) and which is based on a cost function $C : G \rightarrow \mathbb{R}^+$ that is derived from the orientation score. The cost function encourages curves to move over vessel regions (low cost) and penalizes moving over background regions (high cost). Such a cost can, for example, be derived from the orientation score via a vesselness measure (Hannink et al. 2014), a line-fidelity measure based on left-invariant derivatives (Bekkers et al. 2015) or gauge derivatives in $SE(2)$ (Duits et al. 2016; Zhang et al. 2016). The actual computation of the shortest paths then consists of (1) solving the SR eikonal equation in order to obtain a distance map from g_0 to any other point in $SE(2)$ and (2) perform gradient descent on the distance maps from g_1 back to g_0 to obtain the geodesic (cf. Theorem 1). The numerical computation of step 1 can, for example, be done in an iterative upwind scheme with left-invariant finite-difference stencils (Bekkers et al. 2015) or via very efficient fast-marching schemes (Mirebeau and

Table 1 Comparison of successful vessel extractions (Bekkers et al. 2017) via Riemannian geodesics using 2D isotropic metric tensors in the image domain, Riemannian geodesics in the lifted domain $SE(2)$ of orientation scores with spatially isotropic metric tensors and sub-Riemannian geodesics in $SE(2)$

Metric	Nr of successful vessel extractions
Riemannian \mathbb{R}^2	71.7% (132/184)
Riemannian $SE(2)$ - Eq. (33)	82.6% (152/184)
Sub-Riemannian $SE(2)$ - Eq. (34)	92.4% (170/184)

Portegies 2019; Mirebeau 2014) in which the sub-Riemannian metric tensor field is approximated with a highly anisotropic Riemannian metric tensor field (Sanguinetti et al. 2015).

Exemplary results are given in Fig. 13, and a quantitative evaluation of the benefit of a sub-Riemannian versus Riemannian metrics is given in Table 1. The principle that in a sub-Riemannian framework we only consider natural smooth paths, as illustrated in Fig. 1, leads to very clear improvements for vessel tracking. The method for computing such curvature-penalized data-adaptive SR geodesics generalizes well to other Lie groups G and has found several high-impact applications in medical image analysis. See, e.g. Bekkers et al. (2015) and Sanguinetti et al. (2015) for 2D vessel tracking via SR geodesics in $G = SE(2)$. See also Mashtakov et al. (2017) for vessel tracking in retinal images defined on the two-sphere $S^2 = SO(3)/SO(2)$ via SR geodesics in $SO(3)$.

Shortest Curve Applications: Geodesic Vessel and Fibre Tracking in \mathbb{M}_3

In Duits et al. (2018), the anisotropic and sub-Riemannian geodesic tracking theory was developed and extended with more general Finslerian models such as the one given in (46). These Finsler models were called variants of the ‘Reeds-Shepp car model’. Some of these models turn off the reverse gear of the car and tackle the problem of cusps (recall Fig. 9) that can appear in spatial projections of sub-Riemannian geodesics. In Duits et al. (2018), the underlying theory was also extended to 3D (or more precisely to the five-dimensional homogeneous space \mathbb{M}_3 of positions and orientations). It has led to efficient perceptual grouping methods (Bekkers et al. 2017) where vascular trees are constructed from the separate geodesic tracts following 3D blood vessels. When extending the models from the 3D manifold \mathbb{M}_2 to the 5D manifold \mathbb{M}_3 , it is crucial to rely on fast anisotropic fast-marching methods (Mirebeau 2018) that do approximate the sub-Riemannian setting (with infinite anisotropy) reasonably well, as shown by comparison (Duits et al. 2018) to the exact sub-Riemannian geodesics in \mathbb{M}_3 derived in Duits et al. (2016). The idea of using highly anisotropic, advanced, fast-marching methods by Mirebeau (2014, 2018) to approximate sub-Riemannian geodesics was proposed by Sanguinetti et al. (2015) on \mathbb{M}_2 , where numerical comparisons reveal enormous speedups (compared to iterative PDE-techniques in Bekkers et al. 2015; Portegies 2018) while maintaining a neglectable loss of accuracy. It was employed for crossing-preserving fibre tracking (Portegies 2018; Duits et al. 2018)

and for crossing-preserving structural connectivity measures (Portegies et al. 2019) (between anatomical regions of interest) in DW-MRI data of the brain (in response to earlier work by Pechaud et al. (2009)).

Since our previous works (Bekkers et al. 2015; Duits et al. 2018) mainly concentrated on tracking of 2D blood vessels in optical images and 3D neural fibre tracking in DW-MRI (Portegies et al. 2019; Duits et al. 2018), we show a 3D vessel tracking experiment in this book chapter. See Fig. 14. Again we recognize the benefit of the asymmetric version (46) of the 3D sub-Riemannian geometrical model on the lifted space \mathbb{M}_3 over the corresponding isotropic Riemannian geodesic model on \mathbb{M}_3 and over the corresponding geodesic model on \mathbb{R}^3 . The new model does not suffer from nearby elongated structures, does not take wrong exits (as shown in Duits et al. 2018), deals with bifurcations by ‘key points’ (in place rotations) and produces less oscillatory tracts due to the sub-Riemannian geometry (which does not allow for direct sideward motions in contrast to the blue ‘shaky’ tract in Fig. 14).

Furthermore, we note that non-data-adaptive sub-Riemannian distances in $SE(d)$ can be efficiently computed using analytic approximations (Duits and Franken 2011; Portegies et al. 2015; ter Elst and Robinson 1998) which can be used in real-time clustering of local orientations for perceptual grouping of blood vessels (Bekkers et al. 2017) or in morphological convolutions in equivariant deep learning (Smets et al. 2020).

Finally, for detailed evaluations and experiments of geodesic tracking in \mathbb{M}_3 , we refer to Portegies et al. (2019) and Duits et al. (2018) where the experiments are focused on fibre tracking in DW-MRI.

Straight Curve Application: Biomarkers for Diabetes

The total amount of curvature/torsion of blood vessels, which is often summarized in a single tortuosity measure, is associated with severity of several systematic diseases such as diabetes and hypertension (Bekkers et al. 2015; Bekkers 2017; Zhang et al. 2017; Zhu et al. 2016, 2020). Reliable and automatic quantification of tortuosity is therefore a high value aid in the automatic early diagnosis of such systemic diseases and in the study of disease progression via large-scale cohorts. The theory of exponential curve fits in $SE(d)$ enables a unique approach to the quantification of tortuosity in retinal images, which is robust, reliable and fast (Bekkers et al. 2015). Retinal images are obtained by optical devices (Bekkers 2017) and therefore provide an easy noninvasive way to image the quality of blood vessels.

As described in section “Straight Curve Fits”, it is possible to locally fit exponential curves (see, e.g. Figs. 12 and 15) to the orientation score data $U := \mathcal{W}_\psi f : \mathbb{M}_2 \rightarrow \mathbb{R}$ of a retinal image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ via an SVD of left-invariant Hessian $\nabla^{[1],*}dU$. This is akin to fitting straight lines to local intensity patterns in images, underlying classical vesselness measures in Frangi et al. (1998). The exponential curves in $SE(d)$ are equally ‘straight’, but now with respect to the torqued geometry modelled by Lie-Cartan connection $\nabla^{[1]}$, recall Theorem 1. This torqued geometry is also visible in an orientation score; recall Fig. 1 and see Fig. 7.

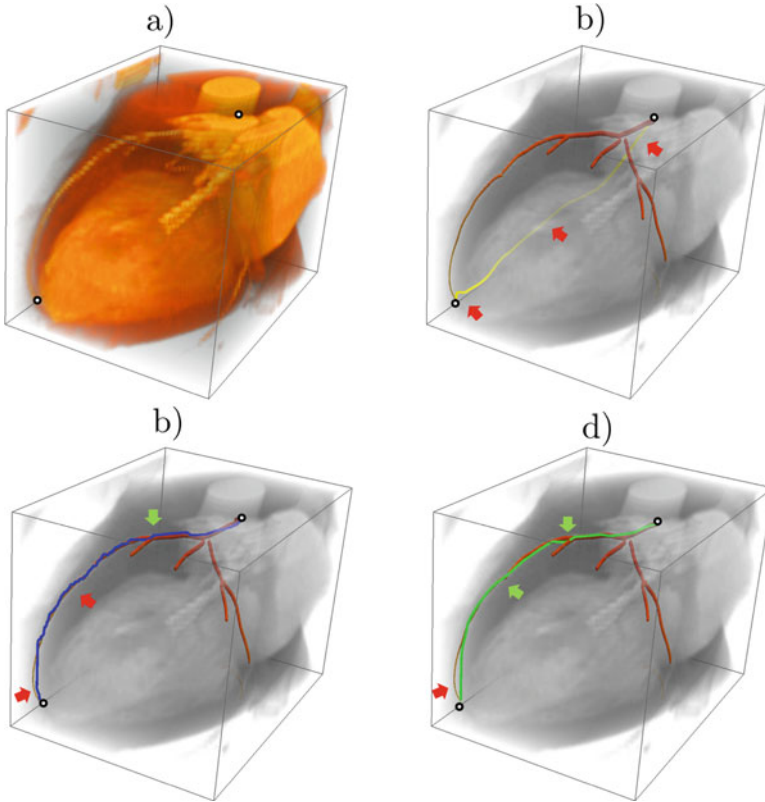


Fig. 14 Tracking of coronary arteries in 3D-X-ray: **(a)** test dataset with two boundary points $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^3$, **(b)** geodesic tracking result (yellow) that is far from ground truth (red) when applying standard geodesic tracking (Caselles et al. 1997) on \mathbb{R}^3 without lifting to \mathbb{M}_3 , **(c)** geodesic tracking result (blue) when applying geodesic tracking in \mathbb{M}_3 using the isotropic Riemannian model (i.e. using Finsler function $\mathcal{F}(\dot{\mathbf{x}}, \dot{\mathbf{n}}) = \sqrt{\xi^2 \|\dot{\mathbf{x}}\|^2 + \|\dot{\mathbf{n}}\|^2}$ with $\xi=0.1$), **(d)** geodesic tracking result when applying geodesic tracking in \mathbb{M}_3 using the sub-Riemannian model (i.e. using asymmetric Finsler function \mathcal{F}_0^+ given by (46) again with $\xi = 0.1$). The spherical parts of the boundary conditions $\mathbf{p}_1 = (\mathbf{x}_1, \mathbf{n}_1)$ and $\mathbf{p}_2 = (\mathbf{x}_2, \mathbf{n}_2)$ in (45) are automatically optimized by checking for the ‘first passing front’, i.e. adjust the source set in eikonal PDE system (40) in Theorem 1 from singleton $\{e\}$ to the set $\mathcal{S} = \{(\mathbf{x}_0, \mathbf{n}) \mid \mathbf{n} \in S^2\}$ and select minimal $\mathbf{n}_1 = \operatorname{argmin}_{\mathbf{n} \in S^2} W(\mathbf{x}_1, \mathbf{n})$ prior to backtracking (39)

These ‘straight’ curves have constant velocity components w.r.t. the left-invariant frame $\{\mathcal{A}\}_{i=1}^n$, and their projections to \mathbb{R}^d are circles/spirals whose curvature κ can directly be computed, e.g. in $SE(2)$, one has $\kappa = \frac{c^3 \operatorname{sign} c^1}{\sqrt{|c^1|^2 + |c^3|^2}}$.

Akin to the vessel enhancement techniques via orientation scores of Zhang et al. (2016) and Hannink et al. (2014), a confidence measure for the presence of a line structure can be extracted from the left-invariant Hessian Franken et al. (2007), Franken and Duits (2009) and Bekkers et al. (2015). Together, the confidence and

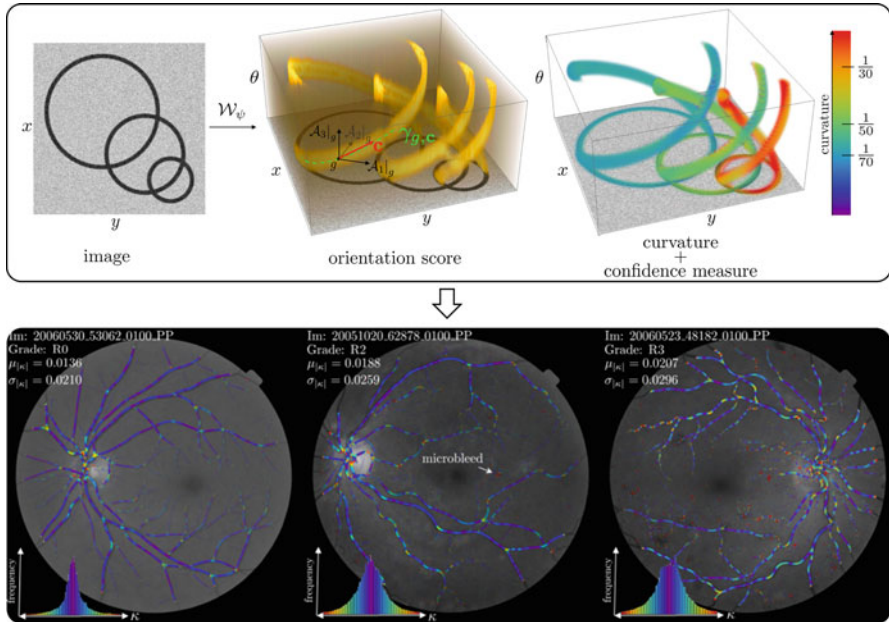


Fig. 15 Top row: Via exponential curve fits in orientation scores (cf. section “Straight Curve Fits”), we are able to locally analyse line structures and compute their corresponding curvature values, as well as assigning confidence scores at each position and orientation. In the right most figure, curvature is colour coded, and confidence is encoded with opacity. Bottom row: confidence and curvature projected to the 2D plane and visualized as in an overlay on top of the original input image. From these summarizing statistics such as the mean and standard deviation of absolute curvature can be computed, which can be used as biomarkers for diabetes and hypertension

curvature measure can be used to obtain summarizing statistics for the amount of tortuosity of blood vessels in medical images, as is illustrated in Fig. 15. Such tortuosity measures are significantly associated with severity of diabetes and hypertension on large-scale clinical datasets with retinal images (Bekkers et al. 2015; Bekkers 2017; Zhang et al. 2017; Zhu et al. 2016, 2020). For quantification of blood vessel tortuosity in 3D medical image data, see Janssen et al. (2017).

Straight Curve Application: PDEs on \mathbb{M}_2 for Denoising

Two key ideas have greatly improved techniques for image enhancement and denoising: the lifting of image data to multi-orientation distributions (e.g. orientation scores Duits 2005) and the application of nonlinear PDEs such as total variation flow (TVF) and mean curvature flow (MCF). These two ideas were recently combined by Chambolle and Pock (for TVF) (2018) and Citti and Sarti (2006) (for MCF) for 2D images.

In our recent works Duits et al. (2019) and Smets et al. (2019), these approaches were extended to enhance and denoise images of arbitrary dimension. The TV flows and MC flows on \mathbb{M}_d showed best results when using locally adaptive frames of a specific type, namely, these locally adaptive frames that were computed via the best-exponential curve fit procedure (i.e. the ‘straight curve’ fit in the torqued and curved space $SE(d)$; recall Theorem 1 and Figs. 1 and 7) explained in section “[Straight Curve Fits](#)”. Then the standard procedure mentioned in section “[A Single Exponential Curve Fit Gives Rise to a Gauge Frame](#)” to compute the induced locally adaptive frame (‘gauge frame’) $\{\mathcal{B}_1, \dots, \mathcal{B}_n\}$ is applied. The principle direction \mathcal{B}_d tangent to the exponential curve is computed as the eigenvector with smallest eigenvalue in the SVD of the Hessian induced by the Lie-Cartan connection with $\nu = 1$; recall Definition 6.

For an illustration, recall Fig. 12 where $d = 2$ and $n = 3$.

In this section, we constrain ourselves to $d = 2$, and we shall summarize the MCF and TVF PDEs on $SE(2)$ (for crossing-preserving flows via invertible orientation scores, recall Fig. 1) and highlight a denoising result, where we compare to a popular denoising method called ‘Block Matching 3D’ (BM3D) (Lebrun 2012; Dabov et al. 2007).

The PDE system for MCF and TVF on $\mathbb{M}_2 = SE(2)$ via the gauge frame $\{\mathcal{B}_1, \dots, \mathcal{B}_3\}$ is best expressed in this frame and is given by

$$\begin{cases} \frac{\partial W}{\partial t}(g, t) = \|\nabla W(g, t)\|^a \sum_{i=1}^3 \mathcal{B}_i \left(\frac{\mathcal{B}_i W(\cdot, t)}{\|\nabla W(\cdot, t)\|} \right)(g), & g \in SE(2), t \geq 0, \\ W(g, 0) = U(g), & g \in SE(2), \end{cases} \tag{56}$$

with parameter $a \in \{0, 1\}$, where we have a total variation flow (TVF) if $a = 0$ and a mean curvature flow (MCF) if $a = 1$. We denote the operator that maps the orientation score $U(\cdot)$ to its denoised version $W(\cdot, t)$ by Φ_t :

$$W(g, t) = (\Phi_t(U))(g), \text{ for all } g = (\mathbf{x}, \theta) \in SE(2), t \geq 0,$$

where we use standard identification $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$ with the corresponding counterclockwise planar rotation about angle θ .

The initial condition U for our TVF/MCF-PDE (56) is set by an orientation score (Duits et al. 2007; Bekkers et al. 2014) of image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ given by

$$U(\mathbf{x}, \theta) := \mathcal{W}_\psi f(\mathbf{x}, \theta) = (\psi_\theta \star f)(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2, \theta \in \mathbb{R}/(2\pi\mathbb{Z}).$$

where \star denotes correlation and ψ_θ is the rotated wavelet aligned with $(\cos \theta, \sin \theta) \in S^1$. For ψ , we use a cake wavelet (Duits et al. 2007; Bekkers et al. 2014) $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$ with standard settings (Martin and Duits 2017). Then we compute

$$f \mapsto \mathcal{W}_\psi f \mapsto \Phi_t(\mathcal{W}_\psi f)(\cdot, \cdot) \mapsto f_t(\cdot) := \int_{-\pi}^{\pi} \Phi_t(\mathcal{W}_\psi f)(\cdot, \theta) d\theta. \quad (57)$$

for $t \geq 0$. The cake wavelets allow us to reconstruct by integration over S^1 only (Duits et al. 2007; Bekkers et al. 2014). By the invertibility of the orientation score, one therefore has $f = f_0$ so due to this reconstruction property the flows depart from the original image at $t = 0$.

Now that the PDEs are set for MCF and TVF and the corresponding image regularization operators $f \mapsto f_t$ via invertible orientation scores are set (by Eq. (57) and (56)), we conclude with a denoising experiment. End times $t > 0$ are chosen such that relative \mathbb{L}_2 -error between the original image and its denoised image is minimal.

We test the effect of MCF and TVF on two images polluted with correlated noise: the (monochrome) Mona Lisa and an electron microscopy image of collagen. We compare the performance (in terms of peak signal-to-noise ratio) against the BM3D method; see Table 2 for the PSNR values and Fig. 16 for a qualitative comparison.

As confirmed by Table 2 and Fig. 16, we observe the following:

- Denoising via orientation scores is beneficial over direct image denoising. For PDE-based image processing, this was already done in Franken and Duits (2009) and by others in Citti et al. (2016), Citti and Sarti (2006), Baspinar (2018), Boscain et al. (2018), and Bertalmío et al. (2019) performing left-invariant PDE-based image processing via ‘orientation liftings’ (expanding the image domain to \mathbb{M}_d). However, our experiments where we use (data-driven) TVF and MCF (56) on \mathbb{M}_2 now show that we considerably improve quantitative results

Table 2 Comparing peak signal-to-noise ratio (dB) for the gauge MCF and TVF methods against BM3D (higher is better)

Gaussian noise	Collagen	Mona Lisa
Noisy image	14.1	14.1
Perona-Malik	20.1	20.5
BM3D	23.1	23.9
Left inv. MCF	21.7	23.3
Gauge MCF	21.7	23.7
Left inv. TVF	22.4	26.0
Gauge TVF	23.0	26.1
Correlated noise	Collagen	Mona Lisa
Noisy image	23.9	23.9
Perona-Malik	24.2	25.1
BM3D	24.0	26.3
Left inv. MCF	23.8	26.2
Gauge MCF	23.9	26.2
Left inv. TVF	24.7	26.8
Gauge TVF	24.9	26.9

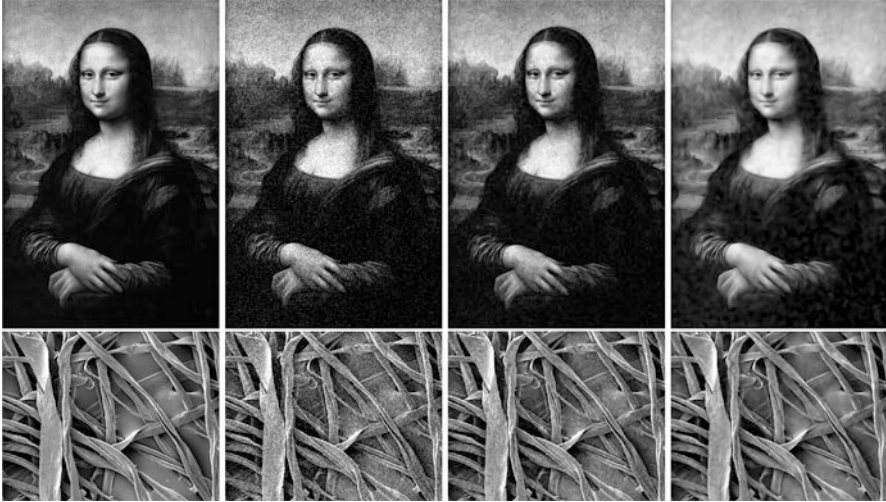


Fig. 16 Comparing Gauge TVF with coherence enhancement and BM3D against correlated noise. Top row, from left to right: (1) original image, (2) original image polluted with correlated Gaussian noise, (3) denoising result using the BM3D method, (4) denoising result using the TVF method via invertible orientation scores given by (57) relying on PDE (56) with $a = 0$. Bottom row, the same as the top row but now applied on a different image containing collagen fibres. The standard deviation for BM3D and evolution time for TVF were adjusted to reach optimal \mathbb{L}_2 error; see Smets et al. (2019) for details

in comparison to a general (not necessarily PDE-based) well-performing image denoising method such as BM3D (Lebrun 2012; Dabov et al. 2007).

- Best performances are obtained by the Gauge TVF method, i.e. the method applying (57) with Φ_t given by (56) with $a = 0$.
- Using the locally adaptive frame $\{\mathcal{B}_i\}$ in (56) increases the performances over their ‘normal left-invariant counterparts’. With the normal left-invariant counterparts, we mean (57) with Φ_t given by a PDE system on $\mathbb{M}_2 \equiv SE(2)$ that arises from (56) replacing each \mathcal{B}_i in (56) by \mathcal{A}_i :

$$\begin{cases} \frac{\partial W}{\partial t}(\mathbf{p}, t) = \|\nabla W(\mathbf{p}, t)\|^a \sum_{i=1}^3 \mathcal{A}_i \left(\frac{\mathcal{A}_i W(\cdot, t)}{\|\nabla W(\cdot, t)\|} \right) (\mathbf{p}), & \mathbf{p} \in \mathbb{M}_2, t \geq 0, \\ W(\mathbf{p}, 0) = U(\mathbf{p}), & \mathbf{p} \in \mathbb{M}_2, \end{cases} \tag{58}$$

as done also in Citti et al. (2016) and Chambolle and Pock (2011). Gauge TVF performs better than normal left-invariant TVF. Gauge MCF performs better than normal left-invariant MCF via invertible orientation scores (57).

These observations are also supported by much more experiments with both quantitative and qualitative comparisons for the case $d = 2$ and $d = 3$; see Smets et al. (2019). Regarding related works and experiments via crossing-preserving diffusions via invertible orientation scores, we refer to Franken and Duits (2009) ($d = 2$) and Janssen et al. (2018) ($d = 3$).

In Smets et al. (2019), we have compared the (crossing-preserving) TVF and MCF PDE flows via invertible orientation scores to (crossing-preserving) nonlinear diffusions via invertible orientation scores. In general, better results are obtained by the MCF and TVF approach than with nonlinear diffusion (Perona and Malik 1990, coherence enhancing diffusion (Weickert 1999)). However, edge-enhancing diffusion techniques (Fabbrini et al. 2013) via invertible orientation scores could advocate otherwise and are left for future work.

Straight Curve Application: PDEs on \mathbb{M}_3 for Denoising FODFs in DW-MRI

In this subsection, we briefly highlight the extensions of the TVF and MCF denoising methods from three-dimensional manifold \mathbb{M}_2 toward five-dimensional manifold \mathbb{M}_3 .

Essentially, the MCF flows and TVF flows given in (58) are generalized to \mathbb{M}_3 by using the left-invariant vector fields on \mathbb{M}_3 instead of the left-invariant vector fields on \mathbb{M}_2 . In our experiments, we optimized the stopping time of the evolutions to get a denoised distribution on \mathbb{M}_3 . For details, see Smets et al. (2019). In Smets et al. (2019) (crossing-preserving) TVF and MCF PDE flows on the five-dimensional manifold \mathbb{M}_3 are compared to nonlinear diffusion methods on this manifold such as:

- Crossing-preserving versions (Creusen et al. 2013) of Perona and Malik (PM) (Perona and Malik 1990) diffusions
- Crossing-preserving versions (Duits and Franken 2011; Duits et al. 2013) of coherence enhancing diffusion (CED)

This has been applied to crossing-preserving enhancement and denoising of diffusion-weighted MRI (DW-MRI) data, where *fibre orientation density functions* (FODF), cf. Tournier et al. (2007) and Descoteaux et al. (2009), are positive, real-valued functions defined on the five-dimensional space \mathbb{M}_3 that are similar to orientation scores of 3D image data. To see the similarity, compare Fig. 17 to the middle column in Fig. 3. So in this application we only rely on the right part of our commutative diagram in Fig. 2.

Remark 14 (DW-MRI: Application Background). The idea of diffusion-weighted MRI is to measure angular diffusivity profiles of water molecules that are generally believed to follow the biological fibres in brain white matter. As such, it provides a noninvasive way to image the structural connectivity between anatomical regions in the brain. This is important for surgical planning. For example, identifying the optic radiation bundle is important as it is responsible for the visual sight of a patient. In case of severe epilepsy, surgery may be applied (e.g. a temporal lobe resection),

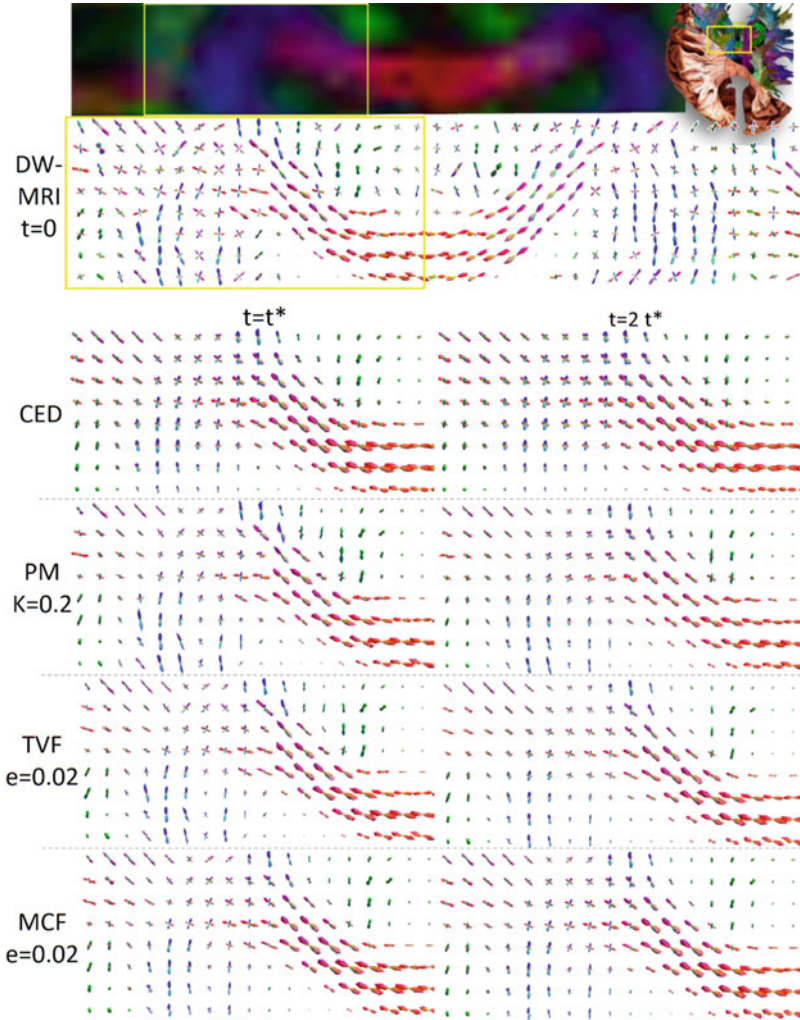


Fig. 17 Qualitative comparison of denoising a FODF obtained by (CSD) (Tournier et al. 2007; Descoteaux et al. 2009) from a standard DW-MRI dataset (with $b = 1000s/mm^2$ and 54 gradient directions). For the CSD, we used up to eighth-order spherical harmonics, and the FODF is then spherically sampled on a tessellation of the icosahedron with 162 orientations. Image is taken from our previous journal article (Smets et al. 2019). For details on this qualitative DW-MRI experiment and related quantitative DW-MRI denoising experiments, see the works by St.-Ongé et al. (2019) and Smets et al. (2019)

where surgeons should not damage the optic radiation bundle as this can lead to a reduction of visual sight. Left-invariant PDE evolutions (such as diffusions) on \mathbb{M}_3 discussed in Duits and Franken (2011), Portegies (2018), Reisert and Kiselev (2011), Momayyez-Siahkhal and Siddiqi (2009), and Duits et al. (2013) are very

beneficial for identifying such bundles, as shown by Meesters et al. (2017) and more generally by Prckovska et al. (2015).

As we can see in Fig. 17, the FODF obtained from raw DW-MRI data via an effective and widely used method CSD produces a lot of spurious peaks in the spatial field of angular distributions that are not well aligned/supported by neighbouring peaks and one needs ‘contextual processing’ (Prckovska et al. 2015; Momayyez-Siahkal and Siddiqi 2009; Reisert and Kiselev 2011) to identify large bundles (Portegies et al. 2015; Meesters et al. 2017) in a stable way. Here we observe that crossing-preserving MCF and TVF on \mathbb{M}_3 better preserve crossings and bundle boundaries than diffusion methods do. For detailed evaluations, see Smets et al. (2019) and St Onge et al. (2019).

Straight Curve Application: PDEs on \mathbb{M}_3 for Denoising 3D X-Ray Data

Denoising of 3D X-ray data is important as reduction of acquisition time and radiation dose typically leads to noisy X-ray images. In Janssen et al. (2018), denoising experiments are provided with crossing-preserving nonlinear diffusions on \mathbb{M}_3 via invertible orientation scores of 3D X-ray data.

These tests do follow the full commutative diagram in Fig. 4 and applied denoising as depicted in the bottom row of Fig. 3 and provide the \mathbb{M}_3 -analogue of (57):

$$f \mapsto \mathcal{W}_\psi f \mapsto \Phi_t(\mathcal{W}_\psi f)(\cdot, \cdot) \mapsto f_t(\cdot) := \int_{S^{d-1}} \Phi_t(\mathcal{W}_\psi f)(\cdot, \mathbf{n}) \, d\sigma(\mathbf{n}), \quad (59)$$

but then with $\Phi_t(U) = W(\cdot, t)$ a nonlinear diffusion process described in gauge frames (relying on an SVD of the Hessian of the orientation scores as explained in section “Exponential Curve Fits of the Second Order Are Found by SVD of the Hessian”) stopped at optimal time $t > 0$. For details, see Janssen et al. (2018, ch:6.1.2). The preservation of complex structures in vasculature is remarkable; see Fig. 18. For qualitative and quantitative comparisons against many other nonlinear diffusion methods, we refer to the work by Janssen et al. (2018).

Conclusion

Geometric processing of multi-feature image representations on a Lie group G requires us to ‘connect’ different tangent spaces in the tangent bundle $T(G)$ by a connection. To this end, we studied all Lie-Cartan connections $\nabla^{[\nu]}$ parameterized by $\nu \in \mathbb{R}$. This holds in particular for our case of interest where $G = SE(d)$ (or more precisely the Lie group quotient \mathbb{M}_d) and where the score is an orientation score. It turned out by our Theorem 1 that the case $\nu = 1$ is the best choice; shortest curves have parallel momentum, whereas straight curves have parallel velocity as intuitively illustrated in Fig. 7. This connection does have torsion with constant

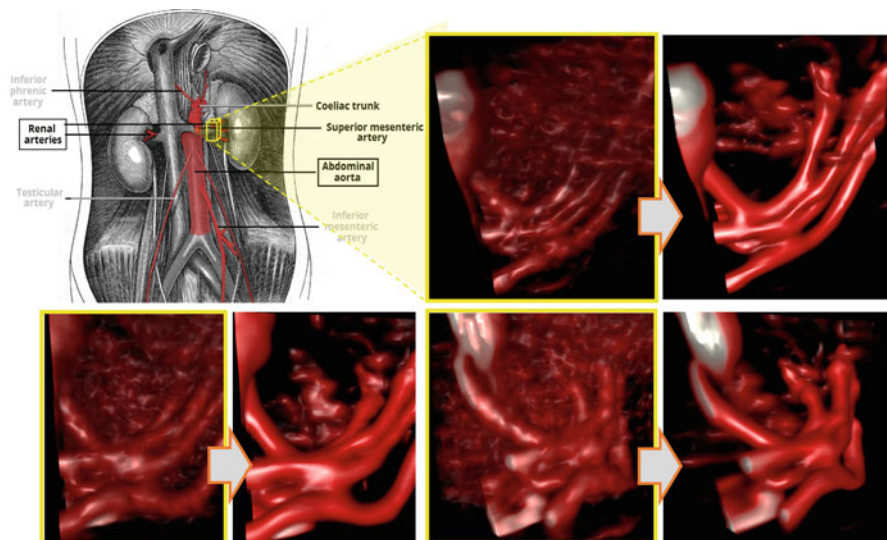


Fig. 18 3D X-ray image of renal arteries. Three view points on the same scene. Input image in yellow frame, output of coherence enhancing diffusion via 3D-orientation scores (CEDOS: Eq. (59)) for a fixed stopping time. For details and comparisons to other methods such as coherence enhancing diffusion (Weickert 1999) acting directly in the image domain, see Janssen et al. (2018)

coefficients relative to the left-invariant frame and co-frame as shown in Lemma 3. It reflects the torsion visible in the domain of an orientation score; see Fig. 1.

We studied the shortest curves in $\mathbb{M}_2 \equiv SE(2)$ for different choices of metric tensor fields (or more general: Finsler functions) and computed the corresponding spheres in \mathbb{M}_2 in section “[The Metric Models on \$\mathbb{M}_d\$: Shortest Curves and Spheres](#)”. Recall Fig. 10, where the spheres were computed via the geodesic wavefront propagation technique explained in Theorem 1. Such geodesic wavefront propagations also allow for data-driven versions (via the external cost C which can be adapted to the orientation score). The major benefit of geodesic wavefront propagation in the orientation score domain \mathbb{M}_d over geodesic wavefront propagation in the image domain \mathbb{R}^d is that fronts do not leak at crossings as illustrated in Fig. 8. This explains the clear advantage for subsequent geodesic tracking (via the steepest descent in Theorem 1) in the tracking of blood vessels presented in section “[Shortest Curve Application: Tracking of Blood Vessels](#)”. Furthermore, we show best results are obtained by the sub-Riemannian model rather than the isotropic Riemannian model. Recall Fig. 13.

We studied the straight curves (exponential curves) in $\mathbb{M}_2 \equiv SE(2)$ in section “[Straight Curve Fits](#)”. Again we presented data-driven versions by presenting an exponential curve fit theory in section “[Straight Curve Fits](#)” that we employed for biomarkers for diabetes in retinal imaging in section “[Straight Curve Application: Biomarkers for Diabetes](#)” and for improved data-driven crossing-preserving denoising PDEs in section “[Straight Curve Application: PDEs on \$\mathbb{M}_2\$ for Denoising](#)”.

Summarizing: we conclude from Theorem 1 and the experiments in section “[Overview of Image Analysis Applications for \$G = SE\(d\)\$](#) ” that the Lie-Cartan connection for $\nu = 1$ is the best choice for geometric multi-orientation image processing, both for crossing-preserving geodesic tracking and for crossing-preserving denoising.

Acknowledgments Etienne St-Onge and Maxime Descoteaux (Sherbrooke Connectivity Imaging Lab, Canada) are gratefully acknowledged for their support in the comparison depicted in Fig. 17. This DW-MRI application is only briefly highlighted in this theoretical overview article, and it is worked out much more profoundly in joint papers with part of the authors of the present paper and them; see previous ISMRM conference article by St.-Onge et al. (2019) and previous JMIV article by Smets et al. (2019).

Former colleague Michiel Janssen (TU/e, Netherlands) and Javier Olivan Bescos (Philips, Netherlands) are gratefully acknowledged for implementing and visualizing data-driven crossing-preserving diffusions on 3D X-ray data shown in Fig. 18. This 3D X-ray application and the underlying invertible orientation theory is only briefly highlighted in this theoretical overview article, and it is worked out much more profoundly in a JMIV article by Michiel Janssen et al. (2018).

The research leading to the results of this article has received funding from the European Research Council under the European Community’s 7th Framework Programme (FP7/20072014)/ERC grant agreement No. 335555 (Lie Analysis).

We gratefully acknowledge the following institutions for their financial support: the Dutch Foundation of Science NWO Talent Programmes VICI 2020 Exact Sciences (Duits, Geometric learning for Image Analysis, V.I.C. 202–031), VENI 2019 Applied and Technical Sciences (Bekkers, Context Aware AI, VI nr.17290) and the European Research Council under EC’s 7th Framework Progr. (Duits, Lie Analysis, nr.335555).

Appendix A: Hamiltonian Flow of the Left-Invariant (Sub-)Riemannian Geodesic Problem on Lie Group G

The family of all geodesics $\gamma(t)$ augmented to $\mathbf{v}(t) = (\gamma(t), \lambda(t))$ with their momentum representation $\lambda(t) = \sum_{i=1}^n \lambda_i(t) \omega^i \Big|_{\gamma(t)}$ along the geodesic are flow-lines of a so-called Hamiltonian flow on the co-tangent bundle $T^*(G)$. Controlling the Hamiltonian flow means controlling the complete family of all geodesics (minimal distance curves) together. Next, we explain the concept of Hamiltonian flows and derive the canonical Hamiltonian equations associated to the left-invariant Riemannian and sub-Riemannian problem of interest.

To a Hamiltonian function \mathfrak{h}

$$T^*(G) \ni (g, \lambda) \mapsto \mathfrak{h}(g, \lambda) \in \mathbb{R}^+$$

one associates a Hamiltonian vector field $\overrightarrow{\mathfrak{h}}$ (or ‘Hamiltonian lift’) in the co-tangent bundle. It is determined via the fundamental symplectic form that is given by

$$\sigma = \sum_{i=1}^n \omega^i \wedge \overrightarrow{d}\lambda_i,$$

where $\overrightarrow{d}\lambda_i$ is defined by $\langle \overrightarrow{d}\lambda_i, \partial_{\lambda_j} \rangle = \delta_j^i$, by means of

$$\forall_{V=(\dot{g}, \dot{\lambda}) \in T_g(G) \times T(T_g^*(G))} : \sigma(\overrightarrow{h}(g, \lambda), V) = \langle d\mathfrak{h}(g, \lambda), V \rangle. \tag{60}$$

Remark 15 (background on Hamiltonian lifts). A direct consequence of (60) is that along the flowlines of the Hamiltonian flow, the Hamiltonian is preserved (take $V = \overrightarrow{h}$) and

$$\frac{d}{dt} \mathfrak{h}(\mathbf{v}(t)) = \sigma(\overrightarrow{h}(\mathbf{v}(t)), \overrightarrow{h}(\mathbf{v}(t))) = 0, \text{ with } \mathbf{v}(t) = (\gamma(t), \lambda(t)),$$

Furthermore, the lifting of a Hamiltonian \mathfrak{h} to its Hamiltonian lift \overrightarrow{h} is a Lie algebra isomorphism (Agrachev and Sachkov 2004):

$$\overrightarrow{\{h_1, h_2\}} = [\overrightarrow{h}_1, \overrightarrow{h}_2] \tag{61}$$

where $\{\cdot, \cdot\}$ denotes Poisson brackets and $[\cdot, \cdot]$ denotes the usual Lie bracket of vector fields. In the left-invariant (co)-frames, Poisson brackets are expressed as

$$\{g, f\} = \sum_{i=1}^n (\mathcal{A}_i f) \frac{\partial g}{\partial \lambda_i} - \frac{\partial f}{\partial \lambda_i} (\mathcal{A}_i g), \tag{62}$$

but this may also be expressed in canonical coordinates (Agrachev and Sachkov 2004, eq.11.21).

Remark 16 (simple example of Hamiltonian lifts on $T^(\mathbb{R})$).* We set $\sigma = dx \wedge d\lambda$. We set $\overrightarrow{h} = h^1 \partial_x + h^2 \partial_\lambda$. Then from (60) one can deduce the following standard canonical equations:

$$\overrightarrow{h} = \frac{\partial \mathfrak{h}}{\partial \lambda} \partial_x - \frac{\partial \mathfrak{h}}{\partial x} \partial_\lambda \Rightarrow \dot{x} \partial_x + \dot{\lambda} \partial_\lambda = \dot{\mathbf{v}} = \overrightarrow{h}(\mathbf{v}) \Leftrightarrow \begin{cases} \dot{x} = \frac{\partial \mathfrak{h}}{\partial \lambda} \text{ (horizontal part),} \\ \dot{\lambda} = -\frac{\partial \mathfrak{h}}{\partial x} \text{ (vertical part).} \end{cases}$$

Generalizing the above example, the next theorem provides the Hamiltonian flows for the left-invariant Riemannian and sub-Riemannian problem on G .

Theorem 2. *The Hamiltonian on Riemannian manifold (G, \mathcal{G}) , with left-invariant metric tensor field \mathcal{G} given by (26), equals*

$$\mathfrak{h} = \frac{1}{2} \sum_{i=1}^n \lambda^i \lambda_i = \frac{1}{2} \sum_{i,j=1}^n g^{ij} \lambda_i \lambda_j \tag{63}$$

and the corresponding Hamiltonian flow (generated by the Hamiltonian vector field $\vec{\mathfrak{h}}$) can be written as (recall the definition of linear map $\tilde{\mathcal{G}}$ (27))

$$\begin{aligned} \dot{\mathbf{v}} = \vec{\mathfrak{h}}(\mathbf{v}) &\Leftrightarrow \begin{cases} \tilde{\mathcal{G}}^{-1} \lambda = \dot{\gamma} & \text{(horizontal part)} \\ \nabla_{\dot{\gamma}}^{[1],*} \lambda = 0 & \text{(vertical part)} \end{cases} \\ &\Leftrightarrow \begin{cases} \dot{\gamma}^i = u^i = \lambda^i := \sum_{j=1}^n g^{ij} \lambda_j & \text{(horizontal part)} \\ \dot{\lambda}_i = \{\mathfrak{h}, \lambda_i\} = - \sum_{j,k=1}^n c_{ij}^k \lambda_k u^j & \text{(vertical part)} \end{cases} \end{aligned} \tag{64}$$

with velocity controls $u^i := \dot{\gamma}^i = \langle \omega^i|_{\gamma(\cdot)}, \dot{\gamma} \rangle$ and $\mathbf{v}(t) = (\gamma(t), \lambda(t))$ a curve in the co-tangent bundle $T^*(G)$ where the geodesic $\gamma(t) \in G$ and the momentum along the geodesic $\lambda(t) \in T_{\gamma(t)}^*(G)$, and with $\{\cdot, \cdot\}$ denoting Poisson brackets, recall (62). The Hamiltonian on sub-Riemannian manifold $(G, \Delta = \text{span}\{\mathcal{A}_j\}_{j \in I}, \mathcal{G}_0)$ equals

$$\mathfrak{h} = \frac{1}{2} \sum_{i \in I} \lambda^i \lambda_i = \frac{1}{2} \sum_{i,j \in I} g^{ij} \lambda_j \lambda_i \tag{65}$$

and the Hamiltonian flow can be written as

$$\begin{aligned} \dot{\mathbf{v}} = \vec{\mathfrak{h}}(\mathbf{v}) &\Leftrightarrow \begin{cases} \tilde{\mathcal{G}}_0^{-1} P_{\Delta^*} \lambda = \dot{\gamma} & \text{(horizontal part)} \\ \nabla_{\dot{\gamma}}^{[1],*} \lambda = 0 & \text{(vertical part)} \end{cases} \Leftrightarrow \\ &\begin{cases} \dot{\gamma}^i = u^i = \lambda^i \text{ for } i \in I \text{ and } u^j = 0 \text{ if } j \notin I & \text{(horizontal part)} \\ \dot{\lambda}_i = \{\mathfrak{h}, \lambda_i\} = - \sum_{k=1}^n \sum_{j \in I} c_{ij}^k \lambda_k u^j & \text{(vertical part)} \end{cases} \end{aligned} \tag{66}$$

where P_{Δ^*} denotes the projection onto the dual Δ^* of Δ , as given in Theorem 1.

Proof. The results (64) and (66) follow from standard application of the Pontryagin maximum principle (PMP Agrachev and Sachkov 2004) to the Riemannian and sub-Riemannian geodesic problem, respectively. First of all, we note that regarding the Hamiltonian in the Riemannian case (63), we have that it is computed by applying the Fenchel transform on the integrand of the action functional (i.e. squared Lagrangian):

$$\mathfrak{h}(g, \lambda) = \sup_{\dot{\gamma} \in T_g(G)} \{ \langle \lambda, \dot{\gamma} \rangle - \mathcal{L}^2(g, \dot{\gamma}) \} \text{ with } \lambda = \sum_{i=1}^n \lambda_i \omega^i \Big|_g \in T_g^*(G);$$

hence, we get the Hamiltonian $\mathfrak{h} : T^*(G) \rightarrow \mathbb{R}^+$ given by

$$\mathfrak{h} = \max_{(v^1, \dots, v^n)} \left\{ \sum_{i=1}^n \lambda_i v^i - \frac{1}{2} \sum_{i,j=1}^n v^i v^j g_{ij} \right\} = \frac{1}{2} \sum_{i,j=1}^n \lambda^i g_{ij} \lambda^j = \frac{1}{2} \sum_{i=1}^n \lambda^i \lambda_i, \tag{67}$$

with $\lambda^i = \sum_{j=1}^n g^{ij} \lambda_j$. The Hamiltonian in the SR-case (65) comes with the constraint

$\dot{\gamma} \in \Delta$ (i.e. $\dot{\gamma}^i = 0$ if $i \notin I$), and then with a similar type of reasoning above (but then with $v^i = 0$ if $i \notin I$), we get $\mathfrak{h} = \frac{1}{2} \sum_{i \in I} \lambda^i \lambda_i$ with $\lambda^i = \sum_{j \in I} g^{ij} \lambda_j$, and we find

the ‘extremal controls’ (Agrachev and Sachkov 2004): $v_{\max}^i = u^i = \lambda^i$.

Note that (64) and (66) are of the form $a \Leftrightarrow b \Leftrightarrow c$. We first comment on $a \Leftrightarrow c$ and then show $b \Leftrightarrow c$.

$a \Leftrightarrow c$ follows by direct computation as we show next. By computing, we have the following relation in Poisson brackets:

$$[\mathcal{A}_i, \mathcal{A}_j] = \sum_{k=1}^n c_{ij}^k \mathcal{A}_k \Leftrightarrow \{ \lambda_i, \lambda_j \} = \mathcal{A}_i \lambda_j - \mathcal{A}_j \lambda_i = \sum_{k=1}^n c_{ij}^k \lambda_k,$$

as the ‘conjugate momentum mapping’ gives rise to a Lie algebra morphism; see Agrachev and Sachkov (2004, p.164). Therefore (via (62), (67)), we find (with Liouville’s theorem and $c_{ij}^k = -c_{ji}^k$):

$$\begin{aligned} \dot{\gamma}^i &= \dot{u}^i = \{ \mathfrak{h}, u^i \} = \dot{\lambda}^i \Rightarrow u^i = \lambda^i, \\ \dot{\lambda}_i &= \{ \mathfrak{h}, \lambda_i \} = \sum_{j \in J} \frac{2}{2} \{ \lambda_j, \lambda_i \} \lambda^j = - \sum_{k=1}^n \sum_{j \in J} c_{ij}^k \lambda_k u^j, \end{aligned} \tag{68}$$

which hold for $i = 1, \dots, n$ in the Riemannian case and for $i \in I$ in the sub-Riemannian case. In the above expression, one must set $J = \{1, \dots, n\}$ in the Riemannian case and $J = I$ in the sub-Riemannian case.

$b \Leftrightarrow c$ follows by (68), and the expression (25) for the Lie-Cartan connection (with $\nu = 1$) and expression (32) for the partial Lie-Cartan connection (again with $\nu = 1$), respectively, are expressed in left-invariant coordinates. \square

Appendix B: Left-Invariant Vector Fields on SE(3) via Two Charts

We need two charts to cover $SO(3)$. When using the following coordinates (ZYZ-Euler angles) for $SE(3) = \mathbb{R}^3 \rtimes SO(3)$ for the first chart:

$$g = (x, y, z, \mathbf{R}_{e_z, \gamma} \mathbf{R}_{e_y, \beta} \mathbf{R}_{e_z, \alpha}), \text{ with } \beta \in (0, \pi), \alpha, \gamma \in [0, 2\pi). \tag{69}$$

Then the left-invariant vector fields are given by

$$\begin{aligned}
 \mathcal{A}_1|_g &= (\cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma) \partial_x + (\sin \alpha \cos \gamma + \cos \alpha \cos \beta \sin \gamma) \partial_y - \cos \alpha \sin \beta \partial_z \\
 \mathcal{A}_2|_g &= (-\sin \alpha \cos \beta \cos \gamma - \cos \alpha \sin \gamma) \partial_x + (\cos \alpha \cos \gamma - \sin \alpha \cos \beta \sin \gamma) \partial_y + \sin \alpha \sin \beta \partial_z \\
 \mathcal{A}_3|_g &= \sin \beta \cos \gamma \partial_x + \sin \beta \sin \gamma \partial_y + \cos \beta \partial_z, \\
 \mathcal{A}_4|_g &= \cos \alpha \cot \beta \partial_\alpha + \sin \alpha \partial_\beta - \frac{\cos \alpha}{\sin \beta} \partial_\gamma, \mathcal{A}_5|_g = -\sin \alpha \cot \beta \partial_\alpha + \cos \alpha \partial_\beta + \frac{\sin \alpha}{\sin \beta} \partial_\gamma, \\
 \mathcal{A}_6|_g &= \partial_\alpha.
 \end{aligned} \tag{70}$$

The above formulas do not hold for $\beta = \pi$ or $\beta = 0$: We need a second chart (Duits and Franken 2011):

$$g = (x, y, z, \mathbf{R}_{\mathbf{e}_x, \tilde{\gamma}} \mathbf{R}_{\mathbf{e}_y, \tilde{\beta}} \mathbf{R}_{\mathbf{e}_z, \alpha}), \text{ with } \tilde{\beta} \in [-\pi, \pi), \alpha \in [0, 2\pi), \tilde{\gamma} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right). \tag{71}$$

Then the left-invariant vector field formulas are (for $|\tilde{\beta}| \neq \frac{\pi}{2}$) given by

$$\begin{aligned}
 \mathcal{A}_1|_g &= \cos \alpha \cos \tilde{\beta} \partial_x + (\cos \tilde{\gamma} \sin \alpha + \cos \alpha \sin \tilde{\beta} \sin \tilde{\gamma}) \partial_y + (\sin \alpha \sin \tilde{\gamma} - \cos \alpha \sin \tilde{\beta} \cos \tilde{\gamma}) \partial_z \\
 \mathcal{A}_2|_g &= -\sin \alpha \cos \tilde{\beta} \partial_x + (\cos \alpha \cos \tilde{\gamma} - \sin \alpha \sin \tilde{\beta} \sin \tilde{\gamma}) \partial_y + (\sin \alpha \sin \tilde{\beta} \cos \tilde{\gamma} + \cos \alpha \sin \tilde{\gamma}) \partial_z \\
 \mathcal{A}_3|_g &= \sin \tilde{\beta} \partial_x - \cos \tilde{\beta} \sin \tilde{\gamma} \partial_y + \cos \tilde{\beta} \cos \tilde{\gamma} \partial_z, \\
 \mathcal{A}_4|_g &= -\cos \alpha \tan \tilde{\beta} \partial_\alpha + \sin \alpha \partial_{\tilde{\beta}} + \frac{\cos \alpha}{\cos \tilde{\beta}} \partial_{\tilde{\gamma}}, \mathcal{A}_5|_g = \sin \alpha \tan \tilde{\beta} \partial_\alpha + \cos \alpha \partial_{\tilde{\beta}} - \frac{\sin \alpha}{\cos \tilde{\beta}} \partial_{\tilde{\gamma}}, \\
 \mathcal{A}_6|_g &= \partial_\alpha.
 \end{aligned} \tag{72}$$

Appendix C: Proofs of Results on Lie-Cartan Connections

Proof of Lemma 2

Let X and Y be vector fields on G and γ the integral curve of X with $\gamma(0) = g$. We write $X = \sum_{i=1}^n x^i \mathcal{A}_i$ and $Y = \sum_{j=1}^n y^j \mathcal{A}_j$. By the definition of $\nabla^{[0]}$, we have that

$$\begin{aligned}
 (\nabla_{\tilde{\gamma}}^{[0]} Y)(g) &= \sum_{i,k=1}^n x^i \mathcal{A}_i|_g(y^k) \mathcal{A}_k|_g = \sum_{k=1}^n X|_g(y^k) \mathcal{A}_k|_g \\
 &= \sum_{k=1}^n \left(\lim_{t \rightarrow 0} \frac{y^k(\gamma(t)) - y^k(g)}{t} \right) \mathcal{A}_k|_g \\
 &= \lim_{t \rightarrow 0} \frac{\sum_{k=1}^n y^k(\gamma(t)) \left(L_{g\gamma(t)^{-1}} \right)_* \mathcal{A}_k|_{\gamma(t)} - Y(g)}{t} \\
 &= \lim_{t \rightarrow 0} \frac{\left(L_{g\gamma(t)^{-1}} \right)_* Y(\gamma(t)) - Y(g)}{t}.
 \end{aligned}$$

This proves (14).

Now let X, Y be left-invariant. Note that $\nabla^{[0]}Y = 0$ because $(L_{g(\gamma(t))^{-1}*})Y(\gamma(t)) = Y(g)$ in (14) regardless of γ . Then the alternative formula (15) for general Lie-Cartan connection $\nabla^{[v]}$ follows, as the structure constants $c_{ij}^k \in \mathbb{R}$ satisfy $\sum_k c_{ij}^k \mathcal{A}_k = [\mathcal{A}_i, \mathcal{A}_j]$ and the Lie bracket is bilinear for left-invariant vector fields, and we find $\nabla_X^{[v]}Y = \nabla_X^{[0]}Y + v[X, Y] = v[X, Y]$.

For our reformulation in (15), we used (17): $(\widetilde{\text{Ad}})_*(X_g)(Y_g) = [X_g, Y_g]$ that we show next. By the derivation in Jost (2011, Lemma.5.4.2), one has $(\text{Ad})_*(X_e)(Y_e) = [X_e, Y_e]$. Now the Cartan-Maurer form is a Lie algebra isomorphism, and we get (17)

$$\begin{aligned} \widetilde{\text{Ad}}_*(X_g)(Y_g) &= \widetilde{\text{Ad}}_*((L_g)_*X_e)((L_g)_*Y_e) \stackrel{(16)}{=} (L_g)_*\text{Ad}_*(X_e, Y_e) \\ &= [(L_g)_*X_e, (L_g)_*Y_e] = [X_g, Y_g]. \end{aligned}$$

Proof of Lemma 3

Let X, Y, Z be left-invariant vector fields. For all computations, we use the characterization of Lie-Cartan connections (15) from Lemma 2.

Torsion of $\nabla^{[v]}$: We have

$$\begin{aligned} T_{\nabla^{[v]}}(X, Y) &= \nabla_X^{[v]}Y - \nabla_Y^{[v]}X - [X, Y] \\ &= v[X, Y] - v[Y, X] - [X, Y] = (2v - 1)[X, Y]. \end{aligned}$$

Curvature of $\nabla^{[v]}$: By the Jacobi identity for Lie brackets, we have

$$\begin{aligned} R_{\nabla^{[v]}}(X, Y)Z &= \nabla_X^{[v]}\nabla_Y^{[v]}Z - \nabla_Y^{[v]}\nabla_X^{[v]}Z - \nabla_{[X, Y]}^{[v]}Z \\ &= v^2 ([X, [Y, Z]] - [Y, [X, Z]]) - v [[X, Y], Z] \\ &= v^2 [[X, Y], Z] - v [[X, Y], Z] = v(v - 1)[[X, Y], Z] \end{aligned}$$

Metric compatibility: We have

$$\begin{aligned} \nabla^{[v]}\mathcal{G}(X, Y, Z) &= X(\mathcal{G}(Y, Z)) - \mathcal{G}(Y, \nabla_X^{[v]}Z) - \mathcal{G}(\nabla_X^{[v]}Y, Z) \\ &= X(\mathcal{G}(Y, Z)) - v \mathcal{G}(Y, [X, Z]) - v \mathcal{G}([X, Y], Z) \\ &= -v (\mathcal{G}(Y, [X, Z]) + \mathcal{G}([X, Y], Z)) , \end{aligned}$$

where we note that $X(\mathcal{G}(Y, Z)) = 0$ because \mathcal{G} is also left invariant.

References

- Agrachev, A.A., Sachkov, Y.L.: Control Theory from the Geometrical Viewpoint, Vol 87. Springer (2004)
- Agrachev, A., Barilari, D., Boscain, U.: A Comprehensive Introduction to Sub-Riemannian Geometry. CUP Cambridge Studies in Advanced Mathematics (2020)
- Ali, S., Antoine, J., Gazeau, J.: Coherent States, Wavelets and Their Generalizations. Springer, New York/Berlin/Heidelberg (1999)
- Barbieri, D., Citti, G., Cocci, G., Sarti, A.: A cortical-inspired geometry for contour perception and motion integration. *J. Math. Imaging Vision* **49**(3), 511–529 (2014)
- Baspinar, E.: Minimal surfaces in Sub-Riemannian structures and functional geometry of the visual cortex. Ph.D. thesis, University of Bologna (2018)
- Bekkers, E.: Retinal Image Analysis using Sub-Riemannian Geometry in $SE(2)$. Ph.D. thesis, Eindhoven University of Technology (2017) cum laude ($\leq 5\%$ best at TU/e). https://pure.tue.nl/ws/files/52750592/20170123_Bekkers.pdf
- Bekkers, E., Duits, R., Berendschot, T., Haar Romeny, B.: A multi-orientation analysis approach to retinal vessel tracking. *JMIV* **49**(3), 583–610 (2014)
- Bekkers, E., Zhang, J., Duits, R., ter Haar Romeny, B.: Curvature based biomarkers for diabetic retinopathy via exponential curve fits in $se(2)$. In: Chen, X.E.A. (ed.) Proceedings of the Ophthalmic Medical Image Analysis International Workshop, Oct 113–120 (2015)
- Bekkers, E., R. Duits, Mashatkov, A., Sanguinetti, G.: A PDE approach to data-driven sub-Riemannian geodesics in $SE(2)$. *SIAM J. Imag. Sci.* **8**(4), 2740–2770 (2015)
- Bekkers, E., Duits, R., Mashtakov, A., Sachkov, Y.: Vessel tracking via sub-Riemannian geodesics on $\mathbb{R}^2 \times P^1$. *LNCS Proc. Geom. Sci. Inf. GSI 2017* **10589**, 1611–3349 (2017)
- Bekkers, E.J., Chen, D., Portegies, J.M.: Nilpotent approximations of sub-Riemannian distances for fast perceptual grouping of blood vessels in 2D and 3D. arXiv:1707.02811 [math], July (2017) arXiv: 1707.02811
- Bekkers, E., Lafarge, M., Veta, M., Eppenhof, K., Pluim, J., Duits, R.: Roto-translation covariant convolutional networks for medical image analysis. In: Frangi, F., et al. (ed.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, pp. 440–448. Springer International Publishing, Cham (2018)
- Bekkers, E., Loog, M., ter Haar Romeny, B., Duits, R.: Template matching via densities on the roto-translation group. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 452–466 (2018)
- Bertalmio, M., Calatroni, L., Franceschi, V., Franceschiello, B., Prandi, D.: A cortical-inspired model for orientation-dependent contrast perception: A link with wilson-cowan equations. In: Lellmann, J., Burger, M., Modersitzki, J., (eds.) Scale Space and Variational Methods in Computer Vision, pp. 472–484. Springer International Publishing, Cham (2019)
- Boscain, U., Chertovskih, R., Gauthier, J.-P., Prandi, D., Remizov, A.: Cortical-inspired image reconstruction via sub-Riemannian geometry and hypoelliptic diffusion. arXiv:1801.03800 (2018)
- Bosking, W.H., Zhang, Y., Schofield, B., Fitzpatrick, D.: Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *J. Neurosci.* **17**, 2112–2127 (1997)
- Bryant, R., Griffiths, P.: Reduction for constrained variational problems and $(1/2) \int \kappa^2 ds$. *Am. J. Math.* **108**(3), 525–570 (1986)
- Bryant, R., Griffiths, P., Grossman, D.: Exterior Differential Systems and Euler-Lagrange Partial Differential Equations. Chicago Lectures in Mathematics, Chicago and London (2003)
- Cartan, É.: Sur une classe remarquable d’espaces de riemann. *Bulletin de la Société Mathématique de France* **54**, 214–264 (1926)
- Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vis.* **22**(1), 61–79 (1997)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* **40**(1), 120–145 (2011)

- Chambolle, A., Pock, T.: Total roto-translation variation. Arxiv, 1–47, July (2018)
- Chen, D.: New minimal paths models for tubular structure extraction and image segmentation. Ph.D. thesis, Université Paris Dauphine, PSL Research University (2016)
- Chen, D., Cohen, L.: Fast asymmetric fronts propagation for image segmentation. *J. Math. Imaging Vision* **60**, 766–783 (2018)
- Chirikjian, G.S., Kyatkin, A.B.: *Engineering Applications of Noncommutative Harmonic Analysis: With Emphasis on Rotation and Motion Groups*. CRC Press, Boca Raton (2001)
- Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. *J. Math. Imaging Vision* **24**(3), 307–326 (2006)
- Citti, G., Sarti, A.: Models of the Visual Cortex in Lie Groups, pp. 1–55. Springer, Basel (2015)
- Citti, G., Franceschiello, B., Sanguinetti, G., Sarti, A.: Sub-Riemannian mean curvature flow for image processing. *SIIMS* **9**(1), 212–237 (2016)
- Cogliati, A., Mastroli, P.: Cartan, schouten and the search for connection. *Hist. Math.* **45**(1), 39–74 (2018)
- Cohen, T., Welling, M.: Group equivariant convolutional networks. In: *Proceedings of the 33rd International Conference on Machine Learning*, Vol. 48, pp. 1–12 (2016)
- Crandall, M., Lions, P.-L.: Viscosity solutions of hamilton-jacobi equations. *Trans. A.M.S.* **277**(1), 1–42 (1983)
- Creusen, E., Duits, R., Dela Haije, T.: Numerical schemes for linear and non-linear enhancement of dw-mri. In: *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 14–25. Springer, Berlin/Heidelberg (2011)
- Creusen, E., Duits, R., Vilanova, A., Florack, L.: Numerical schemes for linear and non-linear enhancement of DW-MRI. *Numer. Math. Theory Meth. Appl.* **6**(1), 138–168 (2013)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Processing* **16**(8), 2080–2095 (2007)
- Descoteaux, M., Deriche, R., Knosche, T.R., Anwander, A.: Deterministic and probabilistic tractography based on complex fibre orientation distributions. *IEEE Trans. Med. Imaging* **28**(2), 269–286 (2009)
- Dieudonné, J.: *Treatise on Analysis*, V. AP, New York (1977)
- Duits, R.: Perceptual organization in image analysis. Ph.D. thesis, Eindhoven University of Technology, Department of Biomedical Engineering (2005)
- Duits, R., Bekkers, E.: Lecture notes of the course Differential Geometry for Image Processing. Part II: Invertible Orientation Scores. tech. rep., TU/e Dep. of Mathematics and Computer Science (2020). www.win.tue.nl/~rduits/partIIversionI.pdf
- Duits, R., Franken, E.M.: Left invariant parabolic evolution equations on $SE(2)$ and contour enhancement via invertible orientation scores, part I: Linear left-invariant diffusion equations on $SE(2)$. *Q. Appl. Math.* **68**, 255–292 (2010a)
- Duits, R., Franken, E.M.: Left invariant parabolic evolution equations on $SE(2)$ and contour enhancement via invertible orientation scores, part II: Nonlinear left-invariant diffusion equations on invertible orientation scores. *Q. Appl. Math.* **68**, 293–331 (2010b)
- Duits, R., Franken, E.M.: Left-invariant diffusions on the space of positions and orientations and their application to crossing-preserving smoothing of HARDI images. *Int. J. Comput. Vis.* **92**, 231–264 (2011)
- Duits, R., Felsberg, M., Granlund, G., ter Haar Romeny, B.M.: Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the Euclidean motion group. *Int. J. Comput. Vis.* **79**(1), 79–102 (2007)
- Duits, R., Fuehr, H., Janssen, B., Florack, L., van Assen, H.: Evolution equations on gabor transforms and their applications. *ACHA* **35**(3), 483–526 (2013)
- Duits, R., Creusen, E., Ghosh, A., Dela Haije, T.: Morphological and linear scale spaces for fiber enhancement in DW-MRI. *J. Math. Imaging Vision* **46**, 326–368 (2013)
- Duits, R., Janssen, M.H., Hannink, J., Sanguinetti, G.R.: Locally adaptive frames in the roto-translation group and their applications in medical imaging. *J. Math. Imaging Vis.* **56**(3), 367–402 (2016)

- Duits, R., Ghosh, A., Dela Haije, T., Mashtakov, A.: On sub-Riemannian geodesics in $SE(3)$ whose spatial projections do not have cusps. *J. Dyn. Control. Syst.* **22**(4), 771–805 (2016)
- Duits, R., Meesters, S.P.L., Mirebeau, J.-M., Portegies, J.M.: Optimal paths for variants of the 2D and 3D Reeds-Shepp car with applications in image analysis. *JMIV* **60**, 816–848 (2018)
- Duits, R., St-Onge, E., Portegies, J., Smets, B.: Total variation and mean curvature PDEs on the space of positions and orientations. In: Lellmann, J., Modersitzki, J., Burger, M. (eds.) *Scale Space and Variational Methods in Computer Vision – 7th International Conference, SSVM 2019, Proceedings, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 211–223. Springer, 6 (2019)
- Duits, R., Bekkers, E.J., Mashtakov, A.: Fourier transform on the homogeneous space of 3d positions and orientations for exact solutions to linear PDEs. *Entropy: Special Issue: Joseph Fourier 250th Birthday: Modern Fourier Analysis and Fourier Heat Equation in Information Sciences for the XXIst century*, Vol. 21, no. 1, pp. 1–38 (2019)
- Evans, L.C.: *Partial Differential Equations*. American Mathematical Society, Providence (2010)
- Fabbrini, L., Greco, M., Messina, M., Pinelli, G.: Improved edge enhancing diffusion filter for speckle-corrupted images. *IEEE Geosci. Remote Sens. Lett.* **11**(1), 99–103 (2013)
- Felsberg, M.: *Adaptive Filtering Using Channel Representations*, pp. 31–48. Springer, London (2012)
- Felsberg, M., Forssen, P.-E., Scharf, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 209–222 (2006)
- Forssen, P.-E.: *Low and Medium Level Vision using Channel Representations*. Ph.D. thesis, Linköping University, Sweden (2004) Dissertation No. 858, ISBN 91-7373-876-X
- Franceschiello, B., Mashtakov, A., Citti, G., Sarti, A.: Geometrical optical illusion via sub-riemannian geodesics in the roto-translation group. *Differ. Geom. Appl.* **65**, 55–77 (2019)
- Frangi, A., et al.: Multiscale vessel enhancement filtering. In: *Proceedings of Medical Image Computing and Computer-Assisted Intervention: Lecture Notes in Computer Science*, Vol. 1496, pp. 130–137 (1998)
- Franken, E.M.: *Enhancement of crossing elongated structures in images*. Ph.D. thesis, Department of Biomedical Engineering, Eindhoven University of Technology, Eindhoven, October (2008) cum laude and selected for promotion prize ($\leq 2\%$ best at TU/e)
- Franken, E.M., Duits, R.: Crossing preserving coherence-enhancing diffusion on invertible orientation scores. *Int. J. Comput. Vis.* **85**(3), 253–278 (2009)
- Franken, E.M., Duits, R., ter Haar Romeny, B.M.: Curvature estimation for enhancement of crossing curves. In: Niessen, W., Westin, C.F., Nielsen, M. (eds.) *Digital Proceedings of the 8th IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, held in conjunction with the IEEE International Conference on Computer Vision (Rio de Janeiro, Brazil), pp. 1–8, Omnipress, Oct (2007) Awarded the MMBIA 2007 best paper award
- Fuehr, H.: *Abstract Harmonic Analysis of Continuous Wavelet Transforms*. Springer, Heidelberg/New York (2005)
- Grossmann, A., Morlet, J., Paul, T.: Integral transforms associated to square integrable representations. *J. Math. Phys.* **26**, 2473–2479 (1985)
- Haar Romenij ter, B.: *Front-end vision and multi-scale image analysis : multi-scale computer vision theory and applications*, written in Mathematica. Computational imaging and vision. Kluwer Academic Publishers, CIVI (2003)
- Hannink, J., Duits, R., Bekkers, E.: Crossing-preserving multi-scale vesselness. In: G. et al. (eds.) *MICCAI vol. 8674*, pp. 603–610 (2014)
- Hormander, L.: Hypoelliptic second order differential equations. *Acta Math.* **119**, 147–171 (1968)
- Janssen, M., Duits, R., Breeuwer, M.: Invertible orientation scores of 3D images. *SSVM-LNCS* **9087**, 563–575 (2014)
- Janssen, M., Dela Haije, T., Martin, F., Bekkers, E., Duits, R.: The hessian of axially symmetric functions on $se(3)$ and application in 3D image analysis. *LNCS* (2017) Submitted to SSVM (2017)

- Janssen, M.H.J., Janssen, A.J.E.M., Bekkers, E.J., Bescós, J.O., Duits, R.: Design and processing of invertible orientation scores of 3D images. *J. Math. Imaging Vision* **60**(9), 1427–1458 (2018)
- Jost, J.: *Riemannian Geometry and Geometric Analysis*. Springer (2011)
- Kobayashi, S., Nomizu, K.: *Foundations of Differential Geometry*, vol. 1. New York (1963)
- Kolar, I., Slovák, J., Michor, P.: *Natural operations in differential geometry*. Springer (1999) corrected version of original version in (1993)
- Lebrun, M.: An analysis and implementation of the bm3d image denoising method. *IEEE Trans. Image Process* **2**, 175–213 (2012)
- Lee, J.M., Chow, B., Chu, S.-C., Glickenstein, D., Guenther, C., Isenberg, J., Ivey, T., Knopf, D., Lu, P., Luo, F., et al.: Manifolds and differential geometry. *Topology* **643**, 658 (2009)
- Mantegazza, C., Mennucci, A.: Hamilton-jacobi equations and distance functions on Riemannian manifolds. *App. Math. Optim.* **47**(1), 1–25 (2002)
- Martin, F., Duits, R.: Lie analysis homepage. <http://www.lieanalysis.nl/> (2017)
- Mashtakov, A., Duits, R., Sachkov, Y., Bekkers, E., Beschastnyi, I.: Tracking of lines in spherical images via sub-Riemannian geodesics in $SO(3)$. *JMIV* **58**(2), 239–364 (2017)
- Meesters, S., Ossenblok, P., Wagner, L., Schijns, O., Boon, P., Florack, L., Vilanova, Duits, R.: Stability metrics for optic radiation tractography: Towards damage prediction after resective surgery. *J. Neurosci. Methods* (2017). <https://doi.org/10.1016/j.jneumeth.2017.05.029>
- Mirebeau, J.: Anisotropic fast-marching on cartesian grids using lattice basis reduction. *SIAM J. Numer. Anal.* **52**(4), 1573–1599 (2014)
- Mirebeau, J.-M.: Fast marching methods for curvature penalized shortest paths. *J. Math. Imaging Vis.* Special Issue: Orientation Analysis and Differential Geometry in Image Processing **60**(6), 784–815 (2018)
- Mirebeau, J., Portegies, J.: Hamiltonian fast marching: A numerical solver for anisotropic and non-holonomic eikonal PDEs. *IPOL* **9**, 47–93 (2019)
- Momayyez-Siahkal, P., Siddiqi, K.: 3D stochastic completion fields for fiber tractography. In: *Proceedings of IEEE Computer Society Conference on Computer Vision Pattern Recognition*, pp. 178–185, June (2009)
- Monti, R., Cassano, F.: Surface measures in Carnot-carathéody spaces. *Calc. Var.* **13**, 339–376 (2001)
- Mumford, D.: *Elastica and computer vision. Algebraic Geometry and Its Applications*. Springer, pp. 491–506 (1994)
- Pechaud, M., Descoteaux, M., Keriven, R.: *Brain Connectivity Using Geodesics in HARDI*, pp. 482–489. Springer, Berlin/Heidelberg (2009)
- Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990)
- Petitot, J.: The neurogeometry of pinwheels as a sub-Riemannian contact structure. *J. Physiol. Paris* **97**, 265–309 (2003)
- Petitot, J.: *Elements of Neurogeometry. Lecture Notes in Morphogenesis*. Springer (2017)
- Piuze, E., Sparring, J., Siddiqi, K.: Maurer-cartan forms for fields on surfaces: Application to heart fiber geometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(12), 2492–2504 (2015)
- Portegies, J.: *PDEs on the Lie Group $SE(3)$ and their Applications in Diffusion-Weighted MRI*. Ph.D. thesis, Department of Mathematics and Computer Science, TU/e, February (2018)
- Portegies, J.M., Fick, R.H.J., Sanguinetti, G.R., Meesters, S.P.L., Girard, G., Duits, R.: Improving fiber alignment in HARDI by combining contextual PDE flow with constrained spherical deconvolution. *PLoS ONE* **10**(10) (2015). <https://doi.org/10.1371/journal.pone.0138122>
- Portegies, J., Sanguinetti, G., Meesters, S., Duits, R.: New approximation of a scale space kernel on $SE(3)$ and applications in neuroimaging. In: *SSVM 2015, LNCS 9087*, pp. 40–52 (2015)
- Portegies, J., Meesters, S., Ossenblo, P., Fuster, A., Florack, L., Duits, R.: *Brain connectivity measures via direct sub-finslerian front propagation on the 5D sphere bundle of positions and directions*, ch. 24, p. 14. Springer (2019)
- Prčková, V., Andorrà, M., Villoslada, P., Martínez-Heras, E., Duits, R., Fortin, D., Rodrigues, P., Descoteaux, M.: Contextual diffusion image post-processing aids clinical applications. In: *Hotz,*

- I., Schultz, T. (eds.) Visualization and Processing of Higher Order Descriptors for Multi-Valued Data, Cham, pp. 353–377. Springer International Publishing (2015)
- Reisert, M., Kiselev, V.G.: Fiber continuity: An anisotropic prior for ODF estimation. *IEEE Trans. Med. Imaging* **30**(6), 1274–1283 (2011)
- Saccon, A., Aguiar, A.P., Hausler, A.J., Hauser, J., Pascoal, A.M.: Constrained motion planning for multiple vehicles on $se(3)$. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp. 5637–5642, Dec (2012)
- Sachkov, Y.: Maxwell strata in the Euler elastic problem. *J. Dyn. Control. Syst.* **14**(2), 169–234 (2008)
- Sachkov, Y.: Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane. *ESAIM: Control Optim. Calc. Var.* **17**, 293–321 (2011)
- Sanguinetti, G., Bekkers, E., Duits, R., Janssen, M.H.J., Mashtakov, A., Mirebeau, J.-M.: *Sub-Riemannian Fast Marching in SE(2)*. Springer (2015)
- Sharma, U., Duits, R.: Left-invariant evolutions of wavelet transforms on the similitude group. *ACHA* **39**, 110–137 (2015)
- Siffre, L.: Rigid-Motion Scattering for Image Classification. Ph.D. thesis, Ecole Polytechnique, Paris (2014)
- Skibbe, H., Reisert, M.: Spherical tensor algebra: A toolkit for 3D image processing. *JMIV* **58**, 349–381 (2017)
- Smets, B.: Geometric image denoising and machine learning (cum laude). Master’s thesis, Industrial and Applied Mathematics, CASA-TU/e, June (2019) Supervisor R.Duits. www.win.tue.nl/~rduits/reportBartSmets.pdf
- Smets, B., Duits, R., St-Onge, E., Portegies, J.: Total variation and mean curvature PDEs on the homogeneous space of positions and orientations. Submitted to JMIV special issue (2019)
- Smets, B., Portegies, J., Bekkers, E., Duits, R.: Pde-based group equivariant convolutional neural networks. Technical report, Department of Mathematics and Computer Science TU/e, Jan (2020)
- St Onge, E., Meesters, S., Bekkers, E., Descoteaux, M., Duits, R.: Hardi denoising with mean-curvature enhancement pde on $SE(3)$. In: J. et al. (eds.) ISMRM Proceedings, Montreal, pp. 1–3 (2019). <http://archive.ismrm.org/2019/3409.html>
- ter Elst, A.F.M., Robinson, D.W.: Weighted subcoercive operators on Lie groups. *J. Funct. Anal.* **157**, 88–163 (1998)
- Tournier, J.D., Calamante, F., Connelly, A.: Robust determination of the fibre orientation distribution in diffusion mri: non-negativity constrained super-resolved spherical deconvolution. *Neuroimage* **35**(4), 1459–1472 (2007)
- Weickert, J.: Coherence-enhancing diffusion filtering. *Int. J. Comput. Vis.* **31**(2/3), 111–127 (1999)
- Zhang, J., Duits, R., ter Haar Romeny, B., Sanguinetti, G.: Numerical approaches for linear left-invariant diffusions on $SE(2)$, their comparisons to exact solutions, and their applications in retinal imaging. *Numer. Math. Theory Methods Appl.* **9**, 1–50 (2016)
- Zhang, J., Dashtbozorg, B., Bekkers, E., Pluim, J., Duits, R., ter Haar Romeny, B.: Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores. *IEEE-TMI* **35**(12), 2631–2644 (2016)
- Zhang, J., Dashtbozorg, B., Huang, F., Berendschot, T.T., ter Haar Romeny, B.M.: Analysis of retinal vascular biomarkers for early detection of diabetes. In: European Congress on Computational Methods in Applied Sciences and Engineering, pp. 811–817. Springer (2017)
- Zhu, S., et al.: Retinal vascular tortuosity in hospitalized patients with type 2 diabetes and diabetic retinopathy in China. *J. Biomed. Sci. Eng.* **9**(10), 143 (2016)
- Zhu, S., Liu, H., Du, R., Annick, D.S., Chen, S., Qian, W.: Tortuosity of retinal main and branching arterioles, venules in patients with type 2 diabetes and diabetic retinopathy in china. *IEEE Access* **8**, 6201–6208 (2020)



PDE-Constrained Shape Optimization: Toward Product Shape Spaces and Stochastic Models

45

Caroline Geiersbach, Estefania Loayza-Romero, and Kathrin Welker

Contents

Introduction	1586
Optimization Over Product Shape Manifolds	1588
Optimization on Shape Spaces with Steklov–Poincaré Metric	1590
Optimization of Multiple Shapes	1598
Stochastic Multi-shape Optimization and the Stochastic Gradient Method	1605
Numerical Investigations	1611
Deterministic Model Problem	1612
Stochastic Model Problem	1614
Numerical Experiments	1617
Conclusion	1624
References	1626

This work has been partly supported by the state of Hamburg within the Landesforschungsförderung under project “Simulation-Based Design Optimization of Dynamic Systems Under Uncertainties” (SENSUS) with project number LFF-GK11, and by the German Academic Exchange Service (DAAD) within the program “Research Grants-Doctoral Programmes in Germany, 2017/18.”

C. Geiersbach (✉)
Weierstrass Institute, Berlin, Germany
e-mail: caroline.geiersbach@wias-berlin.de

E. Loayza-Romero
Institute for Analysis and Numerics, University of Münster, Münster, Germany
e-mail: estefania.loayza-romero@uni-muenster.de

K. Welker
Faculty of Mechanical Engineering and Civil Engineering,
Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, Hamburg,
Germany
e-mail: welker@hsu-hh.de

Abstract

Shape optimization models with one or more shapes are considered in this chapter. Of particular interest for applications are problems in which a so-called shape functional is constrained by a partial differential equation (PDE) describing the underlying physics. A connection can be made between a classical view of shape optimization and the differential geometric structure of shape spaces. To handle problems where a shape functional depends on multiple shapes, a theoretical framework is presented, whereby the optimization variable can be represented as a vector of shapes belonging to a product shape space. The multi-shape gradient and multi-shape derivative are defined, which allows for a rigorous justification of a steepest descent method with Armijo backtracking. As long as the shapes as subsets of a hold-all domain do not intersect, solving a single deformation equation is enough to provide descent directions with respect to each shape. Additionally, a framework for handling uncertainties arising from inputs or parameters in the PDE is presented. To handle potentially high-dimensional stochastic spaces, a stochastic gradient method is proposed. A model problem is constructed, demonstrating how uncertainty can be introduced into the problem and the objective can be transformed by use of the expectation. Finally, numerical experiments in the deterministic and stochastic case are devised, which demonstrate the effectiveness of the presented algorithms.

Keywords

Shape optimization · Stochastic approximation · PDE-constrained optimization under uncertainty · Product manifolds · Optimization on manifolds

Introduction

Shape optimization is concerned with problems in which an objective function is supposed to be minimized with respect to a shape, or a subset of \mathbb{R}^d . One challenge in shape optimization is finding the correct model to describe the set of shapes; another is finding a way to handle the lack of vector structure of the shape space. In principle, a finite dimensional optimization problem can be obtained, for example, by representing shapes as splines. However, this representation limits the admissible set of shapes, and the connection of shape calculus with infinite dimensional spaces (Delfour and Zolésio 2001; Sokolowski and Zolésio 1992) leads to a more flexible approach. It was suggested to embed shape optimization problems in the framework of optimization on shape spaces (Schulz 2014; Welker 2016). One possible approach is to cast the sets of shapes in a Riemannian viewpoint, where each shape is a point on an abstract manifold equipped with a notion of distances between shapes (see, e.g., Michor and Mumford 2005, 2006). From a theoretical and computational point of view, it is attractive to optimize in Riemannian shape manifolds because algorithmic ideas from Absil et al. (2008) can be combined with approaches from differential geometry. Here, the Riemannian shape gradient can be

used to solve such shape optimization problems using the gradient descent method. In the past, major effort in shape calculus has been devoted toward expressions for shape derivatives in the so-called Hadamard form, which are integrals over the surface (cf. Delfour and Zolésio 2001; Sokolowski and Zolésio 1992). During the calculation of these expressions, volume shape derivative terms arise as an intermediate result. In general, additional regularity assumptions are necessary in order to transform the volume forms into surface forms. Besides saving analytical effort, this makes volume expressions preferable to Hadamard forms. In this chapter, the Steklov–Poincaré metric is considered, which allows to use the volume formulations (cf. Schulz et al. 2016). The reader is referred to Hardesty et al. (2020) and Hiptmair et al. (2015) for a comparison on the volume and boundary formulations with respect to their order of convergence in a finite element setting.

In applications, often more than one shape needs to be considered, e.g., in electrical impedance tomography, where the material distribution of electrical properties such as electric conductivity and permittivity inside the body is examined (Cheney et al. 1999; Kwon et al. 2002; Laurain and Sturm 2016) and the optimization of biological cell composites in the human skin (Siebenborn and Vogel 2021; Siebenborn and Welker 2017). If a shape is seen as a point on an abstract manifold, it is natural to view a collection of shapes as a vector of points. Using this perspective, a shape optimization problem can be formulated over multiple shapes. This novel, multi-shape optimization problem is developed in this chapter.

A second area of focus in this chapter is in the development of stochastic models for multi-shape optimization problems. There is an increasing effort to incorporate uncertainty into shape optimization models (see, for instance Dambrine et al. 2015, 2019, Hiptmair et al. 2018, Liu et al. 2017, and Martínez-Frutos et al. 2016). Many relevant problems contain additional constraints in the form of a PDE, which describe the physical laws that the shape should obey. Often, material coefficients and external inputs might not be known exactly but rather be randomly distributed according to a probability distribution obtained empirically. In this case, one might still wish to optimize over a set of these possibilities to obtain a more robust shape. When the number of possible scenarios in the probability space is small, then the optimization problem can be solved over the entire set of scenarios. This approach is not relevant for most applications, as it becomes intractable if the random field has more than a few scenarios. For problems with PDEs containing uncertain inputs or parameters, either the stochastic space is discretized or sampling methods are used. If the stochastic space is discretized, one typically relies on a finite-dimension assumption, where a truncated expansion is used as an approximation of the infinite-dimensional random field. Numerical methods include stochastic Galerkin method (Babuska et al. 2004) and sparse-tensor discretization (Schwab and Gittelson 2011). Sample-based approaches involve taking random or carefully chosen realizations of the input parameters; this includes Monte Carlo or quasi-Monte Carlo methods and stochastic collocation (Babuška et al. 2007). In the stochastic approximation approach, dating back to a chapter by Robbins and Monro (1951), one uses a stochastic gradient in place of a gradient to iteratively minimize the expected value over a random function. Recently, stochastic approximation was proposed to solve problems formulated over a shape space that contains uncertainties (Geiersbach

et al. 2021). A novel stochastic gradient method was formulated over infinite-dimensional shape spaces and convergence of the method was proven. The work was informed by its demonstrated success in the context of PDE-constrained optimization under uncertainty (Geiersbach and Pflug 2019; Haber et al. 2012; Martin et al. 2018; Geiersbach and Wollner 2020; Geiersbach and Scarinci 2021).

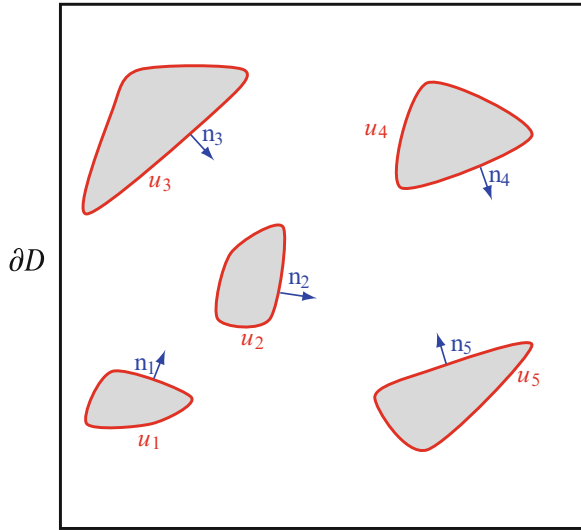
The chapter is structured as follows. Section “[Optimization Over Product Shape Manifolds](#)” is concerned with deterministic shape optimization. First, in section “[Optimization on Shape Spaces with Steklov–Poincaré Metric](#)”, it is summarized how the theory of deterministic PDE-constrained shape optimization problems can be connected with the differential geometric structure of the space of smooth shapes. The novel contribution of this chapter is in section “[Optimization of Multiple Shapes](#)”, which concentrates on more than one shape to be optimized in the optimization model. A framework is introduced to justify a mesh deformation method using a Steklov–Poincaré metric defined on a product manifold. This novel framework is further developed in section “[Stochastic Multi-shape Optimization and the Stochastic Gradient Method](#)” in the context of shape optimization under uncertainty. The stochastic gradient method is revisited in the context of problems depending on multiple shapes. Numerical experiments demonstrating the effectiveness of the deterministic and stochastic methods are shown in section “[Numerical Investigations](#)”. Finally, closing remarks are shared in section “[Conclusion](#)”.

Optimization Over Product Shape Manifolds

This chapter is concerned with a class of optimization problems, where the optimization variable is a vector $u = (u_1, \dots, u_N)$ of non-intersecting shapes contained in a bounded domain $D \subset \mathbb{R}^d$ as shown in Fig. 1 for $d = 2$ and $N = 5$. This domain will sometimes be called the hold-all domain, and its boundary is denoted by ∂D . The outer normal vector field \mathbf{n} on a shape $u \in \mathcal{U}^N$ is defined by $\mathbf{n} = (n_1, \dots, n_N)$, where n_i denotes the unit outward normal vector field to u_i for $i = 1, \dots, N$.

Next, the shape space concept considered in this chapter needs to be clarified. Shapes space definitions have been extensively studied in recent decades. Already in 1984, Kendall introduced the notion of a shape space. Here, a shape space is modeled as a quotient space of not totally degenerate vectors of landmark positions. However, there is a large number of different shape concepts, e.g., plane curves (Michor and Mumford 2007), surfaces in higher dimensions (Bauer et al. 2011; Michor and Mumford 2005), boundary contours of objects (Fuchs et al. 2009; Ling and Jacobs 2007; Wirth and Rumpf 2009), multiphase objects (Wirth et al. 2011), characteristic functions of measurable sets (Zolésio 2007), morphologies of images (Droske and Rumpf 2007), and planar triangular meshes (Herzog and Loayza-Romero 2020). In a lot of processes in engineering, medical imaging, and science, there is a great interest to equip the space of all shapes with a significant metric to distinguish between different shape geometries. In the simplest shape space case (landmark vectors), the distances between shapes can be measured by the Euclidean distance, but in general, the study of shapes and their

Fig. 1 Illustration of the domain D in \mathbb{R}^2 for $N = 5$



similarities is a central problem. In contrast to a parametric optimization problem, which can be obtained, e.g., by representing shapes as splines, the connection of shape calculus with infinite dimensional spaces (Delfour and Zolésio 2001; Ito and Kunisch 2008; Sokolowski and Zolésio 1992) leads to a more flexible approach. As already mentioned, solving PDE-constrained shape optimization problems under a differential geometric paradigm has various advantages (Schulz et al. 2015), one of them being the opportunity to obtain a natural measure of similarity of shapes through the Riemannian metric. Moreover, depending on the metric defined over a manifold, different goals can be achieved. This chapter focuses on the Steklov–Poincaré metric (Schulz et al. 2016) because of its direct relation to the finite element method.

In view of using the Steklov–Poincaré metric, this chapter concentrates on shape spaces as Riemannian manifolds. Thus, it is assumed $u_i \in \mathcal{U}_i$ for all $i = 1, \dots, N$ for Riemannian manifolds (\mathcal{U}_i, G^i) , i.e., u is an element of the *product shape space* $\mathcal{U}^N := \mathcal{U}_1 \times \dots \times \mathcal{U}_N = \prod_{i=1}^N \mathcal{U}_i$. If there is only one shape, the notation \mathcal{U} instead of \mathcal{U}^1 is used. Since a Riemannian metric G^i varies with the point of evaluation, it will be denoted $G_p^i(\cdot, \cdot): T_p\mathcal{U}_i \times T_p\mathcal{U}_i \rightarrow \mathbb{R}$, to highlight its dependence on the point p . Hereby, the *tangent space* at a point $p \in \mathcal{U}_i$ is defined in its geometric version as

$$T_p\mathcal{U}_i = \{c: \mathbb{R} \rightarrow \mathcal{U}_i : c \text{ differentiable, } c(0) = p\} / \sim,$$

where the equivalence relation for two differentiable curves $c, \tilde{c}: \mathbb{R} \rightarrow \mathcal{U}_i$ with $c(0) = \tilde{c}(0) = p$ is defined as follows:

$$c \sim \tilde{c} \Leftrightarrow \frac{d}{dt}\phi_\alpha(c(t))|_{t=0} = \frac{d}{dt}\phi_\alpha(\tilde{c}(t))|_{t=0} \forall \alpha \text{ with } u \in U_\alpha,$$

where $\{(U_\alpha, \phi_\alpha)\}_\alpha$ atlas of \mathcal{U}_i .

A main focus in shape optimization is in the investigation of shape functionals. A shape functional on \mathcal{U}^N is given by a function

$$j: \mathcal{U}^N \rightarrow \mathbb{R}, u \mapsto j(u).$$

An *unconstrained shape optimization problem* is given by

$$\min_{u \in \mathcal{U}^N} j(u). \quad (1)$$

Often, shape optimization problems are constrained by equations, e.g., equations involving an unknown function of two or more variables and at least one partial derivative of this function. The objective may depend on not only the shapes u but also the *state variable* y , where the state variable is the solution of the underlying constraint. In other words, one has a shape functional of the form $\hat{j}: \mathcal{U}^N \times \mathcal{Y} \rightarrow \mathbb{R}$ and an operator $e: \mathcal{U}^N \times \mathcal{Y} \rightarrow \mathcal{W}$, where \mathcal{Y} and \mathcal{W} are Banach spaces. One therefore has a *constrained shape optimization problem* of the form

$$\begin{aligned} \min_{(u,y) \in \mathcal{U}^N \times \mathcal{Y}} \hat{j}(u, y) \\ \text{s.t. } e(u, y) = 0. \end{aligned} \quad (2)$$

When e in (2) represents a PDE, the shape optimization problem is called *PDE-constrained*. Formally, if the PDE has a (unique) solution given any choice of u , then the *control-to-state operator* $S: \mathcal{U}^N \rightarrow \mathcal{Y}, u \mapsto y$ is well-defined. With $j(u) := \hat{j}(u, Su)$ one obtains an unconstrained optimization problem of the form (1). This observation justifies the following work with (1), although later in the application section, a problem of the form (2) is presented.

Section “[Optimization on Shape Spaces with Steklov–Poincaré Metric](#)” concentrates on $N = 1$ and summarizes how the theory of deterministic PDE-constrained shape optimization problems can be connected to the differential geometric structure of shape spaces. Here, in view of obtaining efficient gradient-based algorithms, one focuses on the Steklov–Poincaré metric considered in Schulz et al. (2016). Afterward, section “[Optimization of Multiple Shapes](#)” concentrates on $N > 1$, which leads to product shape manifolds. It will be shown that it is possible to define a product metric and use this to justify the main result of this chapter, Theorem 1. It is rigorously argued that vector fields induced by the shape derivative give descent directions with respect to each individual element of the shape space as well as the corresponding element of the product shape space.

Optimization on Shape Spaces with Steklov–Poincaré Metric

In this subsection, optimization with respect to one shape $u \in \mathcal{U}$ is discussed, i.e., $N = 1$ is chosen. Additionally, the connection between Riemannian geometry on

the space of smooth shapes and shape optimization is analyzed. Please note the following: one shape is both an element of a manifold and a subset of \mathbb{R}^d . In classical shape calculus, a shape is considered to be a subset of \mathbb{R}^d , only. However, this subsection explains that equipping a shape with additional structure provides theoretical advantages, enabling the use of concepts from differential geometry like the pushforward, exponential maps, etc.

Shape calculus. First, notation and terminology of basic shape optimization concepts will be set up. For a detailed introduction into shape calculus, the reader is referred to the monographs (Delfour and Zolésio 2001; Sokolowski and Zolésio 1992). The concept of shape derivatives is needed. In order to define these derivatives, one concentrates on the shape u as subset of $D \subset \mathbb{R}^d$ and considers a family $\{F_t\}_{t \in [0, T]}$ of mappings $F_t: \overline{D} \rightarrow \mathbb{R}^d$ such that $F_0 = \text{id}$, where \overline{D} denotes the closure of D and $T > 0$. This family transforms shapes u into new *perturbed shapes*

$$F_t(u) = \{F_t(x) : x \in u\}.$$

Such a transformation can be described by the *velocity method* or by the *perturbation of identity* (cf. Sokolowski and Zolésio 1992, pages 45 and 49). In the following, the perturbation of identity is considered. It is defined by $F_t^W(x) := x + tW(x)$, where $W: \overline{D} \rightarrow \mathbb{R}^d$ denotes a sufficiently smooth vector field.

Definition 1 (Shape derivative). Let $D \subset \mathbb{R}^d$ be open, $u \subset D$ and $k \in \mathbb{N} \cup \{\infty\}$. The Eulerian derivative of a shape functional j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$ is defined by

$$dj(u)[W] := \lim_{t \rightarrow 0^+} \frac{j(F_t^W(u)) - j(u)}{t}. \quad (3)$$

If for all directions $W \in C_0^k(D, \mathbb{R}^d)$ the Eulerian derivative (3) exists and the mapping

$$C_0^k(D, \mathbb{R}^d) \rightarrow \mathbb{R}, \quad W \mapsto dj(u)[W]$$

is linear and continuous, the expression $dj(u)[W]$ is called the shape derivative of j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$. In this case, j is called shape differentiable of class C^k at u .

The proof of existence of shape derivatives can be done via different approaches like the Lagrangian (Sturm 2013), min-max (Delfour and Zolésio 2001), chain rule (Sokolowski and Zolésio 1992), and rearrangement (Ito et al. 2008) methods, among others. If the objective functional is given by a volume integral, under the assumptions of the Hadamard structure theorem (cf. Sokolowski and Zolésio 1992,

Theorem 2.27), the shape derivative can be expressed as an integral over the domain, the so-called *volume* or *weak* formulation, and also as an integral over the boundary, the so-called *surface* or *strong* formulation. Recent advances in PDE-constrained optimization on shape manifolds are based on the surface formulation, also called *Hadamard form*, as well as intrinsic shape metrics. Major effort in shape calculus has been devoted toward such surface expressions (cf. Delfour and Zolésio 2001; Sokolowski and Zolésio 1992), which are often very tedious to derive. When one derives a shape derivative of an objective functional, which is given by an integral over the domain, one first gets the volume formulation. This volume form can be converted into its surface form by applying the integration by parts formula. In order to apply this formula, one needs a higher regularity of the state and adjoint of the underlying PDE. Recently, it has been shown that the weak formulation has numerical advantages (see, for instance, Berggren 2010, Gangl et al. 2015, Hiptmair and Paganini 2015, and Paganini 2015). In Hardesty et al. (2020) and Laurain and Sturm (2013), practical advantages of volume shape formulations have also been demonstrated.

Shape calculus combined with differential geometric structure of shape manifolds. Solving shape optimization problems is made more difficult by the fact that the set of permissible shapes generally does not allow a vector space structure, which is one of the main difficulties for the formulation of efficient optimization methods. In particular, without a vector space structure, there is no obvious distance measure, which is needed to establish convergence properties. In many practical applications, this difficulty is circumvented by characterizing the shapes of interest by finitely many parameters such that the parameters are elements of a vector space. Often, a priori parametrizations of the shapes of interest are used because of the resulting vector space framework matching standard optimization software. However, this limits the insight into the optimal shapes severely, because only shapes corresponding to the a priori parametrization can be reached. One possibility to avoid this limitation would be to focus on shape optimization in the setting of shape spaces. If one cannot work in vector spaces, shape spaces which allow a Riemannian structure like Riemannian manifolds are the next best option.

Now, a shape $u \subset D$ is viewed also as an element of a Riemannian shape manifold (\mathcal{U}, G) . This means that the shape functional J is defined on the manifold. Next, the derivative of a scalar field $j: \mathcal{U} \rightarrow \mathbb{R}$ needs to be defined.

Definition 2 (Pushforward). For each point $u \in \mathcal{U}$, the pushforward associated with $j: \mathcal{U} \rightarrow \mathbb{R}$ is given by the map

$$(j_*)_u: T_u\mathcal{U} \rightarrow \mathbb{R}, c \mapsto \frac{d}{dt}j(c(t))|_{t=0} = (j \circ c)'(0).$$

Remark 1. In general, the pushforward is defined for a map f between two differential manifolds M and N . The definition depends on the used tangent space.

In this setting, where tangent spaces are defined as equivalence classes of curves, the pushforward of $f: M \rightarrow N$ at a point $p \in M$ is generally given by a map between the tangent spaces, i.e., $(f_*)_p: T_p M \rightarrow T_{f(p)} N$ with $(f_*)_p(c) := \frac{d}{dt} f(c(t))|_{t=0} = (f \circ c)'(0)$.

With the help of the pushforward, it is possible to define the Riemannian shape gradient.

Definition 3 (Riemannian shape gradient). Let (\mathcal{U}, G) be a Riemannian manifold and $j: \mathcal{U} \rightarrow \mathbb{R}$. A Riemannian shape gradient $\nabla j(u) \in T_u \mathcal{U}$ is defined by the relation

$$(j_*)_u w = G_u(\nabla j(u), w) \quad \forall w \in T_u \mathcal{U}.$$

Thanks to the definition of the Riemannian shape gradient, it is possible to formulate the gradient method on the Riemannian manifold (\mathcal{U}, G) (cf. Algorithm 1). The Riemannian shape gradient with respect to G is computed from (4). The negative solution $-v^k$ is then used as descent direction for the objective functional j in each iteration k . In order to update the shape iterates, the exponential map in Algorithm 1 is used; because the calculations of optimization methods on manifolds have to be performed in tangent spaces, points from a tangent space have to be mapped to the manifold in order to define the next iterate. Figure 2

Algorithm 1 Steepest descent method on (\mathcal{U}, G) with Armijo backtracking line search

Require: Objective function j on (\mathcal{U}, G)

Input: Initial shape $u^0 \in \mathcal{U}$

constants $\hat{\alpha} > 0$ and $\sigma, \rho \in (0, 1)$ for Armijo backtracking strategy

for $k = 0, 1, \dots$ **do**

[1] Compute the Riemannian shape gradient $v^k \in T_{u^k} \mathcal{U}$ with respect to G by solving

$$(j_*)_{u^k} w = G_{u^k}(v^k, w) \quad \forall w \in T_{u^k} \mathcal{U}. \tag{4}$$

[2] Compute Armijo backtracking step-size:

Set $\alpha := \hat{\alpha}$.

while $j(\exp_{u^k}(-\alpha v^k)) > j(u^k) - \sigma \alpha \|v^k\|_G^2$

Set $\alpha := \rho \alpha$.

end while

Set $t^k := \alpha$.

[3] Set

$$u^{k+1} := \exp_{u^k}(-t^k v^k). \tag{5}$$

end for

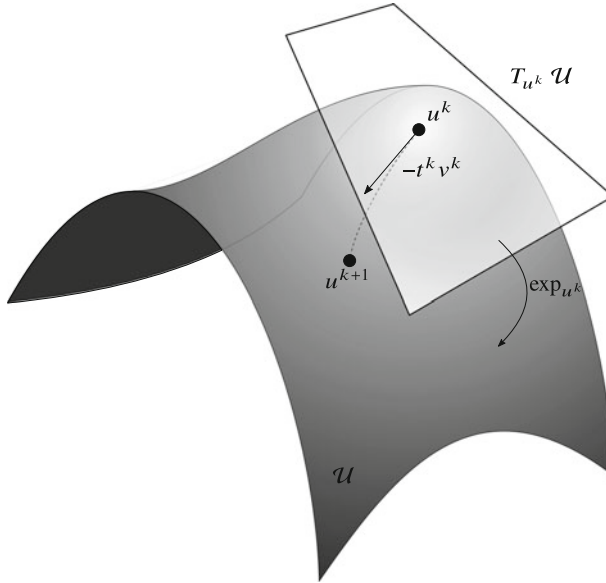


Fig. 2 Iterate $u^{k+1} = \exp_{u^k}(-t^k v^k)$, where $\exp_{u^k} : T_{u^k} \mathcal{U} \rightarrow \mathcal{U}$

illustrates this situation. With (5) the $(k + 1)$ -th shape iterate u^{k+1} is calculated, where $\exp_{u^k} : T_{u^k} \mathcal{U} \rightarrow \mathcal{U}$, $z \mapsto \exp_{u^k}(z)$ denotes the exponential map; this defines a local diffeomorphism between the tangent space $T_{u^k} \mathcal{U}$ and the manifold \mathcal{U} by following the locally uniquely defined geodesic starting in the k -th shape iterate $u^k \in \mathcal{U}$ in the direction $-v^k \in T_{u^k} \mathcal{U}$. In Algorithm 1, an Armijo backtracking line search technique is used to calculate the step-size t^k in each iteration. Here, the norm introduced by the metric under consideration is needed, $\| \cdot \|_G := \sqrt{G(\cdot, \cdot)}$.

Optimization on the space of smooth shapes. This chapter focuses on the manifold of d -dimensional smooth shapes. The set of all $(d - 1)$ -dimensional smooth shapes is considered in Michor and Mumford (2005) and can be characterized by

$$B_e = B_e(S^{d-1}, \mathbb{R}^d) := \text{Emb}(S^{d-1}, \mathbb{R}^d) / \text{Diff}(S^{d-1}).$$

Here, $\text{Emb}(S^{d-1}, \mathbb{R}^d)$ denotes the set of all embeddings from the unit circle S^{d-1} into \mathbb{R}^d , and $\text{Diff}(S^{d-1})$ is the set of all diffeomorphisms from S^{d-1} into itself. In Kriegel and Michor (1997), it is verified that the shape space B_e is a smooth manifold. The tangent space is isomorphic to the set of all smooth normal vector fields along c , i.e.,

$$T_u B_e(S^{d-1}, \mathbb{R}^d) \cong \left\{ h : h = \alpha n, \alpha \in C^\infty(S^{d-1}) \right\},$$

where n denotes the outer unit normal field to the shape u . Next, the connection of shape derivatives with the geometric structure of B_e is addressed. This combination results in efficient optimization techniques on B_e .

In view of obtaining gradient-based optimization approaches, the gradient needs to be specified. The gradient will be characterized by the chosen Riemannian metric on B_e . Several Riemannian metrics on this shape space are examined, e.g., Bauer et al. (2011) and Michor and Mumford (2005, 2007). All these metrics arise from the L^2 -metric by putting weights, derivatives, or both in it. In this manner, one gets three groups of metrics: the *almost local metrics* which arise by putting weights in the L^2 -metric (cf. Bauer et al. 2012 and Michor and Mumford 2007), the *Sobolev metrics* which arise by putting derivatives in the L^2 -metric (cf. Bauer et al. 2011 and Michor and Mumford 2007), and the *weighted Sobolev metrics* which arise by putting both weights and derivatives in the L^2 -metric (cf. Bauer et al. 2012). In Schulz (2014), the curvature weighted metric, which is an almost local metric, was considered in shape optimization to formulate approaches for unconstrained shape optimization problems. The first Sobolev metric was used in Schulz et al. (2015) to formulate gradient-based methods to solve PDE-constrained shape optimization problems. In Welker (2021), the gradient-based results from Schulz et al. (2015) are extended by formulating the covariant derivative with respect to the first Sobolev metric. Thanks to that derivative, a Riemannian shape Hessian with respect to the first Sobolev metric could be specified, which opens the door to formulating higher-order methods in space of smooth shapes. If Sobolev or almost local metrics are considered, one has to deal with strong formulations of shape derivatives. An intermediate and equivalent result in the process of deriving these expressions is the weak expression as already mentioned above. These weak expressions are preferable over strong forms. Not only does one save analytical effort, but one needs lower regularity for the weak expressions. Moreover, the weak expressions are typically easier to implement numerically. However, in the case of the more attractive weak formulation, the shape manifold B_e and the corresponding Sobolev or almost local metrics are not appropriate. One possible approach to use weak forms is addressed in Schulz et al. (2016), which considers Steklov–Poincaré metrics. In the following, some of the main results related to this metric from Schulz et al. (2016) are summarized in view of obtaining efficient optimization methods, also for shape optimization problems under uncertainty. For a comparison of the approach resulting from considering the first Sobolev and the approach based on the Steklov–Poincaré metric, the reader is referred to Schulz and Siebenborn (2016), Welker (2016), and Welker (2021).

The *Steklov–Poincaré metric* is given by

$$g^S : H^{1/2}(u) \times H^{1/2}(u) \rightarrow \mathbb{R},$$

$$(v, w) \mapsto \int_u v \cdot (S^{pr})^{-1} w \, ds. \quad (6)$$

Here S^{pr} denotes the projected Poincaré–Steklov operator, which is given by

$$S^{pr} : H^{-1/2}(u) \rightarrow H^{1/2}(u), \quad v \mapsto \operatorname{tr}(V) \cdot \mathbf{n}$$

with $\operatorname{tr} : H_0^1(D, \mathbb{R}^d) \rightarrow H^{1/2}(u, \mathbb{R}^d)$ denoting the trace operator on Sobolev spaces for vector-valued functions and $V \in H_0^1(D, \mathbb{R}^d)$ solving the Neumann problem

$$a(V, W) = \int_u v (\operatorname{tr}(W) \cdot \mathbf{n}) \, ds \quad \forall W \in H_0^1(D, \mathbb{R}^d),$$

where $a : H_0^1(D, \mathbb{R}^d) \times H_0^1(D, \mathbb{R}^d) \rightarrow \mathbb{R}$ is a symmetric and coercive bilinear form. Note that a Steklov–Poincaré metric depends on the choice of the bilinear form. Thus, different bilinear forms lead to various Steklov–Poincaré metrics. To define a metric on B_e , the Steklov–Poincaré metric is restricted to the mapping $g^S : T_u B_e \times T_u B_e \rightarrow \mathbb{R}$.

Next, the connection between B_e equipped with the Steklov–Poincaré metric g^S and shape calculus is stated. As already mentioned, the shape derivative can be expressed in a weak and strong form under the assumptions of the Hadamard structure theorem. The Hadamard structure theorem actually states the existence of a scalar distribution r on the boundary of a domain. However, in the following, it is always assumed that r is an integrable function. In general, if $r \in L^1(u)$, then r is obtained in the form of the trace on u of an element of $W^{1,1}(D)$. This means that it follows from Hadamard structure theorem that the shape derivative can be expressed more conveniently as

$$d^{\text{surf}} j(u)[W] := \int_u r(s) (W(s) \cdot \mathbf{n}(s)) \, ds. \tag{7}$$

In view of the connection between the shape space B_e with respect to the Steklov–Poincaré metric g^S and shape calculus, $r \in C^\infty(u)$ is assumed. In contrast, if the shape functional is a pure volume integral, the weak form is given by

$$d^{\text{vol}} j(u)[W] := \int_D RW(x) \, dx, \tag{8}$$

where R is a differential operator acting linearly on the vector field W .

Definition 4 (Shape gradient with respect to Steklov–Poincaré metric). Let $r \in C^\infty(u)$ denote the function in the shape derivative expression (7). Moreover, let S^{pr} be the projected Poincaré–Steklov operator. A representation $v \in T_u B_e \cong C^\infty(u)$ of the shape gradient in terms of g^S is determined by

$$g^S(v, w) = (r, w)_{L^2(u)} \quad \forall w \in C^\infty(u),$$

which is equivalent to

$$\int_u w(s) \cdot [(S^{pr})^{-1}v](s)ds = \int_u r(s)w(s)ds \quad \forall w \in C^\infty(u). \tag{9}$$

From (9), one gets that a vector $V \in H_0^1(D, \mathbb{R}^d) \cap C^\infty(D, \mathbb{R}^d)$ can be viewed as an extension of a Riemannian shape gradient to the hold-all domain D because of the identities

$$g^S(v, w) = d^{\text{surf}}j(u)[W] = a(V, W) \quad \forall W \in H_0^1(D, \mathbb{R}^d) \cap C^\infty(D, \mathbb{R}^d), \tag{10}$$

where $v = \text{tr}(V) \cdot n, w = \text{tr}(W) \cdot n \in T_u B_e$. Since the strong formulation of the shape derivative arises from the weak formulation under the assumptions of the Hadamard structure theorem, one could also choose $d^{\text{vol}}j(u)[W]$ in (10). This fact together with identity (10) allows one to consider weak expressions of shape derivatives to compute the shape gradient with respect to g^S . Since both expressions of the shape derivative can be used, only $dj(u)[W]$ is written in the following. In order to compute the shape gradient, one has to solve the so-called deformation equation

$$a(V, W) = dj(u)[W] \quad \forall W \in H_0^1(D, \mathbb{R}^d) \cap C^\infty(D, \mathbb{R}^d). \tag{11}$$

One option for $a(\cdot, \cdot)$ is the bilinear form associated with linear elasticity, i.e.,

$$a^{\text{elas}}(V, W) := \int_D (\lambda \text{tr}(\epsilon(V))\text{id} + 2\mu \epsilon(V)) : \epsilon(W) dx,$$

where $\epsilon(W) := \frac{1}{2}(\nabla W + \nabla W^T)$, $A : B$ denotes the Frobenius inner product for two matrices A, B and $\lambda, \mu \in \mathbb{R}$ denote the so-called Lamé parameters.

Remark 2. Note that it is not ensured that $V \in H_0^1(D, \mathbb{R}^d)$ solving the PDE (in weak form)

$$a(V, W) = dj(u)[W] \quad \forall W \in H_0^1(D, \mathbb{R}^d)$$

is $C^\infty(D, \mathbb{R}^d)$. Thus, $v = S^{pr}r = (\text{tr} V) \cdot n$ is not necessarily an element of $T_u B_e$. However, under special assumptions depending on the coefficients of a second-order partial differential operator and the right-hand side of the PDE, a weak solution V that is at least H_0^1 -regular is C^∞ (cf. Evans 1998, Section 6.3, Theorem 6).

Thanks to the definition of the gradient with respect to g^S , Algorithm 1 can be applied on (B_e, g^S) . In order to be in line with the above theory, it is assumed in Algorithm 1 that in each iteration k , the shape u^k is a subset of the hold-all

domain D . The Riemannian shape gradient is computed with respect to g^S from (11). The negative solution $-v = -\text{tr } V \cdot n$ is then used as descent direction for the objective functional j . The exponential map is used to update the shape iterates in Algorithm 1. Instead of the exponential map, it is also possible to use the concept of a retraction; this is a smooth mapping $\mathcal{R}: T\mathcal{U} \rightarrow \mathcal{U}$ satisfying $\mathcal{R}^{u^k}(0_{u^k}) = u^k$ and the so-called local rigidity condition $\mathcal{R}_*^{u^k}(0_{u^k}) = \text{id}_{T_{u^k}\mathcal{U}}$, where \mathcal{R}^{u^k} denotes the restriction of \mathcal{R} to $T_{u^k}\mathcal{U}$, 0_{u^k} is the zero element of $T_{u^k}\mathcal{U}$, and $\mathcal{R}_*^{u^k}(0_k)$ denotes the pushforward of $0_{u^k} \in T_{u^k}\mathcal{U}$ by \mathcal{R} . An example of a retraction is

$$\mathcal{R}^{u^k}: T_{u^k}\mathcal{U} \rightarrow \mathcal{U}, v \mapsto \mathcal{R}^{u^k}(v) := u^k + v \tag{12}$$

(cf. Schulz and Welker 2018). The retraction is only a local approximation; for large vector fields, the image of this function may no longer belong to B_ϵ . This retraction is closely related to the perturbation of the identity, which is defined for vector fields on the domain D . Given a starting shape u^{k+1} in the k -th iteration of Algorithm 1, the perturbation of the identity acting on the domain D in the direction V^k , where V^k solves (11) for $u = u^k$, gives

$$D(u^{k+1}) = \{x \in D \mid x = x^k - t^k V^k\}. \tag{13}$$

As vector fields induced from solving (11) have less regularity than is required on the manifold, it is worth mentioning that the shape u^{k+1} resulting from this update could leave the manifold B_ϵ . To summarize, either large or less smooth vector fields can contribute to the iterate u^{k+1} leaving the manifold. One indication that the iterate has left the manifold would be that the curve u^{k+1} develops corners. Another possibility is that the curve u^{k+1} self-intersects. One way to avoid this behavior is by preventing the underlying mesh to break (meaning elements from the finite element discretization overlap). One can avoid broken meshes as long as the step-size is not chosen to be too large.

Remark 3. In practice, the hold-all domain is discretized by a mesh, for instance, by finite elements (FE). Then in each iteration k , one computes the vector field V^k defined on the hold-all domain by solving (11) for $u = u^k$. The vector field then informs how to move the computational mesh. For instance, with a FE discretization, V^k acts on each node of the FE mesh, which moves not only the shape but also all other nodes of the mesh. An example of this is later shown in the application in Fig. 7.

Optimization of Multiple Shapes

This subsection extends Algorithm 1 to multiple shapes $u = (u_1, \dots, u_N) \in \mathcal{U}^N$ with $N > 1$ and $\mathcal{U}^N = \prod_{i=1}^N \mathcal{U}_i$ for Riemannian manifolds (\mathcal{U}_i, G^i) . For this,

the concepts of the pushforward, Riemannian shape gradient, and shape derivative need to be generalized. In view of applications in shape optimization, the metric \mathcal{G}^N on the product manifold is related later to the Steklov–Poincaré metric. As a main contribution, the computation of vector fields extended to the hold-all domain is discussed.

Analogously to Abraham et al. (2012, 3.3.12 Proposition), one can identify the tangent bundle $T\mathcal{U}^N$ with the product space $T\mathcal{U}_1 \times \dots \times T\mathcal{U}_N$. In particular, there is an identification of the tangent space of the product manifold \mathcal{U}^N in the point u ; more precisely,

$$T_u\mathcal{U}^N \cong T_{u_1}\mathcal{U}_1 \times \dots \times T_{u_N}\mathcal{U}_N.$$

Let $\pi_i: \mathcal{U}^N \rightarrow \mathcal{U}_i, i = 1, \dots, N$, be the N canonical projections. With these identifications, one can then define the product metric \mathcal{G}^N to the product shape space \mathcal{U}^N . For this, one needs the concept of the pushforward and the pullback by π_i . For each point $u \in \mathcal{U}^N$, the *pushforward associated with canonical projections* $\pi_i, i = 1, \dots, N$, is given by the map

$$(\pi_{i*})_u: T_u\mathcal{U}^N \rightarrow T_{\pi_i(u)}\mathcal{U}_i, \mathbf{c} \mapsto \frac{d}{dt}\pi_i(\mathbf{c}(t))|_{t=0} = (\pi_i \circ \mathbf{c})'(0).$$

The *pullback by the canonical projections* $\pi_i, i = 1, \dots, N$, is the linear map from the space of 1-forms on \mathcal{U}_i to the space of 1-forms on \mathcal{U}^N and denoted by

$$\pi_i^*: T_{\pi_i(u)}^*\mathcal{U}_i \rightarrow T_u^*\mathcal{U}^N,$$

where $T_{\pi_i(u)}^*\mathcal{U}_i$ and $T_u^*\mathcal{U}^N$ are the dual spaces of $T_{\pi_i(u)}\mathcal{U}_i$ and $T_u\mathcal{U}^N$, respectively. Thanks to these definitions, the product metric \mathcal{G}^N to the product shape space \mathcal{U}^N can be defined:

$$\mathcal{G}^N = \sum_{i=1}^N \pi_i^* G^i.$$

In particular, one has

$$\mathcal{G}_u^N(v, w) = \sum_{i=1}^N G_{\pi_i(u)}^i(\pi_{i*}v, \pi_{i*}w) \quad \forall v, w \in T_u\mathcal{U}^N. \tag{14}$$

Arguments identical to the ones in the proof of O’neill (1983, chapter 3, lemma 5) make $(\mathcal{U}^N, \mathcal{G}^N)$ a Riemannian product manifold.

In order to define a shape gradient of a functional $j: \mathcal{U}^N \rightarrow \mathbb{R}$ using the definition of the product metric in (14), Definition 2 needs to be first generalized to the product shape space.

Definition 5 (Multi-pushforward). For each point $u \in \mathcal{U}^N$, the multi-pushforward associated with $j: \mathcal{U}^N \rightarrow \mathbb{R}$ is given by the map

$$(j_*)_u: T_u \mathcal{U}^N \rightarrow \mathbb{R}, \mathbf{c} \mapsto \frac{d}{dt} j(\mathbf{c}(t))|_{t=0} = (j \circ \mathbf{c})'(0).$$

Definition 6 (Riemannian multi-shape gradient). The Riemannian multi-shape gradient for a shape functional $j: \mathcal{U}^N \rightarrow \mathbb{R}$ at the point $u = (u_1, \dots, u_N) \in \mathcal{U}^N$ is given by $v \in T_u \mathcal{U}^N$ satisfying

$$\mathcal{G}_u^N(v, w) = (j_*)_u w \quad \forall w \in T_u \mathcal{U}^N.$$

Notice that because of the identification of $T_u \mathcal{U}^N$ with $T_{u_1} \mathcal{U}_1 \times \dots \times T_{u_N} \mathcal{U}_N$, the elements \mathbf{c} and w from Definitions 5 and 6, respectively, should be understood as vectors of the form $\mathbf{c}(t) = (c_1(t), \dots, c_N(t))$ and $w = (w_1, \dots, w_N)$.

Thanks to the definition of the Riemannian multi-shape gradient, the steepest descent method on $(\mathcal{U}^N, \mathcal{G}^N)$ can be formulated (see Algorithm 2). This method essentially follows the same steps as Algorithm 1. In Algorithm 2, a *multi-exponential map*

$$\exp_{u^k}^N: T_{u^k} \mathcal{U}^N \rightarrow \mathcal{U}^N, z = (z_1, \dots, z_N) \mapsto (\exp_{u_1^k} z_1, \dots, \exp_{u_N^k} z_N) \quad (15)$$

is needed to update the shape vector $u^k = (u_1^k, \dots, u_N^k)$ in each iteration k , where $\exp_{u_i^k}: T_{u_i^k} \mathcal{U}_i \rightarrow \mathcal{U}_i, z \mapsto \exp_{u_i^k}(z)$ for all $i = 1, \dots, N$. An Armijo backtracking line search strategy is used to calculate the step-size t^k in each iteration. Here, the norm introduced on \mathcal{G}^N is given by $\|\cdot\|_{\mathcal{G}^N} := \sqrt{\mathcal{G}^N(\cdot, \cdot)}$.

So far in this subsection, each shape u_i has been considered as an element of the Riemannian shape manifold (\mathcal{U}_i, G^i) , for all $i = 1, \dots, N$, in order to define the multi-shape gradient with respect to the Riemannian metric \mathcal{G}^N . In classical shape calculus, each shape u_i is only a subset of \mathbb{R}^d . If one focuses on this perspective, then it is possible to generalize the classical shape derivative to a partial shape derivative and, thus, to a multi-shape derivative. With these generalized objects, a connection between shape calculus and the differential geometric structure of the product shape manifold \mathcal{U}^N can be made.

Let D be partitioned in N non-overlapping Lipschitz domains $\Delta_1, \dots, \Delta_N$ such that $u_k \subset \Delta_k$. This construction will be referred as an *admissible partition* (see Fig. 3 for an example in \mathbb{R}^2). The indicator function $\mathbb{1}_{\Delta_i}: D \rightarrow \{0, 1\}$ is defined by $\mathbb{1}_{\Delta_i}(x) = 1$, if $x \in \Delta_i$, and $\mathbb{1}_{\Delta_i}(x) = 0$, otherwise.

Definition 7 (Multi-shape derivative). Let $D \subset \mathbb{R}^d$ be open, $u = (u_1, \dots, u_N)$, and observe an arbitrary admissible partition with $u_i \subset \Delta_i$ for all $i = 1, \dots, N$. Further, let $k \in \mathbb{N} \cup \{\infty\}$. For $i = 1, \dots, N$, the i -th partial Eulerian derivative of a shape functional j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$ is defined by

Algorithm 2 Steepest descent method on $(\mathcal{U}^N, \mathcal{G}^N)$ with Armijo backtracking line search

Require: Objective function j on $(\mathcal{U}^N, \mathcal{G}^N)$

Input: Initial shape $u^0 = (u_1^0, \dots, u_N^0) \in \mathcal{U}^N$

constants $\hat{\alpha} > 0$ and $\sigma, \rho \in (0, 1)$ for Armijo backtracking strategy

for $k = 0, 1, \dots$ **do**

[1] Compute the Riemannian multi-shape gradient v^k with respect to \mathcal{G}^N by solving

$$(j_*)_{u^k} w = G_{u^k}(v^k, w) \quad \forall w \in T_{u^k} \mathcal{U}^N. \tag{16}$$

[2] Compute Armijo backtracking step-size:

Set $\alpha := \hat{\alpha}$.

while $j(\exp_{u^k}(-\alpha v^k)) > j(u^k) - \sigma \alpha \|v^k\|_{\mathcal{G}^N}^2$

Set $\alpha := \rho \alpha$.

end while

Set $t^k := \alpha$.

[3] Set

$$u^{k+1} := \exp_{u^k}^N(-t^k v^k). \tag{17}$$

end for

$$d_{u_i} j(u)[W|_{\Delta_i}] := \lim_{t \rightarrow 0^+} \frac{j(u_1, \dots, u_{i-1}, F_t^{W|_{\Delta_i}}(u_i), u_{i+1}, \dots, u_N) - j(u)}{t}. \tag{18}$$

If for all directions $W \in C_0^k(D, \mathbb{R}^d)$ the i -th partial Eulerian derivative (18) exists and the mapping

$$C_0^k(D, \mathbb{R}^d) \rightarrow \mathbb{R}, \quad W \mapsto d_{u_i} j(u)[W|_{\Delta_i}]$$

is linear and continuous, the expression $d_{u_i} j(u)[W|_{\Delta_i}]$ is called the i -th partial shape derivative of j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$. If the i -th partial shape derivatives of j at u in the direction $W \in C_0^k(D, \mathbb{R}^d)$ exist for all $i = 1, \dots, N$, then

$$dj(u)[W] := \sum_{i=1}^N d_{u_i} j(u)[W|_{\Delta_i}] \tag{19}$$

defines the multi-shape derivative of j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$.

Remark 4. For a single shape, by the Hadamard Structure Theorem, the shape derivative takes either the forms (7) or (8). Using the definition above, the

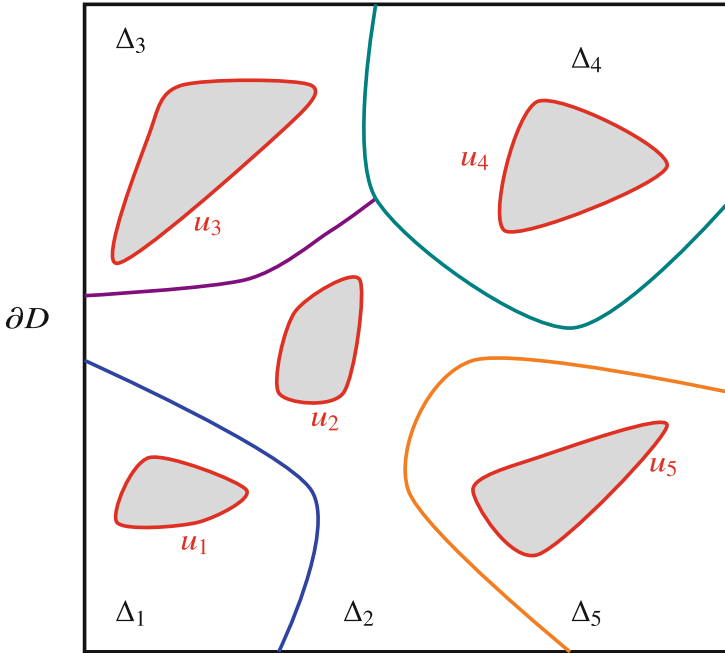


Fig. 3 Illustration of a possible partition of $D \subset \mathbb{R}^2$

Hadamard Structure Theorem for multiple shapes can also be applied. The surface representation for $r_i \in L^1(u_i)$ is

$$d^{\text{surf}} j(u)[W] := \sum_{i=1}^N d_{u_i}^{\text{surf}} j(u)[W|_{\Delta_i}] = \sum_{i=1}^N \int_{u_i} r_i(s) (W|_{\Delta_i}(s) \cdot n(s)) \, ds. \quad (20)$$

The volume form is

$$d^{\text{vol}} j(u)[W] := \sum_{i=1}^N d_{u_i}^{\text{vol}} j(u)[W|_{\Delta_i}] = \sum_{i=1}^N \int_{\Delta_i} R_i W|_{\Delta_i}(x) \, dx, \quad (21)$$

where R_i is a differential operator acting linearly on the vector field $W|_{\Delta_i}$. In the volume form, it is clear that if $R_i = R|_{\Delta_i}$ for all i , the form (21) reduces to

$$d^{\text{vol}} j(u)[W] = \int_D RW(x) \, dx. \quad (22)$$

The expressions (20) and (22) suggest that the multi-shape derivative is in fact independent of the partition, provided it is an admissible one, i.e., with nonintersecting shapes and $u_i \subset \Delta_i$ for nonintersecting subdomains Δ_i . This can

be exploited computationally. It will be shown that to compute descent directions for the shape objective $j: \mathcal{U}^N \rightarrow \mathbb{R}$ according to (16), it is enough to solve the following variational problem:

$$\text{find } V \in H_0^1(D, \mathbb{R}^d) \text{ such that } a(V, W) = dj(u)[W] \quad \forall W \in H_0^1(D, \mathbb{R}^d). \tag{23}$$

By virtue of Remark 2, the solution of (23) is not necessarily $C^\infty(D, \mathbb{R}^d)$, and these elements should be considered only formally.

In preparation for Theorem 1, observe an admissible partition of D . The following Hilbert spaces are defined for all $i = 1 \dots, N$:

$$\begin{aligned} \mathbb{V}_i &:= \{V \in H^1(\Delta_i, \mathbb{R}^d) : V = 0 \text{ on } \partial D \cap \partial \Delta_i\}, \\ \mathbb{V}_i^0 &= H_0^1(\Delta_i, \mathbb{R}^d). \end{aligned}$$

The following trace space for $\Gamma_i := \partial \Delta_i \setminus \partial D$ is defined:

$$\Lambda_i := \left\{ \eta \in H^{1/2}(\Gamma_i, \mathbb{R}^d) : \eta = V|_{\Gamma_i}, \text{ for a suitable } V \text{ in } H_0^1(D, \mathbb{R}^d) \right\}.$$

One has (cf. Quarteroni and Valli 1999, Subchapter 1.2) $\Lambda_i = H^{1/2}(\Gamma_i, \mathbb{R}^d)$ if $\Gamma_i \cap \partial D = \emptyset$. In case $\Gamma_i \cap \partial D \neq \emptyset$, the space Λ_i is strictly included in $H^{1/2}(\Gamma_i, \mathbb{R}^d)$ and is endowed with a norm which is larger than the norm of $H^{1/2}(\Gamma_i, \mathbb{R}^d)$. The trace space over $\Gamma := \cup_{i=1}^N \Gamma_i$ is given by

$$\Lambda := \left\{ \eta \in H^{1/2}(\Gamma, \mathbb{R}^d) : \eta = V|_{\Gamma}, \text{ for a suitable } V \text{ in } H_0^1(D, \mathbb{R}^d) \right\}.$$

The following main theorem justifies solving (23) to obtain a vector field that gives descent directions with respect to each shape.

Theorem 1. *Observe an arbitrary admissible partition of D . Suppose symmetric and coercive $a_i: \mathbb{V}_i \times \mathbb{V}_i \rightarrow \mathbb{R}$ are defined for all $i = 1, \dots, N$ such that $a: H_0^1(D, \mathbb{R}^d) \times H_0^1(D, \mathbb{R}^d) \rightarrow \mathbb{R}$ satisfies $a(V, W) = \sum_{i=1}^N a_i(V|_{\Delta_i}, W|_{\Delta_i})$ for all $V, W \in H_0^1(D, \mathbb{R}^d)$. Then the variational problem: find $V \in H_0^1(D, \mathbb{R}^d)$ such that*

$$a(V, W) = dj(u)[W] \quad \forall W \in H_0^1(D, \mathbb{R}^d) \tag{24}$$

is equivalent to the system of variational problems: find $V_i \in \mathbb{V}_i, i = 1, \dots, N$ such that

$$a_i(V_i, W_i) = d_{u_i} j(u)[W_i] \quad \forall W_i \in \mathbb{V}_i^0, \tag{25a}$$

$$V_i = V_\ell \quad \text{on all nonempty } \partial \Delta_i \cap \partial \Delta_\ell, \tag{25b}$$

$$\sum_{i=1}^N a_i(V_i, E_i \eta_i) = \sum_{i=1}^N d_{u_i} j(u)[E_i \eta_i] \quad \forall \eta \in \Lambda, \tag{25c}$$

where $\eta_i = \eta|_{\Gamma_i}$ and $E_i: \Lambda_i \rightarrow \mathbb{V}_i$ denotes an arbitrary extension operator, i.e., a continuous operator from Λ_i to \mathbb{V}_i satisfying $(E_i \eta_i)|_{\Gamma_i} = \eta_i$.

Proof. This proof follows the arguments from Quarteroni and Valli (1999, Sec. 1.2), generalizing for the case $N > 2$. First, it is shown that (24) yields the system (25). Let V be a solution to (24). Then setting $V_i = V|_{\Delta_i}$ for $i = 1, \dots, N$, one trivially obtains (25b) in the sense of the corresponding traces. Moreover, using $W_i = W|_{\Delta_i}$ for an arbitrary $W \in H_0^1(D, \mathbb{R}^d)$, one has $a_i(V_i, W_i) = d_{u_i} j(u)[W_i]$ for all $W_i \in \mathbb{V}_i$ and in particular for all $W_i \in \mathbb{V}_i^0$, showing (25a). Moreover, the function

$$E\eta := \begin{cases} E_1 \eta_1 & \text{in } \Delta_1, \\ \vdots & \\ E_N \eta_N & \text{in } \Delta_N \end{cases} \tag{26}$$

belongs to $H_0^1(D, \mathbb{R}^d)$. In particular, one has

$$a(V, E\eta) = dj(u)[E\eta],$$

which is equivalent to (25c).

Suppose now that $V_i, i = 1, \dots, N$, are solutions to the system (25). Let

$$V := \begin{cases} V_1 & \text{in } \Delta_1, \\ \vdots & \\ V_N & \text{in } \Delta_N. \end{cases}$$

From the condition $V_i = V_\ell$ on $\partial\Delta_i \cap \partial\Delta_\ell$, one obtains $V \in H_0^1(D, \mathbb{R}^d)$. Now, taking $W \in H_0^1(D, \mathbb{R}^d)$ gives $\eta := W|_\Gamma \in \Lambda$. Defining E as in (26) with $\eta_i = \eta|_{\Gamma_i}$ yields $(W|_{\Delta_i} - E_i \eta_i) \in \mathbb{V}_i^0$ and hence (25a) and (25c) imply

$$\begin{aligned} a(V, W) &= \sum_{i=1}^N a_i(V_i, W|_{\Delta_i} - E_i \eta_i) + a_i(V_i, E_i \eta_i) \\ &= \sum_{i=1}^N d_{u_i} j(u)[W|_{\Delta_i} - E_i \eta_i] + d_{u_i} j(u)[E_i \eta_i] \\ &= dj(u)[W], \end{aligned}$$

meaning V solves (23). □

Remark 5. There are several consequences of Theorem 1. The first is computational: particularly for large-scale problems with many shapes, a decomposition approach can be used by solving (25) for an arbitrary admissible partition instead of the more expensive problem (24). Second, for smaller-scaled problems, the theorem justifies solving (24) “all-at-once” to obtain descent directions with respect to each shape. In particular, the solution V_i to (25a) gives a descent direction $-V_i$ for the shape u_i ; due to the coercivity of a_i , one has

$$d_{u_i} j(u)[-V_i] = a_i(V_i, -V_i) < 0.$$

Remark 6. The second and third conditions of (25) are continuity conditions along Γ for the solution V and the normal flux (normal stress) relating V_i for all $i = 1, \dots, N$. The extension operator E_i can be chosen arbitrarily; one example is the extension-by-zero operator (cf. Hiptmair et al. 2015).

Thanks to Theorem 1, the Riemannian multi-shape gradient with respect to $g^S := \sum_{i=1}^N \pi_i^* g^S$ can be computed by solving (23), and, thus, Algorithm 2 can be applied on (B_e^N, g^S) . In (17), one can also consider a retraction mapping instead of the exponential map. If one chooses the retraction (12) instead of the exponential maps $\exp_{u_i^k}$ in (15) for all $i = 1, \dots, N$ in Algorithm 2, one gets again the relation to the perturbation of the identity. In this setting, Theorem 1 justifies the update

$$D(u^{k+1}) = \{x \in D \mid x = x^k - t^k V^k\} \tag{27}$$

with $u^{k+1} = (u_1^{k+1}, \dots, u_N^{k+1})$ in the k -th iteration.

Remark 7. Notice that the variational problem given in (23) reflects exactly the approach presented, e.g., in Geiersbach et al. (2021), Siebenborn and Vogel (2021), and Siebenborn and Welker (2017) to generate descent directions for problems containing multiple shapes. Hence the above theory supports the numerical approach already used in those papers.

Stochastic Multi-shape Optimization and the Stochastic Gradient Method

Given the framework for understanding shape optimization problems over product shape spaces, it is now possible to incorporate uncertainty. In this section, the focus is on the case where the uncertainty can be characterized by a known probability space, for instance, through prior sampling. The probability space is a triple $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space containing all possible “realizations,” $\mathcal{F} \subset 2^\Omega$ is the σ -algebra of events, and $\mathbb{P}: \Omega \rightarrow [0, 1]$ is a probability measure. Note that in certain applications, there may be different sources of uncertainty that are independent of each other. In this case, one could work with the product probability

space $(\Omega, \mathcal{F}, \mathbb{P}) = (\Omega_1, \mathcal{F}_1, \mathbb{P}_1) \times \dots \times (\Omega_m, \mathcal{F}_m, \mathbb{P}_m)$ for m different sources of uncertainty. For the purposes of optimization, this structure is irrelevant and it is enough to refer to $(\Omega, \mathcal{F}, \mathbb{P})$.

To account for uncertainty, it is natural to parameterize the corresponding objective, which now depends on the probability space. A parametrized shape functional is defined by a function

$$J: \mathcal{U}^N \times \Omega \rightarrow \mathbb{R}, (u, \omega) \mapsto J(u, \omega).$$

Since J depends on ω , it is itself a random variable. To make the parameterized objective amenable to optimization, the following quantity

$$\mathbb{E}[J(u, \cdot)] := \int_{\Omega} J(u, \omega) \, d\mathbb{P}(\omega),$$

is used, i.e., the expectation or average. Other transformations of the parameterized objective are possible, for instance, by use of disutility functions or risk functions (see Shapiro et al. (2009) for an introduction). A *stochastic unconstrained shape optimization problem* is given by

$$\min_{u \in \mathcal{U}^N} j(u) := \mathbb{E}[J(u, \cdot)]. \tag{28}$$

Notice that the function j representing the transformed function J only depends on u , the vector of shapes. Therefore minimizers of (28) do not depend on ω , i.e., they are deterministic.

More interesting problems involve uncertainty in the equality constraint. The equality can be parameterized by the operator $e: \mathcal{U}^N \times \mathcal{Y} \times \Omega \rightarrow \mathcal{W}$, with Banach spaces \mathcal{Y} and \mathcal{W} . A property is said to hold almost surely (a.s.) provided that the set in Ω where the property does not hold is a null set. Of interest are constraints of the form

$$e(u, y, \omega) = 0 \quad \text{a.s.}$$

In other words, $\mathbb{P}(\{\omega \in \Omega : e(u, y, \omega) \neq 0\}) = 0$. The solution $y = y(\omega)$ of this equation is a *random state variable*. In applications, this belongs to the Bochner space $L^p(\Omega, \mathcal{Y})$, which, given $p \in [1, \infty)$, is defined to be the set of all (equivalence classes of) strongly measurable functions $y: \Omega \rightarrow \mathcal{Y}$ having finite norm, where the norm is defined by

$$\|y\|_{L^p(\Omega, \mathcal{Y})} := (\mathbb{E}[\|y\|_{\mathcal{Y}}^p])^{1/p} = \left(\int_{\Omega} \|y(\omega)\|_{\mathcal{Y}}^p \, d\mathbb{P}(\omega) \right)^{1/p}.$$

Letting the objective function depend on the state, a shape functional $\hat{J}: \mathcal{U}^N \times L^p(\Omega, \mathcal{Y}) \times \Omega \rightarrow \mathbb{R}$ is defined. With that, a *constrained stochastic shape*

optimization problem of the form

$$\begin{aligned} \min_{u \in \mathcal{U}^N, y \in L^p(\Omega, \mathcal{Y})} & \mathbb{E}[\hat{J}(u, y(\cdot), \cdot)] \\ \text{s.t.} & \quad e(u, y, \omega) = 0 \quad \text{a.s.} \end{aligned} \tag{29}$$

is obtained. If the equality constraint in (29) is uniquely solvable for any choice of $u \in \mathcal{U}^N$ and almost every $\omega \in \Omega$, then the operator $S(\omega) : \mathcal{U}^N \rightarrow \mathcal{Y}, u \mapsto y(\omega)$ is well-defined for almost every ω . As before, with $J(u, \omega) := \hat{J}(u, S(\omega)u, \omega)$, (29) is formally equivalent to the problem (28). This unconstrained view will be helpful in formulating the stochastic gradient method. However, the reader is reminded that the stochastic gradient implicitly depends on the operator $S(\cdot)$.

If the stochastic dimension is relatively small, the expectation can be approximated using quadrature and Algorithm 2 can be applied. This type of *sample average approximation* approach is not an algorithm, and it becomes intractable as the stochastic dimension grows. For larger stochastic dimensions, the stochastic gradient method is widely used in stochastic optimization. It is a classical method developed by Robbins and Monro (1951). As a sample-based approach, the stochastic gradient method does not suffer from the curse of dimensionality the way the discretizations mentioned in the introduction do. In Geiersbach et al. (2021), the stochastic gradient method was applied to the novel setting of shape spaces, where an example with multiple shapes was also presented. However, a theoretical background over product manifolds was not considered there. To apply the method to the setting containing multiple shapes, several concepts developed in section “Optimization of Multiple Shapes” need to be generalized. To this end, it will sometimes be helpful to use the shorthand $J_\omega(\cdot) := J(\cdot, \omega)$.

Definition 8 (Multi-pushforward for a fixed realization). For each point $u \in \mathcal{U}^N$, the multi-pushforward associated with $j : \mathcal{U}^N \times \Omega \rightarrow \mathbb{R}$ for a fixed realization $\omega \in \Omega$ is given by the map

$$((J_\omega)_*)_u : T_u \mathcal{U}^N \rightarrow \mathbb{R}, \quad \mathbf{c} \mapsto \frac{d}{dt} J_\omega(\mathbf{c}(t))|_{t=0} = (J_\omega \circ \mathbf{c})'(0).$$

Definition 9 (Stochastic Riemannian multi-shape gradient). The Riemannian multi-shape gradient for a parametrized shape functional $J : \mathcal{U}^N \times \Omega \rightarrow \mathbb{R}$ at the point $u = (u_1, \dots, u_N) \in \mathcal{U}^N$ is given by $v = v(\omega) \in T_u \mathcal{U}^N$ satisfying

$$\mathcal{G}_u^N(v, w) = ((J_\omega)_*)_u w \quad \forall w \in T_u \mathcal{U}^N.$$

Now, Definition 7 is generalized to incorporated uncertainties.

Definition 10 (Multi-shape derivative for a fixed realization). Let $D \subset \mathbb{R}^d$ be open, $u = (u_1, \dots, u_N)$, and observe an arbitrary admissible partition with $u_i \subset \Delta_i$ for all $i = 1, \dots, N$. Further, let $k \in \mathbb{N} \cup \{\infty\}$. For $i = 1, \dots, N$, the i -th partial Eulerian derivative of a shape functional J at u for a fixed realization $\omega \in \Omega$ in direction $W \in C_0^k(D, \mathbb{R}^d)$ is defined by

$$d_{u_i} J(u, \omega)[W|_{\Delta_i}] := \lim_{t \rightarrow 0^+} \frac{J(u_1, \dots, u_{i-1}, F_t^{W|_{\Delta_i}}(u_i), u_{i+1}, \dots, u_N, \omega) - J(u, \omega)}{t} \tag{30}$$

If for all directions $W \in C_0^k(D, \mathbb{R}^d)$ the i -th partial Eulerian derivative (30) exists and the mapping

$$C_0^k(D, \mathbb{R}^d) \rightarrow \mathbb{R}, \quad W \mapsto d_{u_i} J(u, \omega)[W|_{\Delta_i}]$$

is linear and continuous, the expression $d_{u_i} J(u, \omega)[W|_{\Delta_i}]$ is called the i -th partial shape derivative of j at u in direction $W \in C_0^k(D, \mathbb{R}^d)$. If the i -th partial shape derivatives of J at u for a fixed realization $\omega \in \Omega$ in the direction $W \in C_0^k(D, \mathbb{R}^d)$ exist for all $i = 1, \dots, N$, then

$$dJ(u, \omega)[W] := \sum_{i=1}^N d_{u_i} J(u, \omega)[W|_{\Delta_i}] \tag{31}$$

defines the multi-shape derivative of J at u for a fixed realization $\omega \in \Omega$ in direction $W \in C_0^k(D, \mathbb{R}^d)$.

Using identical arguments to those in Geiersbach et al. (2021, Lemma 2.14), it is possible to show under what conditions j is shape differentiable in u .

Lemma 1. *Suppose that $J(\cdot, \omega)$ is shape differentiable in u for almost every $\omega \in \Omega$. Assume there exist a $\tau > 0$ and a \mathbb{P} -integrable real function $C : \Omega \rightarrow \mathbb{R}$ such that for all $t \in [0, \tau]$, all $W \in C_0^\infty(D, \mathbb{R}^d)$, all $i = 1, \dots, N$, and almost every ω ,*

$$\frac{J(u_1, \dots, u_{i-1}, F_t^{W|_{\Delta_i}}(u_i), u_{i+1}, \dots, u_N, \omega) - J(u, \omega)}{t} \leq C(\omega).$$

Then j is shape differentiable in u and

$$dj(u)[W] = \mathbb{E}[dJ(u, \cdot)[W]] \quad \forall W \in C_0^\infty(D, \mathbb{R}^d).$$

Equipped with these tools, it is now possible to formulate the stochastic gradient method for objectives formulated on a product shape space in Algorithm 3. Instead of a backtracking procedure as in Algorithm 2 to determine the step-size, the algorithm uses the classical ‘‘Robbins–Monro’’ step-size from the original work

Robbins and Monro (1951):

$$t^k \geq 0, \quad \sum_{k=0}^{\infty} t^k = \infty, \quad \sum_{k=0}^{\infty} (t^k)^2 < \infty. \quad (32)$$

Under additional assumptions on the manifold and function J (cf. Geiersbach et al. 2021), this rule guarantees step-sizes that are large enough to converge to stationary points while asymptotically dampening oscillations in the iterates. In contrast to the backtracking procedure, the step-size sequence is in practice chosen exogenously, and its scaling is either informed by a priori estimates or tuned offline.

Algorithm 3 Stochastic gradient method on $(\mathcal{U}^N, \mathcal{G}^N)$ with Robbins–Monro step-size

Require: Objective function J on $(\mathcal{U}^N, \mathcal{G}^N)$

Input: Initial shape $u^0 = (u_1^0, \dots, u_N^0) \in \mathcal{U}^N$

for $k = 0, 1, \dots$ **do**

[1] Randomly sample ω^k , independent of $\omega^1, \dots, \omega^{k-1}$

[2] Compute the stochastic Riemannian multi-shape gradient $v^k = v^k(\omega^k)$ w.r.t. In this way the margins will be respected \mathcal{G}^N by

solving

$$((J_{\omega^k})_{*})_{u^k} w = G_{u^k}(v^k, w) \quad \forall w \in T_{u^k} \mathcal{U}^N.$$

[3] Set

$$u^{k+1} := \exp_{u^k}^N(-t^k v^k)$$

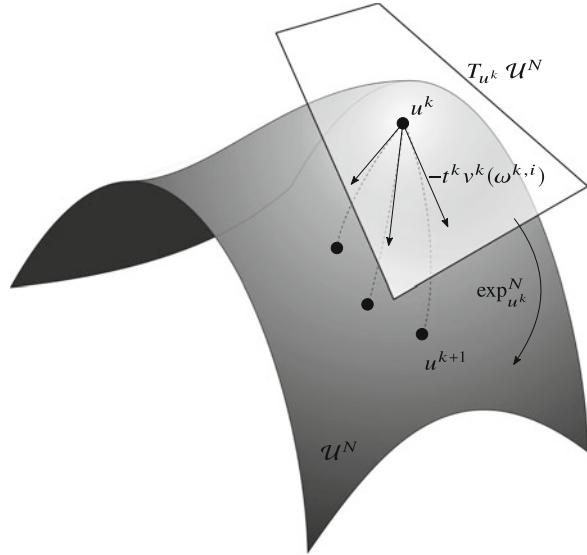
for a step length t^k satisfying (32).

end for

In Algorithm 3, a new random realization ω^k is generated at each iteration k . This is used to compute a stochastic gradient $v^k = v^k(\omega^k)$, which is then used as a descent direction for the objective functional $J(\cdot, \omega^k)$. If ω^k comprises a single sample from the probability space, the computation of the descent direction v^k is as cheap as in the deterministic case. Note that this is *not* necessarily a descent direction for the “true” objective j , which in combination with the exogenous step-size rule t^k does not guarantee descent at each iteration. The exponential map is used to map back to manifold (see Fig. 4).

Some comments on possible improvements to the simple Algorithm 3 in the context of shape spaces are in order. One might ask whether a backtracking

Fig. 4 Random iterates
 $u^{k+1} = \exp_{u^k}^N(-t^k v^k(\omega^{k,i}))$,
 where $\exp_{u^k}^N: T_{u^k} \mathcal{U}^N \rightarrow \mathcal{U}^N$



procedure could also be used for the stochastic setting; however, in Geiersbach (2020), it was demonstrated how the Armijo backtracking rule when combined with stochastic gradients fails in minimizing a function over the real line. Of course, there are modifications possible. In the most basic version of the method, ω^k comprises a single sample randomly drawn from the probability space. One might think that the problem could be remedied by simply taking multiple samples $\omega^k = (\omega^{k,1}, \dots, \omega^{k,m_k})$ at each iteration k and computing the empirical average

$$\nabla J(u^k, \omega^k) = \frac{1}{m_k} \sum_{i=1}^{m_k} \nabla J(u^k, \omega^{k,i}). \tag{33}$$

If m_k is constant, then it is however easy to modify the example from Geiersbach (2020) to show that simply taking more samples does not guarantee convergence of the method when paired with an Armijo backtracking procedure. Asymptotic convergence results are known if one is ready to take $m_k \rightarrow \infty$ (see Shapiro and Wardi 1996; Wardi 1990).

Nevertheless, taking batches of samples like (33) is a simple way to reduce the variance of the gradient and with that the iteration u^{k+1} . How the sampling sequence $\{m_k\}$ is to be chosen strongly depends on the structure of the problem (29) and the computational cost at each iteration n . In the context of optimal control problems with partial differential equations as constraints, one might additionally take into account that the computation is subject to numerical error as well. The authors in Martin et al. (2019) proposed a stochastic gradient step combined with a multilevel Monte Carlo scheme to reduce variance and numerical error. A method such as this one is sometimes referred to as a stochastic quasigradient method in

the literature to emphasize the numerical bias induced by the iteration. The analysis in Martin et al. (2019), which gives efficient choices for the sample size m_k , step-size t_k , and discretization error tolerance, works because the original problem is strongly convex, problem parameters are well-known, and the meshes involved are not deformed as part of the outer optimization loop. For more challenging problems, these choices no longer apply and future analysis would be needed.

Again for optimal control problems with PDEs, but for a larger class of problems, including nonsmooth and convex problems, the authors Geiersbach and Wollner (2020) propose a different approximation scheme without needing to take additional samples (meaning $m_k \equiv 1$ is permissible). The proposed method uses averaging of the *iterate* u^k instead of the stochastic gradient. The descent is smoothed indirectly without having to take additional samples at each iteration. This was shown to work efficiently in combination with a mesh refinement rule, carefully coupled with the step-size rule t^k . Extending these results to the context of shape optimization would also be challenging as well, not only due to the analysis of numerical error and lack of convexity; here, u^k represents a shape, not an element from a Banach space, and its “average” would need to be made precise.

A final connection to the shape space (B_e^N, \mathfrak{g}^S) is now desirable in view of the following numerical experiments. Using the theoretical justification from Theorem 1, it is possible to compute a deformation vector $V = V(\omega) \in H_0^1(D, \mathbb{R}^d)$ in the point $u = (u_1, \dots, u_N) \in B_e^N$ by solving the variational problem

$$a(V, W) = dJ(u, \omega)[W] \quad \forall W \in H_0^1(D, \mathbb{R}^d). \quad (34)$$

This deformation vector can be seen as an extension of the stochastic gradient $v = v(\omega)$ to the hold-all domain D . This stochastic deformation vector can then be used in the expression (27).

Numerical Investigations

In this section, the shape optimization model is formulated in order to demonstrate the algorithms. The deterministic model is given in section “[Deterministic Model Problem](#)”. Here, the focus is on a stationary version of the multi-shape model introduced in Siebenborn and Welker (2017). For the stochastic example in section “[Stochastic Model Problem](#)”, the model from Geiersbach et al. (2021) is used, with adjustments to include multiple shapes and random fields. There are several motivations for the models, for instance, the identification of cellular structures in biology (Siebenborn and Welker 2017) or electrical impedance tomography (Dambrine et al. 2019). In section “[Numerical Experiments](#)”, the results of the experiments are shown. In particular, the effectiveness and performance of Algorithms 2 and 3 are demonstrated. Moreover, an experiment on a single shape is done, which shows the robustness of a stochastic solution.

Deterministic Model Problem

Consider a partition of the domain D into $N + 1$ disjoint subdomains $D_i \subset D$ in such a way that $(\sqcup_{i=0}^N D_i) \sqcup (\sqcup_{i=1}^N u_i) = D$, where $u_i = \partial D_i$, $i = 1, \dots, N$ and \sqcup denotes the disjoint union. In particular, D depends on u , i.e., $D = D(u)$. Note that this partition is a new construction that is related to the physical model and is not to be confused with the arbitrary partition constructed in section “[Optimization of Multiple Shapes](#)”. For a given function $f : D \rightarrow \mathbb{R}$, f_i denotes the restriction $f|_{D_i} : D_i \rightarrow \mathbb{R}$. Additionally, $\mathbb{1}_{D_i}$ denotes the indicator function of the set D_i , meaning $\mathbb{1}_{D_i}(x) = 1$ if $x \in D_i$ and $\mathbb{1}_{D_i}(x) = 0$ if $x \notin D_i$.

Let $\bar{y} \in H^1(D)$ be the target distribution and $g \in L^2(\partial D)$ be a source term. The permeability coefficient is defined on each subdomain D_i by $\kappa_i \in C^1(D_i)$. The shorthand $\kappa := \sum_{i=0}^N \kappa_i \mathbb{1}_{D_i}$ will be useful in representing this function in the weak form.

In the following, the objective function

$$j(u) := j^{\text{obj}}(u) + j^{\text{reg}}(u)$$

with

$$j^{\text{obj}}(u) := \frac{1}{2} \int_D (y(x) - \bar{y}(x))^2 dx = \frac{1}{2} \sum_{i=0}^N \int_{D_i} (y_i(x) - \bar{y}_i(x))^2 dx, \tag{35}$$

$$j^{\text{reg}}(u) := \sum_{i=1}^N \nu_i \int_{u_i} dS \tag{36}$$

is considered. The tracking-type functional (35) gives the distance in $L^2(D)$ between the function y and the target \bar{y} . In (36), dS is used to characterize a surface integral. Note that the functional (36) regularizes the perimeter with respect to each shape and different choices for $\nu_i \geq 0$ can be made.

The following PDE-constrained problem in strong form is given:

$$\min_{u \in B_e^N} j(u) \tag{37}$$

$$\text{s.t.} \quad -\nabla \cdot (\kappa_i(x) \nabla y_i(x)) = 0 \quad \text{in } D_i, \quad i = 0, \dots, N, \tag{38}$$

$$\kappa_0(x) \frac{\partial y_0}{\partial \mathbf{n}_0}(x) = g(x) \quad \text{on } \partial D, \tag{39}$$

where \mathbf{n}_0 represents the outward normal vector on D_0 . The equations (38)–(39) are complemented by the transmission conditions

$$\kappa_i(x) \frac{\partial y_i}{\partial \mathbf{n}_i}(x) + \kappa_0(x) \frac{\partial y_0}{\partial \mathbf{n}_0}(x) = 0, \quad y_i(x) - y_0(x) = 0 \quad \text{on } u_i, \quad i = 1, \dots, N. \tag{40}$$

Note that the system (38), (39), and (40) can be compactly represented in the weak formulation: find $y \in H_{\text{av}}^1(D) := \{v \in H^1(D) \mid \int_D v \, dx = 0\}$ such that

$$\int_D \kappa(x) \nabla y(x) \cdot \nabla v(x) \, dx = \int_{\partial D} g(x) v(x) \, dx \quad \forall v \in H_{\text{av}}^1(D).$$

Remark 8. Thanks to Ito et al. (2008, Proposition 3.1), the regularity of $y_i, i = 0, \dots, N$, is better than the one of y . More precisely, the solution $y \in H_{\text{av}}^1(D)$ of (38), (39), and (40) satisfies $y_i \in H^2(D_i), i = 0, \dots, N$.

Remark 9. In general, the distribution \bar{y} and the diffusion coefficient κ do not need to have as high a regularity as assumed above to formulate the PDE-constrained problem (37), (38), (39), (40). The regularity above is only needed for shape differentiability of the objective functional (see Ito et al. 2008, Section 3.2).

The shape derivative to (37), (38), (39), (40) can be achieved using standard calculation techniques like the one mentioned in section “[Optimization on Shape Spaces with Steklov–Poincaré Metric](#)” combined with the help of the partial shape derivative definition and Remark 4. Its volume formulation is given by

$$\begin{aligned} & dj(u)[W] \\ &= \int_D -\kappa(x) \nabla y(x) \cdot (\nabla W(x) + \nabla W^\top(x)) \nabla p(x) - (y(x) - \bar{y}(x)) \nabla \bar{y}(x) \cdot W(x) \\ &\quad + (\nabla \kappa(x) \cdot W(x)) \nabla y(x) \cdot \nabla p(x) \\ &\quad + \operatorname{div}(W(x)) \left(\frac{1}{2} (y(x) - \bar{y}(x))^2 + \kappa(x) \nabla y(x) \cdot \nabla p(x) \right) \, dx \\ &\quad + \sum_{i=1}^N v_i \int_{u_i} \mathbf{v}_i(x) W(x) \cdot \mathbf{n}_i(x) \, dS, \end{aligned} \tag{41}$$

where \mathbf{v}_i and denotes the curvature of the shape $u_i, i = 1, \dots, N$, $y(x)$ satisfies the state equation (38), (39), and (40), and $p(x)$ satisfies adjoint equation given in strong form by

$$-\nabla \cdot (\kappa_i(x) \nabla p_i(x)) = \bar{y}(x) - y_i(x) \quad \text{in } D_i, \quad i = 0, \dots, N, \tag{42}$$

$$\kappa_0(x) \frac{\partial p_0}{\partial \mathbf{n}_0}(x) = 0 \quad \text{on } \partial D \tag{43}$$

with the corresponding transmission conditions

$$\kappa_i(x) \frac{\partial p_i}{\partial \mathbf{n}_i}(x) + \kappa_0(x) \frac{\partial p_0}{\partial \mathbf{n}_0}(x) = 0, \quad p_i(x) - p_0(x) = 0 \quad \text{on } u_i, \quad i = 1, \dots, N. \tag{44}$$

The sum of integrals over u_i in (41) is the shape derivative of the perimeter regularization, which is computed with the help of the partial shape derivative definition as follows:

$$dj^{\text{reg}}(u)[W] = \frac{d^+}{dt} \Big|_{t=0} \sum_{i=1}^N v_i \int_{F_i^{W|\Delta_i}(u_i)} dS,$$

where the ℓ -th partial shape derivative of j^{reg} at u in direction W is given by

$$\begin{aligned} d_{u_\ell} j^{\text{reg}}(u)[W|\Delta_\ell] &= \frac{d^+}{dt} \Big|_{t=0} \left(\sum_{\substack{i=1 \\ i \neq \ell}}^N v_i \int_{u_i} dS \right) \\ &\quad + v_\ell \frac{d^+}{dt} \Big|_{t=0} \int_{F_i^{W|\Delta_\ell}(u_\ell)} dS = v_j \frac{d^+}{dt} \Big|_{t=0} \int_{F_i^{W|\Delta_\ell}(u_\ell)} dS \\ &= \int_{u_\ell} \mathbf{v}_\ell(x) W|\Delta_\ell(x) \cdot \mathbf{n}_\ell(x) \, dS, \end{aligned}$$

where the last equality holds, thanks to Novruzi and Pierre (2002, Proposition 5.1). This gives the ℓ -th partial shape derivative $d_{u_\ell} j^{\text{reg}}(u)[W|\Delta_\ell]$ and thus the shape derivative of the regularization term in (41).

Now, every object needed for the application of Algorithm 2 is given. In section “Numerical Experiments”, this algorithm is applied to solve the deterministic model problem.

Stochastic Model Problem

For the stochastic model, the domain D is partitioned as described for the deterministic model above. For a function $f : D \times \Omega \rightarrow \mathbb{R}$, the function f_i denotes the restriction $f|_{D_i} : D_i \times \Omega \rightarrow \mathbb{R}$. The slightly abusive notation $\nabla f_i(x, \omega) = \nabla_x f_i(x, \omega)$ means ω is fixed and the gradient is to be understood with respect to the variable x only. Additionally, the notation for the directional derivative means $\frac{\partial f_i}{\partial \mathbf{n}_i}(x, \omega) = \lim_{t \rightarrow 0} \frac{1}{t} (f_i(x + t \mathbf{n}_i(x), \omega) - f_i(x, \omega))$. A parametrized objective function is now given by

$$J(u, \omega) := J^{\text{obj}}(u, \omega) + J^{\text{reg}}(u),$$

where

$$J^{\text{obj}}(u, \omega) := \frac{1}{2} \int_D (y(x, \omega) - \bar{y}(x))^2 dx = \frac{1}{2} \sum_{i=0}^N \int_{D_i} (y_i(x, \omega) - \bar{y}_i(x))^2 dx \tag{45}$$

and J^{reg} is defined as in (36). For simplicity, the source term g and the target term \bar{y} are deterministic with the same regularity as in the previous section. Suppose however that the source of uncertainty comes from the coefficients, i.e., $\kappa_i = \kappa_i(x, \omega)$ are random fields with regularity $\kappa_i \in L^2(\Omega, C^1(D_i))$. This leads to a modification of the deterministic problem

$$\min_{u \in B_e^N} \left\{ j(u) := \mathbb{E}[J(u, \omega)] \right\} \tag{46}$$

$$\text{s.t.} \quad -\nabla \cdot (\kappa_i(x, \omega) \nabla y_i(x, \omega)) = 0 \quad \text{in } D_i \times \Omega, \quad i = 0, \dots, N, \tag{47}$$

$$\kappa_0(x, \omega) \frac{\partial y_0}{\partial \mathbf{n}_0}(x, \omega) = g(x) \quad \text{on } \partial D \times \Omega \tag{48}$$

The following transmission conditions are also imposed:

$$\begin{aligned} \kappa_i(x, \omega) \frac{\partial y_i}{\partial \mathbf{n}_i}(x, \omega) + \kappa_0(x, \omega) \frac{\partial y_0}{\partial \mathbf{n}_0}(x, \omega) &= 0 && \text{on } u_i \times \Omega, \quad i = 1, \dots, N, \\ y_i(x, \omega) - y_0(x, \omega) &= 0 && \text{on } u_i \times \Omega, \quad i = 1, \dots, N. \end{aligned} \tag{49}$$

Using standard techniques for calculating the shape derivative (see Geiersbach et al. 2021, Appendix B), the shape derivative in volume formulation for a fixed ω is given by

$$\begin{aligned} dJ(u, \omega)[W] &= \int_D -\kappa(x, \omega) \nabla y(x, \omega) \cdot (\nabla W(x) + \nabla W^\top(x)) \nabla p(x, \omega) \\ &\quad - (y(x, \omega) - \bar{y}(x)) \nabla \bar{y}(x) \cdot W(x) + (\nabla \kappa(x, \omega) \cdot W(x)) \nabla y(x, \omega) \cdot \nabla p(x, \omega) \\ &\quad + \text{div}(W(x)) \left(\frac{1}{2} (y(x, \omega) - \bar{y}(x))^2 + \kappa(x, \omega) \nabla y(x, \omega) \cdot \nabla p(x, \omega) \right) dx \\ &\quad + \sum_{i=1}^N v_i \int_{u_i} \mathbf{v}_i(x) W(x) \cdot \mathbf{n}_i(x) dS, \end{aligned}$$

where $y = y(x, \omega)$ satisfies the state equation (47), (48), and (49) and $p = p(x, \omega)$ satisfies adjoint equation

$$-\nabla \cdot (\kappa_i(x, \omega) \nabla p_i(x, \omega)) = \bar{y}(x) - y_i(x, \omega), \quad \text{in } D_i \times \Omega, \quad i = 0, \dots, N, \tag{50}$$

$$\kappa_0(x, \omega) \frac{\partial p_0}{\partial \mathbf{n}_0}(x, \omega) = 0, \quad \text{on } \partial D \times \Omega, \tag{51}$$

with corresponding interface conditions

$$\begin{aligned} \kappa_i(x, \omega) \frac{\partial p_i}{\partial \mathbf{n}_i}(x, \omega) + \kappa_0(x, \omega) \frac{\partial p_0}{\partial \mathbf{n}_0}(x, \omega) &= 0 && \text{on } u_i \times \Omega, i = 1, \dots, N, \\ p_i(x, \omega) - p_0(x, \omega) &= 0 && \text{on } u_i \times \Omega, i = 1, \dots, N. \end{aligned} \tag{52}$$

The construction of the coefficients κ for the purpose of simulations requires some discussion. Karhunen–Loève expansions are frequently used to simulation random perturbations of a coefficient within a material and are also used in the experiments in section “[Numerical Experiments](#)”. Given a domain \tilde{D} , a (truncated) Karhunen–Loève expansion of a random field $a: \tilde{D} \times \Omega \rightarrow \mathbb{R}$ takes the form

$$a(x, \omega) = \bar{a}(x) + \sum_{k=1}^m \sqrt{\gamma_k} \phi_k(x) \xi_k(\omega),$$

where $\bar{a}: \tilde{D} \rightarrow \mathbb{R}$ and $\xi(\omega) = (\xi_1(\omega), \dots, \xi_m(\omega)) \in \mathbb{R}^m$ is a random vector. The truncation is done for the purposes of numerical simulation and the choice of m should be informed by error analysis. The terms γ_k and ϕ_k are eigenvalues and eigenfunctions that depend on the domain \tilde{D} . In particular, they are associated with the compact self-adjoint operator defined via the covariance function $C \in L^2(\tilde{D} \times \tilde{D})$ by $C(\phi)(x) = \int_{\tilde{D}} C(x, y) \phi(y) dy$ for all $x \in \tilde{D}$. For general domains, formulas giving explicit representations of γ_k and ϕ_k do not exist and need to be numerically computed. However, since the subdomains vary as part of the optimization procedure, their computation here would be extremely expensive. Moreover, from a modeling perspective, it seems more realistic that the model for uncertainty in a specific material is constructed beforehand using samples on a fixed domain $\tilde{D} \supset D_i$. Ideally \tilde{D} should be much larger than D_i to limit the effects of the boundary of the larger domain on the sample. Then, to approximate κ_i on D_i , one can first produce a sample on the larger domain \tilde{D} and then use its restriction on the domain D_i for computations. To be more precise, one would first define over \tilde{D}

$$\tilde{\kappa}_i(x, \omega) = \bar{\kappa}_i(x) + \sum_{k=1}^{m_i} \sqrt{\gamma_{i,k}} \phi_{i,k}(x) \xi_{i,k}(\omega), \tag{53}$$

where $\bar{\kappa}: \tilde{D} \rightarrow \mathbb{R}$, $\xi_{i,k}(\omega) = (\xi_{i,1}(\omega), \dots, \xi_{i,m_i}(\omega)) \in \mathbb{R}^{m_i}$ is a random vector, and $\gamma_{i,k}$ and $\phi_{i,k}$ denote the eigenvalues and eigenfunctions that depend on the domain

\tilde{D} . Finally, $\kappa_i = \tilde{\kappa}_i|_{D_i}$. The coefficient κ over the domain D is then stitched together by definition of

$$\kappa(x, \omega) = \kappa_0(x, \omega) + \sum_{i=1}^N \kappa_i(x, \omega) \mathbb{1}_{D_i}(x).$$

An example of this construction is shown in the next subsection in Fig. 9.

Numerical Experiments

The purpose of this section is to demonstrate the behavior and performance of Algorithms 2 and 3. Simulations were run on FEniCS (Alnæs et al. 2015). For all experiments, the hold-all domain is set to $D = [0, 1]^2$ and a mesh with 2183 nodes and 4508 elements is used.

For methods relying on mesh deformation, one challenge is to ensure that meshes maintain good quality and do not become destroyed over the course of optimization. Many techniques have been developed along the years to overcome this challenge. There is the option of remeshing (see, for instance, Morin et al. 2012, Sturm 2016, and Feppon et al. 2019). Of course, one could also use mesh regularization techniques and space adaptivity, among others as described, for example, in Bänsch et al. (2005) and Doğan et al. (2007). There is also the possibility of projecting the descent directions onto the subspace of perturbation fields generated only by normal forces, inspired by the Hadamard structure theorem (Etling et al. 2020). Following the Riemannian setting one could define Riemannian metrics whose main aim is to preserve the quality of the meshes as the one proposed in Herzog and Loayza-Romero (2020). Recently, a simultaneous shape and mesh quality optimization approach based on pre-shape calculus has also been proposed (Luft and Schulz 2021a,b). Another option is to consider the method of mappings and impose certain restrictions on the maps that preserve mesh quality (see Haubner et al. 2020; Onyshkevych and Siebenborn 2021).

In this chapter, the techniques developed in Schulz et al. (2016) and Schulz and Siebenborn (2016) are considered. As discussed in Schulz et al. (2016), an unmodified right-hand side of the discretized deformation equation leads to deformation fields causing meshes with bad aspect ratios. One possibility is to set the values of the shape derivative to zero if the corresponding element does not intersect with the shapes, i.e.,

$$dj(u)[W] = 0 \quad \forall W \text{ with } \text{supp}(W) \cap u_i = \emptyset, \quad i = 1, \dots, N.$$

Additionally, following the ideas from Schulz and Siebenborn (2016), at each iteration k , an additional PDE is solved to choose values for the Lamé parameters in the deformation equation. The parameter λ is set to zero, and μ is chosen from the interval $[\mu_{\min}, \mu_{\max}]$ such that it is decreasing smoothly from u_i , $i = 1, \dots, N$, to the outer boundary ∂D . One possible way to model this behavior is to solve the Poisson equation

$$\begin{aligned} \Delta\mu &= 0 && \text{in } D_i, \quad i = 0, \dots, N \\ \mu &= \mu_{\max} && \text{on } u_i, \quad i = 1, \dots, N, \\ \mu &= \mu_{\min} && \text{on } \partial D. \end{aligned}$$

In all experiments, $\mu_{\min} = 10$ and $\mu_{\max} = 25$ is chosen.

Deterministic Case: Behavior of Algorithm 2

The deterministic shape optimization problem formulated in section “[Deterministic Model Problem](#)” is considered to demonstrate the behavior of Algorithm 2. For the numerical experiments, an example with two shapes is used, i.e., $N = 2$, and the algorithm runs for 400 iterations. The Neumann boundary condition in (37), (38), (39), and (40) is set to $g = 1000$, and the perimeter regularization is set to $\nu_1 = \nu_2 = 2 \cdot 10^{-5}$.

In order to generate the target data \bar{y} in the tracking-type objective functional (37), a target shape vector $u^* = (u_1^*, u_2^*)$ is chosen, which is displayed in dotted lines in Fig. 5. The target shapes, i.e., an ellipse and a (non-convex) curved tube, are chosen, so the configuration is non-symmetric, making their identification more difficult. The permeability coefficients are assumed to be piecewise constant on each subdomain with the choices $\kappa_0 = 1000$ for the outer domain D_0^* , $\kappa_1 = 7.5$ corresponding to the ellipse D_1^* , and $\kappa_2 = 5$ corresponding to the curved tube D_2^* . The data \bar{y} is computed by solving the state equation (38), (39), and (40) on the target configuration $D^* = (\sqcup_{i=0}^2 D_i^*) \sqcup (\sqcup_{i=1}^2 u_i^*)$ (see Fig. 6).

Let $D^k = (\sqcup_{i=0}^2 D_i^k) \sqcup (\sqcup_{i=1}^2 u_i^k)$ be the configuration of the subdomains at iteration k . The subdomains D_i^k correspond to the different colors in Fig. 5. As for the computation for the target distribution, the coefficients are assumed to be piecewise constant on each subdomain with the choices $\kappa_0 = 1000$ for the outer domain D_0^k , $\kappa_1 = 7.5$ corresponding to D_1^k , and $\kappa_2 = 5$ corresponding to D_2^k . For the Armijo rule, the values $\hat{\alpha} = 0.0175$, $\rho = 0.9$, and $\sigma = 10^{-4}$ are used. Since the algorithm is designed to deform the mesh, the initial step-size $\hat{\alpha}$ is scaled to be proportional to the maximal diameter of the elements, which is used as a heuristic solution to avoid mesh destruction. Figure 5 shows the progression of the subdomains. Within 400 iterations, one sees that the configuration D^k obtained by the method comes quite close to the target. Figure 7 gives a visualization of the vector fields V^k induced by solving the deformation equation (23). In Fig. 8 one sees the decay of the objective function values and the H^1 -norm of the deformation vector as a function of iteration number. The Armijo line search procedure ensures that $j(u^{k+1}) \leq j(u^k)$ for all k . The H^1 -norm of the descent directions serves as a stationary measure, and the plots show decreasing as a function of the iterations.

Stochastic Case: Behavior of Algorithm 3

Similar experiments to the one in section “[Deterministic Case: Behavior of Algorithm 2](#)” are now shown. These experiments use the stochastic model formulated in section “[Stochastic Model Problem](#)” to demonstrate the performance of

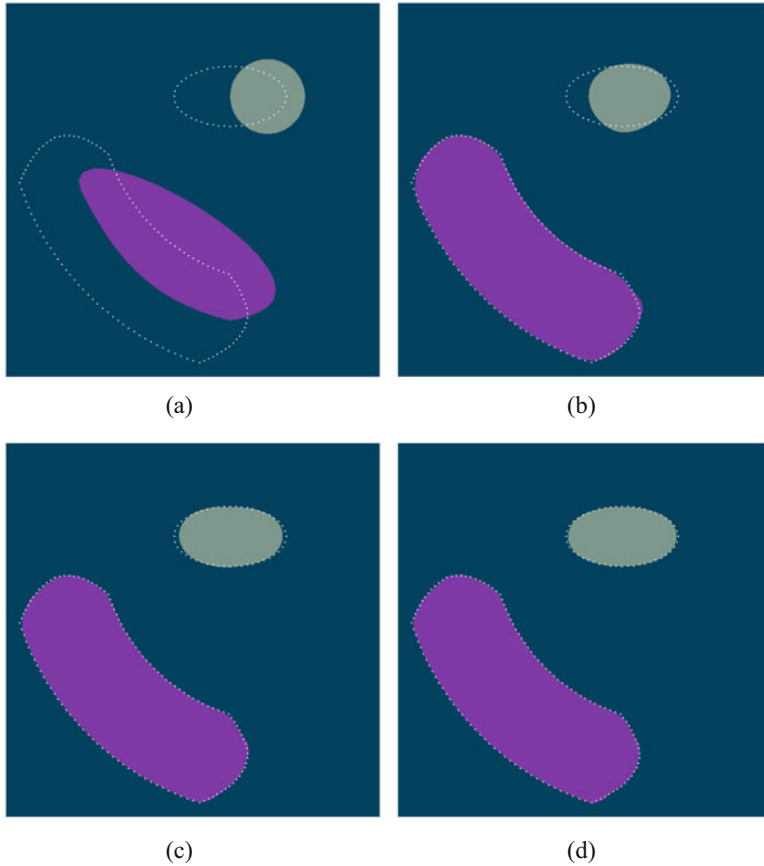


Fig. 5 The target shapes are displayed by the dotted lines. The outer domain D_0^k is displayed in teal, the domain D_1^k is displayed in light green, and the subdomain D_2^k is shown in purple. The figures show the progression of the initial configuration D^0 to the final subdomain configuration D^{400} . (a) Initial configuration D^0 . (b) D^{50} . (c) D^{200} . (d) D^{400}

Algorithm 3. An example with two shapes is used again, i.e., $N = 2$, and the same target shape vector u^* as in section “Deterministic Case: Behavior of Algorithm 2” is considered. The same values for g and $\nu_1 = \nu_2$ are used.

To generate samples according to the discussion at the end of section “Stochastic Model Problem”, for simplicity $\tilde{D} = D$ is used, allowing for the explicit representations of the eigenfunctions and eigenvalues in (53). From Lord et al. 2014, Example 9.37, the eigenfunctions and eigenvalues on D are given by the formula

$$\tilde{\phi}_j^k(x) := 2 \cos(j\pi x_2) \cos(k\pi x_1), \quad \tilde{\gamma}_j^k := \frac{1}{4} \exp(-\pi(j^2 + k^2)l^2), \quad j, k \geq 1,$$

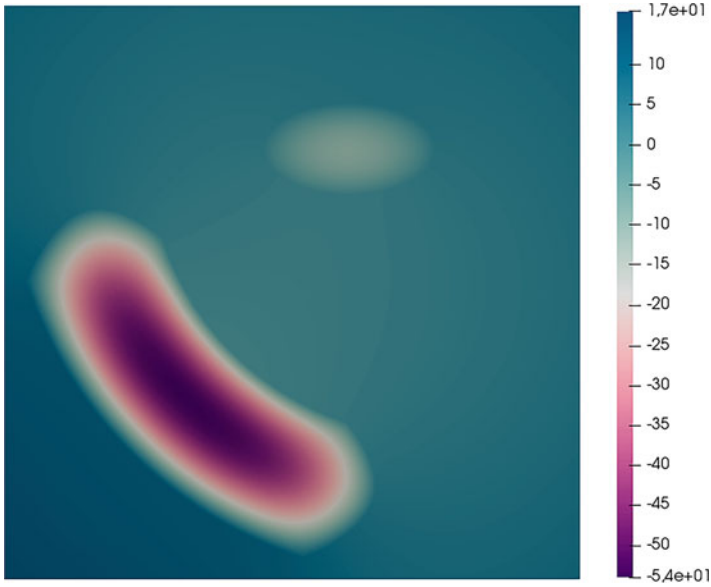


Fig. 6 Values of the target data \bar{y}

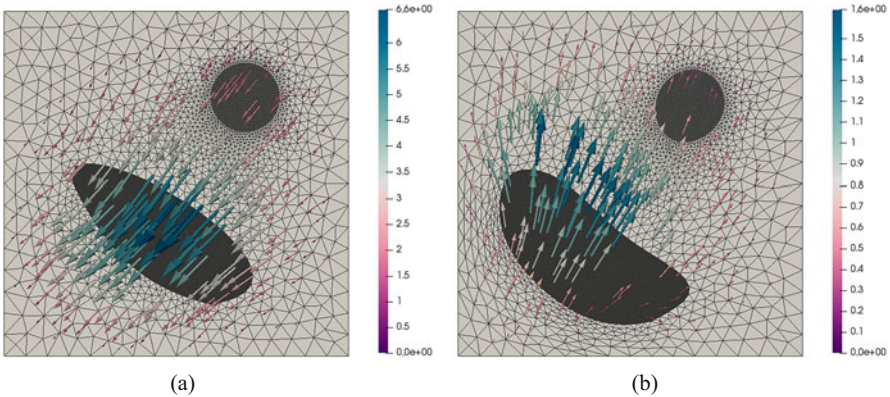


Fig. 7 Vector fields V^k are displayed that result from solving the deformation equation (23) at iteration k . (a) Vector field V^0 . (b) Vector field V^3

where terms are then reordered so that the eigenvalues appear in descending order (i.e., $\phi_1 = \tilde{\phi}_1^1$ and $\lambda_1 = \tilde{\lambda}_1^1$). The correlation length $l = 0.5$ and the number of summands $M = 20$ are fixed. For the simplicity of presentation, each subdomain has the same eigenfunctions and eigenvalues, and only the means and random vectors are modified. More precisely, (53) has the representation

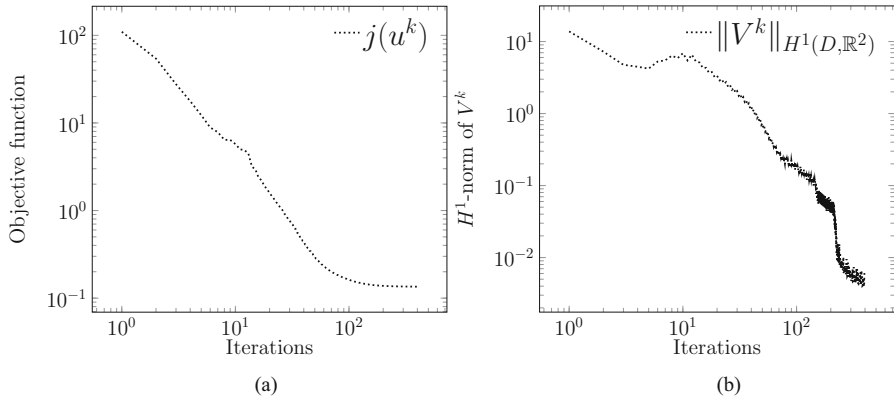


Fig. 8 Objective function and norm of the shape gradient as a function of iteration number (log/log scale). (a) Objective function decay. (b) Deformation vector field

$$\tilde{\kappa}_i(x, \omega) = \bar{\kappa}_i(x) + \sum_{k=1}^{20} \sqrt{\gamma_k} \phi_k(x) \xi_{i,k}(\omega), \tag{54}$$

for every $i = 0, 1, 2$. Using the same labeling convention as in the deterministic study, the values $\bar{\kappa}_0 = 1000$, $\bar{\kappa}_1 = 7.5$, and $\bar{\kappa}_2 = 5$ are used for the mean in the outer, ellipse, and tube domains, respectively. Notice that these are compatible with the choices used in the deterministic experiment. Deviations from this mean are simulated using the centered distributions $\xi_{0,k} \sim U[-50, 50]$, $\xi_{1,k} \sim U[-2.5, 2.5]$, $\xi_{2,k} \sim U[-1, 1]$, with $U[a, b]$ standing for the uniform distribution on the interval $[a, b] \subset \mathbb{R}$. Figure 9 shows two examples of the random fields. Since these are shown for different iterations, one also sees how a single sample in the definition of κ is adapted to the movement of the shapes.

The target \bar{y} in the objective functional (45) is computed by solving the deterministic state equation (38), (39), and (40) on the target configuration with the mean values $\bar{\kappa}_0$, $\bar{\kappa}_1$, and $\bar{\kappa}_2$ on the target configuration $D^* = (\sqcup_{i=0}^2 D_i^*) \sqcup (\sqcup_{i=1}^2 u_i^*)$. The target is the same as in section “Deterministic Case: Behavior of Algorithm 2” (see Fig. 6).

Regarding the choice of the step-size according to (32), experiments showed that a rule of the form $t^k = c/k$ performed poorly in practice. This is mostly due to the fact that the choice c is limited by the fineness of the mesh; if this parameter is chosen to be too large, then the mesh deforms too drastically in the first few iterations, leading to broken meshes. However, if c is chosen to be too small, the progress—although guaranteed to produce stationary points in the limit—is much too slow. To mitigate this effect, a warm start of 250 iterations using the constant step-size $t^k = c = 0.015$ is used until the shapes appear to be in the neighborhood of the optimum. Then the rule $t^k = c/(k - 250)$ is used for $k = 251, \dots, 400$. This

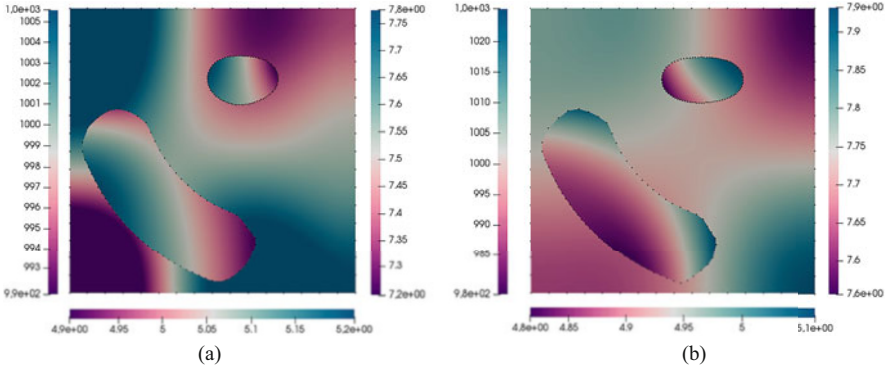


Fig. 9 Two examples of random field κ , with the left, right, and bottom scales corresponding to the outer domain D_0^k , the ellipse D_1^k , and the tube D_2^k , respectively. (a) Example realization of the κ at iteration $k = 100$. (b) Example realization of the κ at iteration $k = 300$

produces excellent results as shown in Fig. 10. Even in the presence of noise, the progression of the subdomains resembles that shown in Fig. 5.

Figure 11 provides a stochastic counterpart to Fig. 8, in which one sees the progression of the parametrized functional $J(u^k, \omega^k)$ as well as the vector field $V^k = V^k(\omega^k)$, where ω^k represents the abstract realization from the probability space in iteration k , which is manifested by the specific realizations of the random vectors $(\xi_{i,1}(\omega^k), \dots, \xi_{i,20}(\omega^k))$, $i = 0, 1, 2$, used in the random fields. In contrast to the Armijo line search rule, the Robbins–Monro step-size rule does not guarantee descent in every iteration. Moreover, the information displayed in the plots can only provide estimates for the true objective $j(u^k) = \mathbb{E}[J(u^k, \cdot)]$ and the average $\mathbb{E}[\|V^k(\cdot)\|_{H^1(D, \mathbb{R}^2)}]$. Although small oscillations in the shapes were observed in the course of the algorithm, the oscillations from the plots come more from the stochastic error occurring due to $J(u^k, \omega^k) \approx \mathbb{E}[J(u^k, \cdot)]$ and $\|V^k(\omega^k)\|_{H^1(D, \mathbb{R}^2)} \approx \mathbb{E}[\|V^k(\omega)\|_{H^1(D, \mathbb{R}^2)}]$. The log/log scale misleadingly exaggerates these oscillations for higher iteration numbers and the Robbins–Monro step-size rule tended to dampen oscillations in the shapes for higher iterations. However, even with the oscillations, descent is seen *on average* in both the parametrized objective and in the H^1 -norm of the randomly generated deformation vector fields.

Robustness: Deterministic vs. Stochastic Model

A final experiment justifies the use of the stochastic model if experimental parameters are uncertain. To demonstrate the concept, only a single shape is used, i.e., $N = 1$. The perimeter regularization is fixed with $\nu = 5 \cdot 10^{-2}$. The expansion (54) is used for $i = 0, 1$ with the same eigenfunctions, eigenvalues, and choices of the correlation length l and number of summands M . In each iteration k , on the outer domain D_0^k , the mean is given by $\bar{\kappa}_0 = 1000$ and distribution is chosen to be $\xi_{0,k} \sim U[-75, 75]$. On the domain D_1^k , the mean and distribution are given by $\bar{\kappa}_1 = 7.5$ and $\xi_{1,k} \sim U[-4.5, 4.5]$.

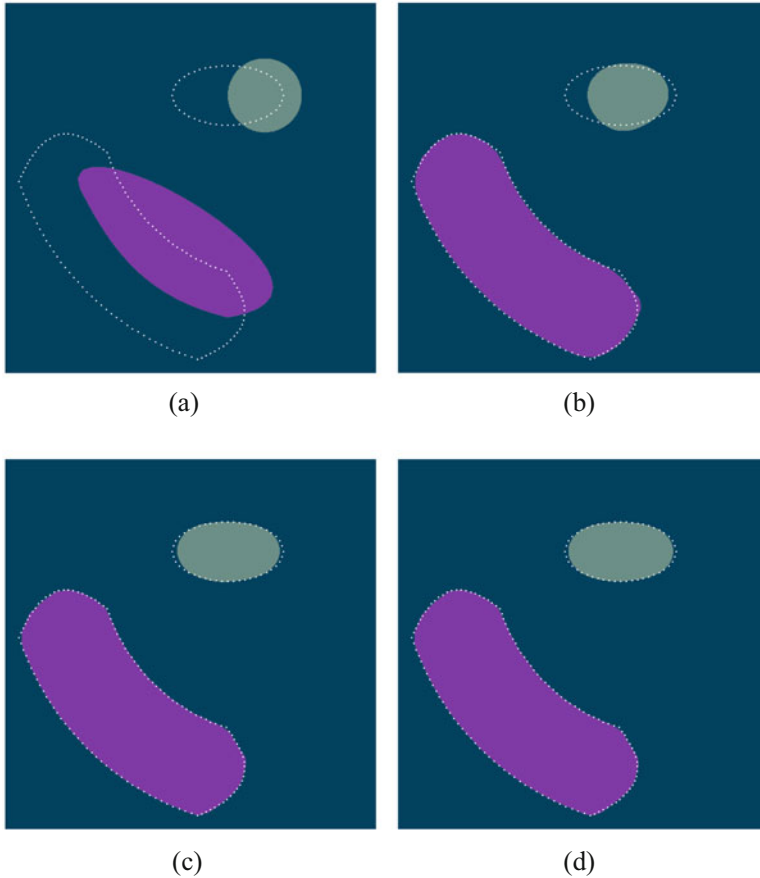


Fig. 10 The target shapes are displayed by the dotted lines. The figures show the progression from the initial configuration of domains D^0 to the final configuration of domains D^{400} . (a) Initial configuration D^0 . (b) D^{50} . (c) D^{200} . (d) D^{400}

For the generation of the target data \bar{y} in the tracking-type objective functional, the target shape u^* is chosen to be the boundary of an ellipse as illustrated by the dotted lines in Fig. 12. The target distribution \bar{y} is computed on the target domain $D^* = D_0^* \sqcup D_1^* \sqcup u^*$ by solving the state equation (38), (39), and (40) using the constant values $\bar{\kappa}_0 = 1000$ over the outer domain D_0^* and $\bar{\kappa}_1 = 7.5$ defined over the ellipse D_1^* . The target data can be seen in Fig. 13. As in the previous experiments, algorithms are run for 400 iterations. The results of the simulation are shown in Fig. 12, where the target shape u^* is represented by dotted lines. The same initial configuration, shown in Fig. 12a, is used for three separate runs of the algorithm.

In the first run, the stochastic model with the parameters described in the previous paragraph is used, and the stochastic gradient method (Algorithm 3) is used with the step-size rule $t^k = 0.026$ for $k = 0, \dots, 200$, and $t^k = 0.026/(k - 200)$ for

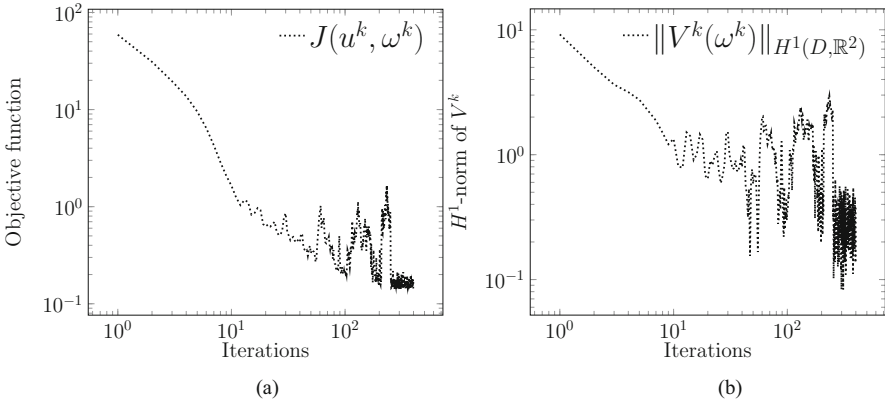


Fig. 11 Objective function and norm of the shape gradient as a function of iteration number (log/log scale). (a) Objective function decay. (b) Deformation vector field

$k = 201, \dots, 400$. The configuration obtained at 400 iterations approximates the desired configuration nicely as shown in Fig. 12b.

Incorrect choices for the parameters are used for the next two runs. In the disastrous case, where these parameters are incorrectly chosen at the upper or lower limits of the probability distributions, the deterministic Algorithm 2 does not correctly identify the desired shape u^* . Using the choices $\kappa_{0,\min} = 937.3$ and $\kappa_{1,\min} = 3.7$, which are chosen in such a way such that $\kappa_{i,\min} \leq \kappa_i(x, \omega)$ for all $(x, \omega) \in D \times \Omega, i = 0, 1$, produces a result as shown in Fig. 12c. Alternatively, with the choices $\kappa_{0,\max} = 1062.7$ and $\kappa_{1,\max} = 11.3$, analogously chosen so that $\kappa_{i,\max} \geq \kappa_i(x, \omega)$ for all $(x, \omega) \in D \times \Omega, i = 0, 1$, results in the configuration shown in Fig. 12d. One clearly sees in both Fig. 12c and d that the correct shape is not identified, even for this very simple example. In summary, when parameters are subject to uncertainty, but a good model for the uncertainty is available, it is always better to use the stochastic model. The corresponding solution to the stochastic model is robust with respect to these uncertainties.

Conclusion

This chapter gives an overview how the theory of (PDE-constrained) shape optimization can be connected with the differential geometric structure of shape space and how this theory can be adapted to handle harder problems containing multiple shapes and uncertainties. The framework presented is focused on shape spaces as Riemannian manifolds, in particular, on the space of smooth shapes and the Steklov–Poincaré metric. The Steklov–Poincaré metric allows for the usage of the shape derivative in its volume expression in optimization methods. A novel framework developed in this chapter is a product shape, which allows for shape optimization over a vector of shapes. As part of this framework, new concepts including the

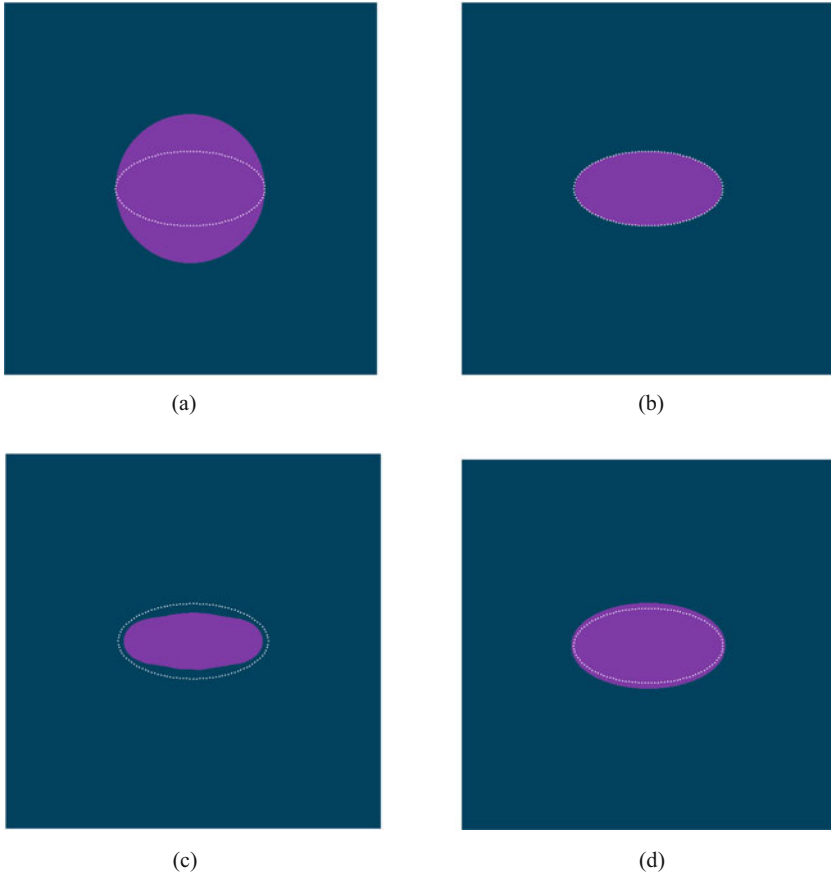
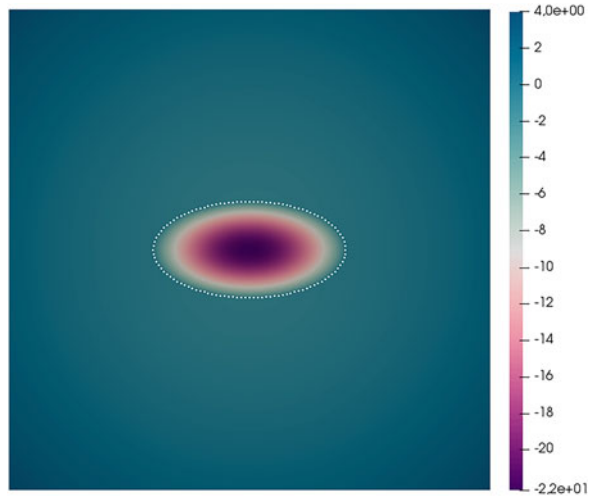


Fig. 12 The figures show the initial configuration in **(a)** and the configuration computed using the stochastic model and stochastic gradient approach in **(b)**. Using the lower bound choices produces an incorrect identification in **(c)**; with the upper bound choices, the target shape is likewise incorrectly identified. **(a)** Initial configuration D^0 . **(b)** D^{400} for stochastic model. **(c)** D^{400} produced using the constants $\kappa_{i,\min}$, $i = 0, 1$. **(d)** D^{400} using the constants $\kappa_{i,\max}$, $i = 0, 1$

partial and multi-shape derivatives are presented. The steepest descent method with Armijo backtracking on product shape spaces is formulated to solve a shape optimization problem over a vector of shapes.

The second area of focus in this chapter is concerned with shape optimization problems subject to uncertainty. The problem is posed as a minimization of the expectation of a random objective functional depending on uncertain parameters. Using the product shape space framework, it is no trouble to consider stochastic shape optimization problems depending on shape vectors. Corresponding definitions for the stochastic partial and multi-shape gradient are presented. These are needed to present the stochastic gradient method on product shape spaces. It is

Fig. 13 Values of the target data \bar{y}



discussed how the stochastic shape derivative in its volume expression can be used algorithmically.

The final part of the chapter is dedicated to carefully designed numerical simulations showing the performance of the algorithms. Compatible deterministic and stochastic problems are presented. A novel technique for producing stochastic samples of the Karhunen–Loève type is presented. The stochastic model is shown in experiments to be robust if a model for the uncertainties is present.

The new framework provides a rigorous justification for computing descent vectors “all-at-once” on a hold-all domain. Moreover, new concepts like the partial shape derivatives and multi-shape derivatives provide tools that could be used in other applications. There are some open questions; for one, it is not clear how descent directions in general prevent shapes from intersecting as part of the optimization procedure. Mesh deformation methods like the kind used here would result in broken meshes. While the algorithms presented do not rely on remeshing, it is notable that meshes lose their integrity if initial shapes are chosen too far away from the target. These challenges will be addressed in other works.

References

- Abraham, R., Marsden, J., Ratiu, T.: *Manifolds, tensor analysis, and applications*, vol. 75. Springer Science & Business Media, New York, USA (2012)
- Absil, P., Mahony, R., Sepulchre, R.: *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, USA (2008)
- Alnæs, M.S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The FEniCS project version 1.5. *Arch. Numer. Softw.* **3**(100) (2015). <https://doi.org/10.11588/ans.2015.100.20553>
- Babuska, I., Tempone, R., Zouraris, G.: Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.* **42**(2), 800–825 (2004)

- Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **45**(3), 1005–1034 (2007)
- Bänsch, E., Morin, P., Nochetto, R.H.: A finite element method for surface diffusion: the parametric case. *J. Comp. Phys.* **203**(1), 321–343 (2005). <https://doi.org/10.1016/j.jcp.2004.08.022>
- Bauer, M., Harms, P., Michor, P.: Sobolev metrics on shape space of surfaces. *J. Geom. Mech.* **3**(4), 389–438 (2011)
- Bauer, M., Harms, P., Michor, P.: Sobolev metrics on shape space II: weighted Sobolev metrics and almost local metrics. *J. Geom. Mech.* **4**(4), 365–383 (2012)
- Berggren, M.: A unified discrete-continuous sensitivity analysis method for shape optimization. In: Fitzgibbon, W., et al. (eds.) *Applied and numerical partial differential equations. Computational methods in applied sciences*, vol. 15, pp. 25–39. Springer (2010)
- Cheney, M., Isaacson, D., Newell, J.: Electrical impedance tomography. *SIAM Rev.* **41**(1), 85–101 (1999)
- Dambrine, M., Dapogny, C., Harbrecht, H.: Shape optimization for quadratic functionals and states with random right-hand sides. *SIAM J. Control Optim.* **53**, 3081–3103 (2015)
- Dambrine, M., Harbrecht, H., Puig, B.: Incorporating knowledge on the measurement noise in electrical impedance tomography. *ESAIM: Control Optim. Calc. Var.* **25**, 84 (2019)
- Delfour, M., Zolésio, J.P.: *Shapes and geometries: Metrics, analysis, differential calculus, and optimization. Advanced design control*, vol. 22, 2nd edn. SIAM, Philadelphia, USA (2001)
- Doğan, G., Morin, P., Nochetto, R.H., Verani, M.: Discrete gradient flows for shape optimization and applications. *Comput. Meth. Appl. Mech. Eng.* **196**(37–40), 3898–3914 (2007). <https://doi.org/10.1016/j.cma.2006.10.046>
- Droske, M., Rumpf, M.: Multi scale joint segmentation and registration of image morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(12), 2181–2194 (2007)
- Etiling, T., Herzog, R., Loayza, E., Wachsmuth, G.: First and second order shape optimization based on restricted mesh deformations. *SIAM J. Scient. Comput.* **42**(2), A1200–A1225 (2020). <https://doi.org/10.1137/19m1241465>
- Evans, L.: *Partial differential equations. graduate studies in mathematics*, vol. 19. American Mathematical Society, Providence, USA (1998)
- Feppon, F., Allaire, G., Bordeu, F., Cortial, J., Dapogny, C.: Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework. *SeMA J. Boletín de la Sociedad Española de Matemática Aplicada.* **76**(3), 413–458 (2019). <https://doi.org/10.1007/s40324-018-00185-4>
- Fuchs, M., Jüttler, B., Scherzer, O., Yang, H.: Shape metrics based on elastic deformations. *J. Math. Imaging Vis.* **35**(1), 86–102 (2009)
- Gangl, P., Laurain, A., Meftahi, H., Sturm, K.: Shape optimization of an electric motor subject to nonlinear magnetostatics. *SIAM J. Sci. Comput.* **37**(6), B1002–B1025 (2015)
- Geiersbach, C.: *Stochastic approximation for PDE-constrained optimization under uncertainty. Ph.D. thesis, University of Vienna* (2020)
- Geiersbach, C., Pflug, G.C.: Projected stochastic gradients for convex constrained problems in Hilbert spaces. *SIAM J. Optim.* **29**(3), 2079–2099 (2019)
- Geiersbach, C., Scarinci, T.: Stochastic proximal gradient methods for nonconvex problems in Hilbert spaces. *Comput. Optim. Appl.* **3**(78), 705–740 (2021). <https://doi.org/10.1007/s10589-020-00259-y>
- Geiersbach, C., Wollner, W.: A stochastic gradient method with mesh refinement for pde-constrained optimization under uncertainty. *SIAM J. Sci. Comput.* **42**(5), A2750–A2772 (2020)
- Geiersbach, C., Loayza-Romero, E., Welker, K.: Stochastic approximation for optimization in shape spaces. *SIAM J. Optim.* **31**(1), 348–376 (2021)
- Haber, E., Chung, M., Herrmann, F.: An effective method for parameter estimation with PDE constraints with multiple right-hand sides. *SIAM J. Optim.* **22**(3), 739–757 (2012)
- Hardesty, S., Kouri, D., Lindsay, P., Ridzal, D., Stevens, B., Viertel, R.: *Shape optimization for control and isolation of structural vibrations in aerospace and defense applications. techreport, Office of Scientific and Technical Information (OSTI)* (2020). <https://doi.org/10.2172/1669731>

- Haubner, J., Siebenborn, M., Ulbrich, M.: A continuous perspective on shape optimization via domain transformations. *SIAM J. Scient. Comput.* **43**(3), A1997–A2018 (2020). <https://doi.org/10.1137/20m1332050>
- Herzog, R., Loayza-Romero, E.: A manifold of planar triangular meshes with complete riemannian metric (2020). ArXiv:2012.05624
- Hiptmair, R., Paganini, A.: Shape optimization by pursuing diffeomorphisms. *Comput. Methods Appl. Math.* **15**(3), 291–305 (2015)
- Hiptmair, R., Jerez-Hanckes, C., Mao, S.: Extension by zero in discrete trace spaces: inverse estimates. *Math. Comput.* **84**(296), 2589–2615 (2015)
- Hiptmair, R., Paganini, A., Sargheini, S.: Comparison of approximate shape gradients. *BIT. Num. Math.* **55**(2), 459–485 (2015). <https://doi.org/10.1007/s10543-014-0515-z>
- Hiptmair, R., Scarabosio, L., Schillings, C., Schwab, C.: Large deformation shape uncertainty quantification in acoustic scattering. *Adv. Comput. Math.* **44**(5), 1475–1518 (2018)
- Ito, K., Kunisch, K.: Lagrange Multiplier Approach to Variational Problems and Applications. *Advanced Design Control*, vol. 15. SIAM, Philadelphia, USA (2008)
- Ito, K., Kunisch, K., Peichl, G.: Variational approach to shape derivatives. *ESAIM Control Optim. Calc. Var.* **14**(3), 517–539 (2008)
- Kendall, D.: Shape manifolds, procrustean metrics, and complex projective spaces. *Bull. Lond. Math. Soc.* **16**(2), 81–121 (1984)
- Kriegl, A., Michor, P.: The convenient setting of global analysis. In *Mathematical surveys and monographs*, vol. 53. American Mathematical Society, Providence, USA (1997). <https://books.google.de/books?id=-XxBwAAQBAJ>
- Kwon, O., Woo, E.J., Yoon, J., Seo, J.: Magnetic resonance electrical impedance tomography (MREIT): simulation study of J -substitution algorithm. *IEEE Trans. Biomed. Eng.* **49**(2), 160–167 (2002)
- Laurain, A., Sturm, K.: Domain expression of the shape derivative and application to electrical impedance tomography. Technical Report No. 1863, Weierstraß-Institut für angewandte Analysis und Stochastik, Berlin (2013)
- Laurain, A., Sturm, K.: Distributed shape derivative via averaged adjoint method and applications. *ESAIM: Math. Model. Numer. Anal.* **50**(4), 1241–1267 (2016)
- Ling, H., Jacobs, D.: Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(2), 286–299 (2007)
- Liu, D., Litvinenko, A., Schillings, C., Schulz, V.: Quantification of airfoil geometry-induced aerodynamic uncertainties—comparison of Approaches (2017)
- Lord, G., Powell, C., Shardlow, T.: An introduction to computational stochastic PDEs. Cambridge University Press, Cambridge, UK (2014)
- Luft, D., Schulz, V.: Pre-shape calculus and its application to mesh quality optimization. *Control. Cybern.* **50**(3), 263–301 (2021a) <https://doi.org/10.2478/candc-2021--0019>. ArXiv:2012.09124 ArXiv:2012.09124
- Luft, D., Schulz, V.: Simultaneous shape and mesh quality optimization using pre-shape calculus. *Control. Cybern.* **50**(4), 473–520 (2021b) <https://doi.org/10.2478/candc-2021--0028>. ArXiv:2103.15109
- Martin, M., Krumscheid, S., Nobile, F.: Analysis of stochastic gradient methods for PDE-constrained optimal control problems with uncertain parameters. Tech. rep., École Polytechnique MATHICSE Institute of Mathematics (2018)
- Martin, M., Nobile, F., Tsilifis, P.: A multilevel stochastic gradient method for pde-constrained optimal control problems with uncertain parameters. arXiv preprint arXiv:1912.11900 (2019)
- Martínez-Frutos, J., Herrero-Pérez, D., Kessler, M., Periago, F.: Robust shape optimization of continuous structures via the level set method. *Comput. Methods Appl. Mech. Eng.* **305**, 271–291 (2016)

- Michor, P., Mumford, D.: Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. *Doc. Math.* **10**, 217–245 (2005)
- Michor, P., Mumford, D.: Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc. (JEMS)* **8**(1), 1–48 (2006)
- Michor, P., Mumford, D.: An overview of the Riemannian metrics on spaces of curves using the Hamiltonian approach. *Appl. Comput. Harmon. Anal.* **23**(1), 74–113 (2007)
- Morin, P., Nochetto, R.H., Pauletti, M.S., Verani, M.: Adaptive finite element method for shape optimization. *ESAIM Control Optim. Calc. Var.* **18**(4), 1122–1149 (2012). <https://doi.org/10.1051/cocv/2011192>
- Novruzı, A., Pierre, M.: Structure of shape derivatives. *J. Evol. Equ.* **2**(3), 365–382 (2002)
- O’neill, B.: *Semi-Riemannian geometry with applications to relativity*. Academic Press, London, UK (1983)
- Onyshkevych, S., Siebenborn, M.: Mesh quality preserving shape optimization using nonlinear extension operators. *J Optim. Theory. Appl.* **189**(1), 291–316 (2021). <https://doi.org/10.1007/s10957-021-01837-8>
- Paganini, A.: Approximative shape gradients for interface problems. In: Pratelli, A., Leugering, G. (eds.) *New trends in shape optimization*. International series of numerical mathematics, vol. 166, pp. 217–227. Springer (2015)
- Quarteroni, A., Valli, A.: *Domain decomposition methods for partial differential equations*. Oxford University Press, Oxford, UK (1999)
- Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**(3), 400–407 (1951)
- Schulz, V.: A Riemannian view on shape optimization. *Found. Comput. Math.* **14**(3), 483–501 (2014)
- Schulz, V., Siebenborn, M.: Computational comparison of surface metrics for PDE constrained shape optimization. *Comput. Methods Appl. Math.* **16**(3), 485–496 (2016)
- Schulz, V., Welker, K.: On optimization transfer operators in shape spaces. In: *Shape optimization, homogenization and optimal Control*, pp. 259–275. Springer (2018)
- Schulz, V., Siebenborn, M., Welker, K.: Structured inverse modeling in parabolic diffusion problems. *SIAM J. Control Optim.* **53**(6), 3319–3338 (2015)
- Schulz, V., Siebenborn, M., Welker, K.: Efficient PDE constrained shape optimization based on Steklov-Poincaré type metrics. *SIAM J. Optim.* **26**(4), 2800–2819 (2016)
- Schwab, C., Gittelsohn, C.: Sparse tensor discretizations of high-dimensional parametric and stochastic pdes. *Acta Numer.* **20**, 291–467 (2011)
- Shapiro, A., Wardi, Y.: Convergence analysis of gradient descent stochastic algorithms. *J. Optim. Theory Appl.* **91**(2), 439–454 (1996)
- Shapiro, A., Dentcheva, D., Ruszczyński, A.: *Lectures on stochastic programming: modeling and theory*. SIAM, Philadelphia, USA (2009)
- Siebenborn, M., Vogel, A.: A shape optimization algorithm for cellular composites. *PINT Computing and Visualization in Science* (2021). [ArXiv:1904.03860](https://arxiv.org/abs/1904.03860)
- Siebenborn, M., Welker, K.: Algorithmic aspects of multigrid methods for optimization in shape spaces. *SIAM J. Sci. Comput.* **39**(6), B1156–B1177 (2017)
- Sokolowski, J., Zolésio, J.: Introduction to shape optimization. In: *Computational mathematics*, vol. 16. Springer (1992)
- Sturm, K.: Lagrange method in shape optimization for non-linear partial differential equations: a material derivative free approach. Technical Report No. 1817, Weierstraß-Institut für angewandte Analysis und Stochastik, Berlin (2013)
- Sturm, K.: Shape optimization with nonsmooth cost functions: from theory to numerics. *SIAM J. Control Optim.* **54**(6), 3319–3346 (2016). <https://doi.org/10.1137/16M1069882>

- Wardi, Y.: Stochastic algorithms with armijo stepsizes for minimization of functions. *J. Optim. Theory Appl.* **64**(2), 399–417 (1990)
- Welker, K.: Efficient PDE constrained shape optimization in shape spaces. Ph.D. thesis, Universität Trier (2016)
- Welker, K.: Suitable spaces for shape optimization. *Appl. Math. Optim.* (2021). <https://doi.org/10.1007/s00245-021-09788-2>
- Wirth, B., Rumpf, M.: A nonlinear elastic shape averaging approach. *SIAM J. Imag. Sci.* **2**(3), 800–833 (2009)
- Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A continuum mechanical approach to geodesics in shape space. *Int. J. Comput. Vis.* **93**(3), 293–318 (2011)
- Zolésio, J.P.: Control of moving domains, shape stabilization and variational tube formulations. *Int. Ser. Numer. Math.* **155**, 329–382 (2007)



Iterative Methods for Computing Eigenvectors of Nonlinear Operators

46

Guy Gilboa

Contents

Introduction and Preliminaries	1632
One-Homogeneous Functionals	1632
Eigenvectors of Nonlinear Operators	1634
Nossek-Gilboa (NG)	1636
NG Flow Properties	1637
NG Iteration Algorithm Properties	1639
Aujol-Gilboa-Papadakis (AGP)	1639
AGP Flow Properties	1640
AGP Iteration Algorithm Properties	1641
Feld-Aujol-Gilboa-Papadakis (FAGP)	1641
Cohen-Gilboa (CG)	1644
Bungert-Hait-Papadakis-Gilboa (BHPG)	1647
Evaluation and Examples	1649
Global and Local Measures	1649
Numerical Examples	1651
Conclusion, Discussion and Open Problems	1652
References	1656

Abstract

In this chapter we are examining several iterative methods for solving nonlinear eigenvalue problems. These arise in variational image processing, graph partition and classification, nonlinear physics, and more. The canonical eigenproblem we solve is $T(u) = \lambda u$, where $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is some bounded nonlinear operator. Other variations of eigenvalue problems are also discussed. We present a progression of five algorithms, coauthored in recent years by the author and

G. Gilboa (✉)
Technion – IIT, Haifa, Israel
e-mail: guy.gilboa@ee.technion.ac.il

colleagues. Each algorithm attempts to solve a unique problem or to improve the theoretical foundations. The algorithms can be understood as nonlinear PDEs which converge to an eigenfunction in the continuous time domain. This allows a unique view and understanding of the discrete iterative process. Finally, it is shown how to evaluate numerically the results, along with some examples and insights related to priors of nonlinear denoisers, both classical algorithms and ones based on deep networks.

Keywords

Nonlinear spectral analysis · Nonlinear eigenvectors · Spectral total variation · One-homogeneous functionals.

Introduction and Preliminaries

In this section, we outline some basic notations and properties which will be used throughout this chapter. A main type of functionals we are discussing are one-homogeneous functionals, used frequently as regularizers in image processing and learning.

One-Homogeneous Functionals

We consider an absolutely one-homogeneous functional J that takes as input a function $u : x \in \Omega \rightarrow \mathbb{R}$ defined on a domain $\Omega \subset \mathbb{R}^2$. Ω can either be a discrete domain of size $|\Omega| = N$ or an open convex bounded set with Lipschitz boundary. u are elements of some Hilbert space X (e.g., X can be $L^2(\Omega)$) embedded with some inner product $\langle \cdot, \cdot \rangle$. $J : X \rightarrow \mathbb{R} \cup \{+\infty\}$ is assumed to be proper, convex, and lower semicontinuous (lsc). Absolutely one-homogeneous functionals satisfy

$$J(cu) = |c|J(u), \quad \forall c \in \mathbb{R}, \forall u \in X. \quad (1)$$

The functional J in finite dimensions can be, for instance, of the general form:

$$J(u) = \sum_{i=1}^N \left(\sum_{j=1}^N w_{ij} |u_i - u_j|^q \right)^{1/q}, \quad (2)$$

for $q \geq 1$, with $w_{ij} \geq 0$ (usually symmetric weights are assumed $w_{ij} = w_{ji}$). This formulation can be understood as a typical one-homogeneous functional on weighted graphs. In this case u_i is the value of the function u at node i on the graph, and w_{ij} is the weight between node i and node j . As grids of any dimension can be realized by specific graph structures, this formulation applies to standard grids as well. Thus (2), with appropriate weights, can be the spatial discrete version of

anisotropic total variation (TV) ($q = 1$), isotropic TV ($q = 2$), and anisotropic or isotropic nonlocal TV.

We recall the subgradient definition for general convex functionals:

$$p \in \partial J(u) \Leftrightarrow J(v) - J(u) \geq \langle p, v - u \rangle, \forall v.$$

We also note the relation to the convex conjugate J^* :

$$J(u) = \sup_p \langle u, p \rangle - J^*(p).$$

Below we state some properties of one-homogeneous functionals.

Property. A function J defined in (2) admits:

- (a) If $p \in \partial J(u)$, then $J(u) = \langle p, u \rangle$,
- (b) If $p \in \partial J(u)$, then $J(v) \geq \langle p, v \rangle, \forall v$.

Notice in particular that from (b), we get that $\partial J(u) \subset \partial J(0) \forall u \in X$.

Property. The convex conjugate J^* of a one-homogeneous functional is the characteristic function of the convex set $\{\partial J(0)\}$. Moreover, when Ω is included in a finite-dimensional space, we have (Burger et al. 2016):

$$\exists C > 0 \text{ s.t. } \|p\|_2 \leq C, \forall p \in \partial J(0). \quad (3)$$

From the equivalence of norms, we have that if u is of zero mean, there exists a constant $\kappa > 0$ for which

$$\|u\|_2 \leq \kappa J(u), \forall u \text{ such that } \langle u, \mathbf{1} \rangle = 0. \quad (4)$$

The null-space of the functional is defined by

$$\mathcal{N}(J) = \{u \in X \mid J(u) = 0\}. \quad (5)$$

The properties below are shown in Burger et al. (2016).

Property. An absolutely one-homogeneous functional J is a seminorm, and its null-space is a linear subspace.

Property. If a unit constant function $u = \mathbf{1}$ is in $\mathcal{N}(J)$, then any subgradient p admits:

$$\langle p, \mathbf{1} \rangle = 0.$$

We use ℓ_2 and ℓ_1 norms of u defined as $\|u\|_2 = \sqrt{\langle u, u \rangle}$ and $\|u\|_1 = \langle u, \text{sign}(u) \rangle$.

Eigenvectors of Nonlinear Operators

We give here a brief introduction to the broad topic of eigenvectors of nonlinear operators. More details are provided in relation to the variational setting. We would like to extend the linear eigenvalue problem:

$$Lu = \lambda u,$$

given a matrix L , to a generalized problem, given a bounded nonlinear operator $T : X \rightarrow X$. Replacing L by T , we get the nonlinear eigenvalue problem associated with T :

$$T(u) = \lambda u, \tag{6}$$

where $\lambda \in \mathbb{R}$ is the associated eigenvalue. In the variational context, given a convex functional J , the eigenvalue problem induced by J is

$$p = \lambda u, \quad p \in \partial J(u). \tag{7}$$

As an example, for the Dirichlet energy $J = \frac{1}{2} \|\nabla u\|^2$, the associated eigenvalue problem is a linear one:

$$-\Delta u = \lambda u,$$

where Δ denotes the Laplacian. For appropriate boundary conditions, sines and cosines are solutions to this problem, which are the basis elements of the Fourier transform. For one-homogeneous regularizing functionals, such as total variation, one obtains different (sharp) eigenfunctions, which can serve for representing signals based on nonlinear spectral transforms, as shown in Gilboa (2013, 2014, 2018), Burger et al. (2016), and Bungert et al. (2019a). We would not elaborate on this direction, which is beyond the scope of this chapter.

For absolutely one-homogeneous functionals, the eigenvalues are nonnegative, since $J(u) = \langle \lambda u, u \rangle = \lambda \|u\|_2^2$ and $\lambda = \frac{J(u)}{\|u\|_2^2} \geq 0$. An interesting insight on the eigenvalue λ shown in Aujol et al. (2018) can be gained by the following proposition. We define $K = \{\partial J(0)\}$ to be the set of possible subgradients for any u . Indeed if $p \in \partial J(u)$, then $p \in \partial J(0)$. We first note that an eigenfunction that admits $\lambda u \in \partial J(u)$ has zero mean from Property above. Next, we have the following result.

Proposition. *For any nonconstant eigenfunction u , we have $\forall \mu \geq \lambda$:*

$$\lambda u = \text{Proj}_K(\mu u),$$

where Proj_K is the orthogonal projection onto $K = \{\partial J(0)\}$.

Eigenfunctions in the form of (7) have analytic solutions, when used as initial conditions in gradient flows. Let a gradient flow be defined by

$$u_t = -p \quad u|_{t=0} = f, \quad p \in \partial J(u), \tag{8}$$

where u_t is the first-time derivative of $u(t; x)$. As shown in Burger et al. (2016), when the flow is initialized with an eigenfunction (i.e., $\lambda f \in \partial J(f)$), the following solution is obtained:

$$u(t; x) = (1 - \lambda t)^+ f(x), \tag{9}$$

where $(q)^+ = q$ for $q > 0$ and 0 otherwise. This means that the shape $f(x)$ is spatially preserved and changes only by contrast reduction throughout time. An analytic solution (see Benning and Burger 2013, and Burger et al. 2016) can be shown for the proximal problem as well, that is, a minimization with the square 2 norm:

$$\min_u J(u) + \frac{\alpha}{2} \|f - u\|_2^2. \tag{10}$$

In this case, when f is an eigenfunction and $\alpha \in \mathbb{R}^+$ ($\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$) is fixed, the problem has the following solution:

$$u(x) = \left(1 - \frac{\lambda}{\alpha}\right)^+ f(x). \tag{11}$$

In this case also, $u(x)$ preserves the spatial shape of $f(x)$ (as long as $\alpha > \lambda$). This was already observed by Meyer in (2001) for the case of a disk with J the TV functional. Earlier research on nonlinear eigenfunctions induced by TV, which are set indicator functions, has been referred as *calibrable sets*. First aspects of this line of research can be found in the work of Bellettini et al. (2002). They introduced a family of convex bounded sets C with finite perimeter in \mathbb{R}^2 that preserve their boundary throughout the TV flow (gradient flow (8) where J is TV). It is shown that the indicator function of a set C , $\mathbf{1}_C$, with perimeter $P(C)$ which admits

$$\text{ess sup}_{p \in \partial C} \kappa(p) \leq \frac{P(C)}{|C|} \tag{12}$$

is an eigenfunction, in the sense of (7), where $u = \lambda_C \mathbf{1}_C$ and

$$\lambda_C = \frac{P(C)}{|C|}. \tag{13}$$

A further generalization of (6), referred to as the *double-nonlinear* eigenvalue problem, is formulated by introducing another bounded nonlinear operator Q , to have

$$T(u) = \lambda Q(u). \quad (14)$$

Here $Q(u)$ may be high-order polynomials or trigonometric functions. In physics, a variant of (14) is quite common, where T is a linear operator (mostly the Laplacian). For example, the one-dimensional Schrodinger equation

$$-u_{xx} = \lambda(u^3 - u).$$

We will address here ways also of how to solve such problems. In the variational context, T and Q are two subgradient elements of different convex functionals, J and H ; thus (14) is rewritten as

$$p = \lambda q, \quad p \in \partial J(u), \quad q \in \partial H(u). \quad (15)$$

This type of problem appears in the relaxation of the Cheeger cut problem, where J is TV and H is ℓ^1 ; see Hein and Bühler (2010), Szlam and Bresson (2010), and Feld et al. (2019). There are several additional algorithms which attempt to compute nonlinear eigenfunctions in some specific settings. In Bozorgnia (2016, 2019), algorithms for computing the smallest eigenvalue and eigenfunction of the p -Laplacian are proposed, along with convergence proofs. As part of analyzing variational networks, (Efland et al. 2020) analyze the learned regularizers by computing their eigenfunctions. This is performed by minimizing a generalized Rayleigh quotient using accelerated gradient descent. In the process of nonlinear spectral decomposition based on gradient descent (Gilboa 2014; Burger et al. 2016), near extinction time only a single eigenfunction “survives.” This idea is formalized in Bungert et al. (2019b) where eigenfunctions are computed by taking the limit at extinction time of a gradient flow. Gautier et al. (2019, 2020) have used power iterations to solve several nonlinear eigenpair problems. Existence and uniqueness results were obtained based on Perron-Frobenius theory.

We will now present in detail five algorithms, coauthored by the author and colleagues, to solve various types of nonlinear eigenvalue problems. Some of the iterative algorithms can be understood as a discretization in time of a continuous nonlinear flow.

Nossek-Gilboa (NG)

This simple algorithm, presented first in Nossek and Gilboa (2018), was the first of a series of algorithms, which stem from nonlinear flows. These flows reach a steady state only at eigenfunctions. Different initial conditions yield different steady states. The goal for the (NG) algorithm is to provide a solution to the nonlinear eigenvalue problem (7), where J is an absolutely one-homogeneous functional, admitting (1). We assume a constant unit vector is in its null-space (Property). The proposed nonlinear flow is

$$u_t = \frac{u}{\|u\|_2} - \frac{p}{\|p\|_2}, \quad p \in \partial J(u), \quad (16)$$

where $u(0) = u_0 \in X$ is an initial condition, with $\langle u_0, \mathbf{1} \rangle = 0$. The associated iterative algorithm for solving (7) is detailed in Algorithm 1.

Algorithm 1 (NG). Compute a nonlinear eigenfunction $\lambda u \in \partial J(u)$, associated with an absolutely one-homogeneous functional J

Data: u_0 with $\langle u_0, \mathbf{1} \rangle = 0$, $\Delta t \in (0, \|u_0\|_2)$, ϵ .

Result: Eigenfunction and eigenvalue, $\{u^k, \lambda^k\}$, where $\lambda^k = J(u^k)/\|u^k\|_2^2$.

Initialization: $k \leftarrow 0$, $u^k \leftarrow u_0$.

repeat

$$u^{k+1} = u^k + \Delta t \left(\frac{u^{k+1}}{\|u^k\|_2} - \frac{p^{k+1}}{\|p^k\|_2} \right), \quad (17)$$

until $\|u^{k+1} - u^k\|_2 < \epsilon$;

Equation (17) is computed by solving the following convex optimization problem:

$$u^{k+1} = \arg \min_v \left\{ J(v) + \frac{\|p^k\|_2}{2\Delta t} \left(1 - \frac{\Delta t}{\|u^k\|_2} \right) \left\| \frac{u^k}{1 - \frac{\Delta t}{\|u^k\|_2}} - v \right\|_2^2 \right\}. \quad (18)$$

NG Flow Properties

There are several desired properties of this flow. Although it does not emerge as a gradient flow of a certain energy functional, the solution becomes smoother with time (in terms of the regularizing functional J). On the other hand, the ℓ^2 norm of the solution is increasing. The main properties are summarized in the following theorem. In this case, the proof is presented, and is relatively simple to follow (it is based on Nossek and Gilboa 2018 and Aujol et al. 2018). This allows us to get the intuition of how such flows behave. In subsequent parts, proofs are omitted, and we refer the reader to the relevant papers for details, to avoid a lengthy presentation.

Theorem 1. Assume that there exists a solution u in $W^{1,2}((0, T); X)$, $T > 0$, of the flow (16). Then the following properties hold:

$$\frac{d}{dt} \frac{1}{2} \|u(t)\|_2^2 \geq 0, \quad (19)$$

moreover, we have $\langle u(t), \mathbf{1} \rangle = 0$, and in addition,

$$\frac{d}{dt} J(u(t)) \leq 0 \text{ for almost every } t. \quad (20)$$

We conclude that $t \mapsto J(u(t))$ is nonincreasing for all $t \geq 0$.

Proof. Recalling that $\langle p, u \rangle \leq \|p\|_2 \|u\|_2$, this flow ensures that

$$\frac{d}{dt} \frac{1}{2} \|u(t)\|_2^2 = \langle u, u_t \rangle = \left\langle u, \frac{u}{\|u\|_2} - \frac{p}{\|p\|_2} \right\rangle = \|u\|_2 - \frac{\langle u, p \rangle}{\|p\|_2} \geq 0$$

We can also remark that

$$\frac{d}{dt} \frac{1}{2} \|u(t)\|_2^2 \leq \|u(t)\|_2$$

so that

$$\|u(t)\|_2 \leq \|u_0\|_2 + 2t.$$

Additionally, if u_0 is of zero mean, Property ensures that $u(t)$ is of zero mean, for all $t > 0$. To show (20) we make use of Lemma 3.3 page 73 in Brezis (1973) (see also Lemma 4.1 in Vassilis et al. 2018). It allows us to use the “chain rule for differentiation.” Let us first recall this lemma.

Lemma 1 (Brezis ’73). *Let $T > 0$ and F be a convex, lower semicontinuous, proper function and $v \in W^{1,2}((0, T); X)$. Let also $h \in L^2((0, T); X)$, such that $h \in \partial F(v(t))$ a.e. in $(0, T)$. Then the function $F \circ v : [0, T] \rightarrow \mathbb{R}$ is absolutely continuous in $[0, T]$ with*

$$\frac{d}{dt} (F(v(t))) = \langle z, v_t \rangle, \quad \forall z \in \partial F(v(t)) \text{ a.e. in } (0, T).$$

From Lemma 1, if u is in $W^{1,2}((0, T); X)$, we get that $J(u(t))$ is absolutely continuous in $[0, T]$ with

$$\frac{d}{dt} J(u(t)) = \langle p, u_t \rangle = \left\langle p, \frac{u}{\|u\|_2} - \frac{p}{\|p\|_2} \right\rangle = \frac{\langle u, p \rangle}{\|u\|_2} - \|p\|_2 \leq 0.$$

This inequality holds for almost every t , and since $t \mapsto J(u(t))$ is an absolutely continuous function, we deduce that it is a nonincreasing function.

The flow (16) converges iff $u_t = 0$ so that

$$p = \frac{\|p\|_2}{\|u\|_2} u \in \partial J(u) \Rightarrow p = \frac{J(u)}{\|u\|_2^2} u$$

and u is an eigenfunction of J with eigenvalue $\lambda = \frac{J(u)}{\|u\|_2^2}$.

NG Iteration Algorithm Properties

The iterations in Algorithm 1 can be viewed as a semi-implicit scheme of the flow (16). The properties of the discrete flow are similar in nature to those of the continuous flow (but not precisely the same). They are summarized in the following theorem (details are given in Nossek and Gilboa 2018).

Theorem 2. *The solution u^k of the discrete flow (17) of Algorithm 1 has the following properties:*

- (i) $\langle u^k, \mathbf{1} \rangle = 0$.
- (ii) $\|p^{k+1}\|_2 \leq \|p^k\|_2$.
- (iii) $\|u^{k+1}\|_2 \geq \|u^k\|_2$.
- (iv) $\frac{J(u^{k+1})}{\|u^{k+1}\|_2} \leq \frac{J(u^k)}{\|u^k\|_2}$.
- (v) *A sufficient and necessary condition for steady-state $u^{k+1} = u^k$ holds if u^k is an eigenfunction, admitting (7).*

Aujol-Gilboa-Papadakis (AGP)

In Aujol et al. (2018), the authors proposed a generalized flow for solving (7), which is more stable than (NG) and can be better analyzed theoretically. The general flow, for $\alpha \in [0; 1]$, is

$$u_t = \left(\frac{J(u)}{\|u\|_2^2} \right)^\alpha u - \left(\frac{J(u)}{\|p\|_2^2} \right)^{1-\alpha} p, \quad p \in \partial J(u), \tag{21}$$

with $u(0) = u_0 \in X, \langle u_0, \mathbf{1} \rangle = 0$. Notice that for $\alpha = 1/2$, we retrieve the (NG) flow, (16), up to a normalization with $J^{1/2}(u)$. For the case $\alpha = 1$, the flow becomes

$$u_t = \left(\frac{J(u)}{\|u\|_2^2} \right) u - p, \quad p \in \partial J(u). \tag{22}$$

In this case, there is no term with $\|p\|_2$ in the denominator, and the analysis simplifies. Uniqueness of the flow and convergence of the iterative algorithm are established.

For the case $\alpha = 1$, we get that the ℓ^2 norm is fixed in time. This allows us to have a unit norm throughout the evolution. In the discrete iterations, however, an additional normalization step is required to maintain this property. Given any input f , to obtain a valid initial condition u_0 , we first subtract the mean and then

normalize by the ℓ^2 norm. The associated iterative algorithm, $\alpha = 1$, for solving (7) is detailed in Algorithm 2.

Algorithm 2 (AGP). Compute a nonlinear eigenfunction $\lambda u \in \partial J(u)$, associated with an absolutely one-homogeneous functional J

Data: u_0 with $\langle u_0, \mathbf{1} \rangle = 0, \|u_0\|_2 = 1, \Delta t \in (0, \|u_0\|_2^2/J(u_0)), \epsilon.$

Result: Eigenfunction and eigenvalue, $\{u^k, \lambda^k\}$, where $\lambda^k = J(u^k)/\|u^k\|_2^2.$

Initialization: $k \leftarrow 0, u^k \leftarrow u_0.$

repeat

$$\begin{aligned}
 u^{k+1/2} &= u^k + \Delta t \left(\frac{J(u^k)u^{k+1/2}}{\|u^k\|_2^2} - p^{k+1/2} \right), \\
 u^{k+1} &= \frac{u^{k+1/2}}{\|u^{k+1/2}\|_2}.
 \end{aligned}
 \tag{23}$$

until $\|u^{k+1} - u^k\|_2 < \epsilon;$

The term $u^{k+1/2}$ in Eq. (23) is computed by solving

$$u^{k+1/2} = \arg \min_v \left\{ J(v) + \frac{1}{2\Delta t} \|v - u^k\|_2^2 - \frac{J(u^k)}{2\|u^k\|_2^2} \|v\|_2^2 \right\}.
 \tag{24}$$

There is a unique minimizer v for any time step Δt which is in the range specified above.

AGP Flow Properties

Theorem 3. For u_0 of zero mean and $\forall \alpha \in [0; 1]$, if u is in $W^{1,2}((0, T); X)$, then the trajectory $u(t)$ of the flow (21) satisfies the following properties:

- (i) $\langle u(t), \mathbf{1} \rangle = 0.$
- (ii) $\frac{d}{dt} J(u(t)) \leq 0$ for almost every $t.$ Moreover, $t \mapsto J(u(t))$ is nonincreasing. If $\alpha = 0,$ we have for almost every t that $\frac{d}{dt} J(u(t)) = 0$ and $t \mapsto J(u(t))$ is constant.
- (iii) $\frac{d}{dt} \|u(t)\|_2 \geq 0$ and $\frac{d}{dt} \|u(t)\|_2 = 0$ for $\alpha = 1.$
- (iv) If the flow converges to $u^*,$ we have $p^* = J^{2\alpha-1}(u^*) \frac{\|p^*\|_2^{2(1-\alpha)}}{\|u^*\|_2^{2\alpha}} u^* \in \partial J(u^*)$ so that u^* is an eigenfunction.

Uniqueness. For the case $\alpha = 1,$ one can establish uniqueness of the flow (22), under mild conditions.

Theorem 4. *Let u and v be two solutions of (22) in $W^{1,2}((0, T); X)$ with respective initial condition u_0 and v_0 , such that $J(u_0) < +\infty$ and $J(v_0) < +\infty$, with $\|u_0\|_2 = \|v_0\|_2 = 1$. Then we have:*

$$\frac{d}{dt} \left(\frac{1}{2} \|u - v\|_2^2 \right) \leq \frac{J(u) + J(v)}{2} \|u - v\|_2^2. \tag{25}$$

By the fact that $J(u)$ is decreasing and using Gronwall lemma, we obtain

$$\|u - v\|_2^2 \leq \|u_0 - v_0\|_2^2 \exp \left((J(u_0) + J(v_0))(t - t_0) \right). \tag{26}$$

AGP Iteration Algorithm Properties

The iterations in Algorithm 2 can be viewed as a semi-implicit scheme of the flow (22). The algorithm’s properties are detailed below.

Theorem 5. *Let u_0 in X , and the sequence u_k defined by (23). Then the sequences $J(u_k)$ and $\|p_k\|_2$ are nonincreasing, $\|u_k\|_2 = \|u_0\|_2$ for all k , and $u_{k+1} - u_k \rightarrow 0$.*

Convergence. Finally, it is shown that Algorithm 2 converges to an eigenfunction.

Theorem 6. *Let u_0 be in X , and the sequence u_k be defined by (23). There exist some u and p in X such that, up to a subsequence, u_k converges to u in X and p_k converges to p in X , with $p \in \partial J(u)$, and $J(u_k)$ converges to $J(u)$. Moreover, u is a nonlinear eigenfunction, in the sense of (7).*

Feld-Aujol-Gilboa-Papadakis (FAGP)

In Feld et al. (2019), the aim is to solve the problem (15) for the case when J and H are both absolutely one-homogeneous functionals. Let us consider the generalized nonlinear Rayleigh quotient:

$$R(u) := \frac{J(u)}{H(u)}. \tag{27}$$

In an analogue to the linear case, eigenfunctions in the sense of (15) are critical points of (27). In segmentation, classification, and clustering, often we seek eigenfunctions with the least (strictly positive) eigenvalue. Thus, excluding the null-space of J and H , we seek to minimize the Rayleigh quotient (27). A classical way to reach a local minimizer of $R(u)$ is by using a gradient-descent flow:

$$u_t = -\nabla R(u).$$

Taking the variational derivative of $R(u)$, with $q \in \partial H(u)$, $p \in \partial J(u)$, the gradient descent flow is

$$u_t = \frac{J(u)q - H(u)p}{H^2(u)}. \tag{28}$$

The flow can also be written as

$$u_t = \frac{R(u)q - p}{H(u)}.$$

This flow is hard to analyze theoretically, mainly due to the division by $H(u)$. Therefore, Feld et al. (2019) proposed the following flow to minimize $R(u)$:

$$u_t = R(u)q - p. \tag{29}$$

This is essentially a gradient-descent type flow, without the division by $H(u)$, which can be interpreted as a dynamic rescaling of the time parameter. The flow reduces monotonically the quotient $R(u)$, and the steady state admits the nonlinear eigenvalue problem (15).

A second flow is proposed that minimizes the log of the Rayleigh quotient:

$$u_t = -\nabla(\log R(u)),$$

which can be written as

$$u_t = \frac{q}{H(u)} - \frac{p}{J(u)}. \tag{30}$$

This is motivated by a widely used practice of using the log of a function involving multiplicative expressions. It is commonly employed in statistics and machine learning algorithms, such as maximum likelihood estimation and policy learning. The flow is essentially a time rescaling of (29) by $1/J(u)$. We note that it is not in the form of Brezis Lemma 1 and therefore is harder to analyze. We will not focus on this flow here. It is worth mentioning, however, that in the context of the Cheeger cut problem, we found out that numerically it is very stable and highly resilient to the choice of the discrete time step. Thus a large time step can be chosen, which speeds up numerical convergence (see details in Feld et al. 2019).

The algorithm is based on the following semi-explicit scheme of the flow:

$$\begin{cases} (u^{k+1/2} - u^k)/\Delta t = R(u^k)q_k - p_{k+1/2}, & q_k \in \partial H(u^k), p_{k+1/2} \in \partial J(u^{k+1/2}) \\ u^{k+1} = u^{k+1/2}/\|u^{k+1/2}\|_2. \end{cases} \tag{31}$$

This scheme is associated with the minimization of a convex functional:

$$u^{k+1/2} = \arg \min_{u \in X} F(u) := \frac{1}{2\Delta t} \|u - u^k\|_2^2 - R(u^k) \langle q_k, u \rangle + J(u), \tag{32}$$

where $u^{k+1/2}$ being a minimizer of F implies that there exist $p_{k+1/2} \in \partial J(u^{k+1/2})$ such that

$$\frac{1}{dt}(u^{k+1/2} - u^k) - R(u^k)q_k + p_{k+1/2} = 0.$$

This leads directly to Algorithm 3.

Algorithm 3 (FAGP). Rayleigh quotient minimization of absolutely one-homogeneous functionals

Data: u_0 with $\langle u_0, \mathbf{1} \rangle = 0, \|u_0\|_2 = 1, \Delta t > 0, \epsilon > 0.$

Result: Local minimizer u of the Rayleigh quotient $R = J/H.$

Initialization: $k \leftarrow 0, u^k \leftarrow u_0.$

repeat

$$\left| \begin{array}{l} u^{k+1/2} = \arg \min_{u \in X} F(u) := \frac{1}{2\Delta t} \|u - u^k\|_2^2 - R(u^k) \langle q^k, u \rangle + J(u). \\ u^{k+1} = u^{k+1/2} / \|u^{k+1/2}\|_2 \end{array} \right.$$

until $\|u^{k+1} - u^k\|_2 < \epsilon;$

end while

Remark 1. Notice that since J and H are absolutely one-homogeneous, their subgradients do not change by the normalization step of the flow, i.e., $q_{k+1} = q_{k+1/2}$ and $p_{k+1} = p_{k+1/2}$. We also have $R(u^{k+1}) = R(u^{k+1/2})$ as a quotient of two one-homogeneous functionals.

The sequence u^k of Algorithm 3 satisfies the following properties:

1. $1 = \|u^k\|_2^2 \leq \langle u^{k+1/2}, u^k \rangle \leq \|u^{k+1/2}\|_2^2.$
2. $\|u^{k+1} - u^k\|_2 \leq \|u^{k+1/2} - u^k\|_2.$
3. Monotonicity: $R(u^{k+1}) \leq R(u^k).$
4. Compactness: $\|u^{k+1} - u^k\|_2^2 \rightarrow 0.$

Convergence. It is shown that Algorithm 3 converges to a (double nonlinear) eigenfunction, in the sense of (15).

Theorem 7 (Convergence). *Let u_0 in X and u^k is computed by Algorithm 3. Then there exist $u, p,$ and q in X such that up to a subsequence $u^k \rightarrow u, p_{k+1/2} \rightarrow p, q_k \rightarrow q, \|u\|_2 = 1,$ and*

$$p = R(u)q, \quad q \in \partial H(u), \quad p \in \partial J(u). \tag{33}$$

Further relations to calibrable sets and variants of Algorithm 3 for Cheeger cut minimization on graphs are provided in detail in Feld et al. (2019).

Cohen-Gilboa (CG)

Nonlinear eigenvalue problems emerge naturally also in physical modeling of nonlinear phenomena in fields such as photoelectronics and quantum physics. In 1895 Korteweg-de Vries formulated a mathematical model of waves on shallow water surfaces which were previously described by Russell. The KdV equation, as expressed in Zabusky and Kruskal (1965), is

$$u_t + uu_x + \delta^2 u_{xxx} = 0,$$

with δ a small real scalar. Reformulating this expression for a stationary wave yields

$$-u_{XX} = \lambda \left(-cu + \frac{u^2}{2} \right), \quad (34)$$

where c is the wave velocity, $X = x - ct$, and $\lambda = \delta^{-2}$. Naturally, λ can be understood as an eigenvalue. The solution to this equation models well a family of solitary waves referred to as solitons. In this specific case, one can obtain an analytic solution:

$$u(X) = 3c \cdot \operatorname{sech}^2 \left(\frac{\sqrt{c \cdot \lambda} X}{2} \right).$$

In recent decades there has been a growing research concerning nonlinear physical models, where more complex nonlinear eigenvalue problems emerge, such as the two-dimensional nonlinear Schrodinger equation:

$$u_{xx} + u_{yy} - V_0 \left(\sin^2 x + \sin^2 y \right) u + \sigma |u|^2 u = -\mu u. \quad (35)$$

In Cohen and Gilboa (2018), a method for solving such problems was proposed, following the flows of Nossek and Gilboa (2018) and Aujol et al. (2018). The basic formulation was to solve the (double) nonlinear problem:

$$T(u) = \lambda Q(u), \quad (36)$$

where $T(u) \in \partial J(u)$, $J(u)$ is a convex, proper, lsc regularizing functional and $Q(u)$ is a bounded nonlinear operator, with both $T, Q \in L^2(\Omega)$. The following flow is a natural generalization of Nossek and Gilboa (2018):

$$u_t(t) = M(u(t)), \quad u(t=0) = u_0, \quad (37)$$

where

$$M(u) = s \frac{Q(u)}{\|Q(u)\|_2} - \frac{T(u)}{\|T(u)\|_2}, \tag{38}$$

and $s = \text{sign}(\langle Q(u), T(u) \rangle)$. It can be shown that $\frac{d}{dt} J(t) \leq 0$ a.e. for $t \in (0, \infty)$ and that a steady state admits the nonlinear eigenvalue problem (36).

A problem arises here, where one can reach the null-space of J , thus yielding degenerate solutions with eigenvalues $\lambda = 0$. This did not happen in previous algorithms, which ensured u to be of zero mean and unit norm (or increasing norm with time in Nossek and Gilboa 2018). This prevented the case where u can be a constant function. For (36), however, these assumptions do not necessarily hold; moreover we do not control u directly. Such flows tend to find smoother solutions with low eigenvalues; thus reaching a very smooth degenerate solution is not only a theoretical problem but a phenomenon which is actually encountered in numerical experiments. Thus, one needs to “push” the evolution “away” from degenerate solutions. This is formulated in general by defining a subspace which does not include all eigenfunctions with zero eigenvalues. We would like our flow to always stay in that subspace. An additional term is added to the flow, which directs it toward this subspace. Let us explain it in more details for the case where J is the Dirichlet energy; hence $T(u) = -\Delta u$. We thus want to solve

$$-\Delta u = \lambda Q(u). \tag{39}$$

This is an eigenvalue problem with left-sided linear operator and right-sided nonlinear operator (common in physics). For Neumann boundary conditions, the null-space of J is the space of constant functions. Therefore, the following energy is defined:

$$E(u) = \frac{1}{2} \langle Q(u), 1 \rangle^2, \tag{40}$$

with

$$\partial E = \langle Q(u), 1 \rangle \partial Q,$$

and ∂Q is the variational derivative of $\langle Q(u), 1 \rangle$. We would like $E(u) = 0$ at steady state to ensure we obtain a meaningful solution. A variant of a gradient descent with respect to E is defined by

$$u_t = C(u) \tag{41}$$

where

$$C(u) = -\partial_u E + \frac{\langle \partial_u E, T(u) \rangle}{\|T(u)\|_2^2} T(u). \tag{42}$$

It ensures one decreases E while not increasing J . We call this the complementary flow. Let us compute the time derivatives of J and E :

$$\begin{aligned} \frac{d}{dt} J(u) &= \langle T(u), u_t \rangle = \langle T(u), C(u) \rangle \\ &= \langle T(u), -\partial_u E + \frac{\langle \partial_u E, T(u) \rangle}{\|T(u)\|_2^2} T(u) \rangle = 0. \end{aligned} \tag{43}$$

For E we have

$$\begin{aligned} \frac{d}{dt} E(u) &= \langle \partial_u E, u_t \rangle = \langle \partial_u E, C(u) \rangle \\ &= -\|\partial_u E\|_2^2 + \frac{\langle \partial_u E, T(u) \rangle^2}{\|T(u)\|_2^2} \leq 0, \end{aligned} \tag{44}$$

where the last inequality follows Cauchy-Schwarz. We thus can merge the main flow (37) and the complementary one (41), with some weight parameter α to obtain the final flow:

$$u_t = M(u) + \alpha C(u), \tag{45}$$

where $\alpha \in \mathbb{R}_+$ and $M(u)$ and $C(u)$ are defined in (38), and (42), respectively. This combined flow admits $(d/dt)J(u) \leq 0$ and $(d/dt)E(u) \leq 0$ (for α large enough). Numerically, iterations which follow this flow are provided in Cohen and Gilboa (2018), using the following adaptive time step for the main flow:

$$dt_M = 2 \frac{\langle \Delta u^k, M(u^k) \rangle}{\|\nabla M(u^k)\|_2^2}, \tag{46}$$

and an adaptive step size for the complementary flow

$$dt_C = -\frac{E(u^{k+\frac{1}{2}})}{\langle \partial E(u^{k+\frac{1}{2}}), C(u^{k+\frac{1}{2}}) \rangle}. \tag{47}$$

The choice of dt_C was such that it approximates in a single step $E(u) \approx 0$, within a first Taylor approximation. The numerical algorithm, a dissipating flow with respect to the energy term J (ensured to be nonincreasing), is shown in Algorithm 4. Since it is basically an explicit scheme with carefully chosen time steps, each iteration requires a low computational effort.

Algorithm 4 (CG). Nonlinear eigenpair generation for the Laplacian problem:

$$-\Delta u = \lambda Q(u)$$

Data: $u_0, Q(u), \epsilon > 0$.

Result: Eigenfunction and eigenvalue, $\{u^k, \lambda^k\}$, where $\lambda^k = \langle T(u), u \rangle / \langle Q(u), u \rangle$.

Initialization: $k \leftarrow 1, u^k \leftarrow u_0, T(u) = -\Delta u$.

Set $dt_C(u_0)$ according to (47).

$$u^1 \leftarrow u^0 + dt_C(u_0) \cdot C(u^0).$$

repeat

Set dt_M according to (46) and $M(u^k)$ according to (38).

$$u^{k+\frac{1}{2}} \leftarrow u^k + dt_M \cdot M(u^k).$$

Set dt_C according to (47) and $C(u^{k+\frac{1}{2}})$ according to (42).

$$u^{k+1} \leftarrow u^{k+\frac{1}{2}} + dt_C \cdot C(u^{k+\frac{1}{2}}).$$

until $\|u^{k+1} - u^k\|_2 < \epsilon$;

Bungert-Hait-Papadakis-Gilboa (BHPG)

The last algorithm presented here is related to very general and complex nonlinear operators, which often cannot be expressed analytically. In Hait-Fraenkel and Gilboa (2019) and Bungert et al. (2020), the operators considered were nonlinear denoisers, which can be based on classical algorithms or on deep neural networks.

The setting is as follows. Let $T : \mathcal{H} \rightarrow \mathcal{H}$ be a generic (nonlinear) operator on a real Hilbert space \mathcal{H} with norm $\| \cdot \|$. In the case of a neural network, one typically has $\mathcal{H} = \mathbb{R}^n$, equipped with the Euclidean norm. We aim at solving the nonlinear eigenproblem (6):

$$T(u) = \lambda u,$$

where $u \in \mathcal{H}$ and $\lambda \in \mathbb{R}$ denote the eigenvector and eigenvalue, respectively. Since the operator assumed here is very general and is not based on any energy functional, one needs to resort to a very simple iterative process, which does not involve any minimization. Such a simple algorithm exists for the linear case, the power method.

Linear power method is a simple classical algorithm for solving linear eigenvalue problems $Lu = \lambda u$, where $u \in \mathbb{R}^n$ is a vector and $L \in \mathbb{R}^{n \times n}$ is a diagonalizable matrix. Given some initial condition $u_0, k \leftarrow 0, u^k \leftarrow u_0$, the following process is iterated until convergence:

$$u^{k+1} \leftarrow \frac{Lu^k}{\|Lu^k\|_2}, \quad k \leftarrow k + 1. \tag{48}$$

Under mild conditions, it is known to converge to the eigenvector with the largest eigenvalue, although convergence is slow. A straightforward analogue of this process for the nonlinear case, having an operator $T(u)$, is to initialize similarly and to iterated until convergence:

$$u^{k+1} \leftarrow \frac{T(u^k)}{\|T(u^k)\|_2}, \quad k \leftarrow k + 1. \tag{49}$$

One can analyze this process more easily in a restricted nonlinear case, where J is an absolutely one-homogeneous functional, based on a proximal operator of J :

$$\text{prox}_\alpha^J(u) := \arg \min_{v \in \mathcal{H}} \frac{1}{2} \|v - u\|^2 + \alpha J(v), \tag{50}$$

where $u \in \mathcal{H}$ and $\alpha > 0$ denotes the regularization parameter. The operator is a classical variational denoiser:

$$T(u) = \text{prox}_\alpha^J(u), \tag{51}$$

which for $J = TV$ coincides with the ROF denoising model (Rudin et al. 1992). In Bungert et al. (2020), it was shown that the process is well defined for a range of parameters α , that the energy is decreasing, $J(u^{k+1}) \leq J(u^k)$, along with a full proof of convergence to a nonlinear eigenvector, in the sense of (6).

For more complex nonlinear operators, however, certain modifications are required. A critical issue is the range of the operator. Unlike linear or homogeneous operators, general nonlinear operators often are expected to perform only in a certain range. This is certainly true in neural networks, where the range is dictated implicitly by the range of the images in the training set. Thus normalization by the norm, as in (49), can drastically change the range of u^k and cause unexpected behavior of the operator. Furthermore, the mean value of u^k is a significant factor. For denoisers, we often expect that a denoising operation does not change the mean value of the input image, that is

$$\langle T(u), 1 \rangle = \langle u, 1 \rangle. \tag{52}$$

It can be shown that for any vector $u \neq 0$ with nonnegative entries and a denoiser T admitting (52), if u is an eigenvector, then $\lambda = 1$. Another issue is the invariance to a constant shift in illumination. We expect the behavior of T to be invariant to a small global shift in image values. That is, $T(u + c) = T(u) + c$, for any $c \in \mathbb{R}$, such that $(u + c) \in \mathcal{H}$.

We thus relax the basic eigenproblem (6) as follows:

$$T(u) - \overline{T(u)} = \lambda(u - \bar{u}), \tag{53}$$

where $\lambda \in \mathbb{R}$, $\bar{u} = \langle 1, u \rangle / |\Omega|$ is the mean value of u over the image domain Ω . Note that now (relaxed) eigenvectors, admitting (53), can have any eigenvalue, keeping the assumptions on T stated above. In addition, if u is an eigenvector, so is $u + c$, as expected for operators with invariance to global value shifts. A suitable Rayleigh quotient, associated with the relaxed eigenvalue problem (53), is

$$R^\dagger(u) = \frac{\langle u - \bar{u}, T(u) - \overline{T(u)} \rangle}{\|u - \bar{u}\|_2^2}, \tag{54}$$

which still has the property that $\lambda = R^\dagger(u)$ whenever u fulfills (53). The modified nonlinear power method is detailed in Algorithm 5, aiming at computing a relaxed eigenvector (53) by explicitly handling the mean value and keeping the norm of the initial condition. We found this adaptation to perform well on denoising networks.

Algorithm 5 (BHPG). Nonlinear power method for nonhomogeneous operators

Data: $u_0, \epsilon > 0$.

Result: Relaxed eigenpair (u^*, λ^*) in the sense of (53), where $u^* = u^k, \lambda^* = R^\dagger(u^*)$, with R^\dagger defined in (54).

Initialization: $k \leftarrow 0, u^k \leftarrow u_0$.

repeat

$$\left| \begin{array}{l} u^{k+1} \leftarrow T(u^k). \\ u^{k+1} \leftarrow u^{k+1} - \overline{u^{k+1}}. \\ u^{k+1} \leftarrow \frac{u^{k+1}}{\|u^{k+1}\|} \|u_0 - \bar{u}_0\|. \\ u^{k+1} \leftarrow u^{k+1} + \overline{u^k}, \quad k \leftarrow k + 1. \end{array} \right.$$

until $\|u^{k+1} - u^k\|_2 < \epsilon$;

Evaluation and Examples

We present here several results of the algorithms presented earlier. First we discuss how the numerical solutions can be evaluated. Then we show several numerical examples related to image processing, learning, and physics.

Global and Local Measures

Since there is often no ground truth or analytic solutions for nonlinear eigenvalue problems, we need to find alternative ways to determine whether the algorithm converged to an eigenfunction. Often exact convergence is very slow; thus knowing that you approximately reached an eigenfunction numerically may also speed up the algorithm and serve as a good stopping criterion for the iterative process.

One general formulation for any operator T is by the angle (see Nossek and Gilboa 2018). For eigenvectors, vectors u and $T(u)$ are collinear. Thus their respective angle is either 0 (for positive eigenvalues) or π (for negative eigenvalues). Since both u and $T(u)$ are real, eigenvalues are also real. Thus, the angle is a simple scalar measure that quantifies how close u and $T(u)$ are to collinearity. We define the angle θ between u and $T(u)$ by

$$\cos(\theta) = \frac{\langle u, T(u) \rangle}{\|u\| \|T(u)\|}. \tag{55}$$

Fig. 1 Global measure θ , (55). Measures the angle between u and $T(u)$. For $\theta = 0$, we have a precise eigenfunction (also for 180 degrees, negative eigenvalues)

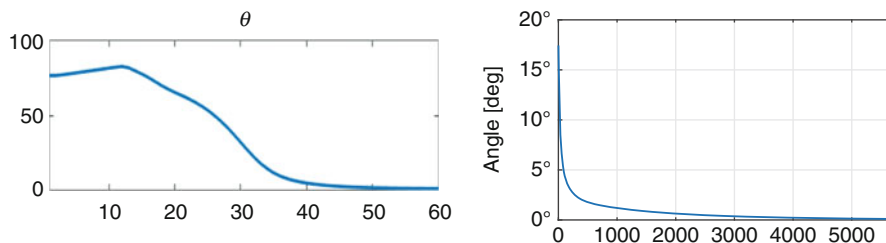
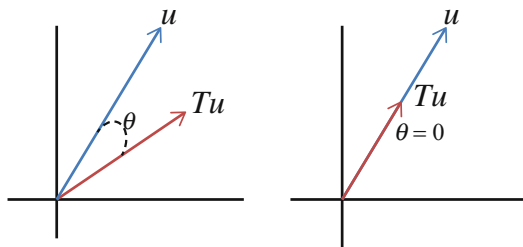


Fig. 2 θ (degrees) as a function of iterations, for (NG) flow, $J = TGV$, and for (CG) flow, Nonlinear Schrodinger equation. (Taken from Nossek and Gilboa 2018 and Cohen and Gilboa 2018)

See Fig. 1 for an illustration of θ . In most cases discussed here, we have positive eigenvalues; thus we aim to reach an angle close to 0. In Fig. 2 we show two examples of the behavior of theta over time for (NG) and (CG) algorithms. Note that θ may not be monotonic and may increase in some time range. The angle θ is a good global measure. In the iterative algorithms, it can be used as a stopping criterion. Instead of requiring $\|u^{k+1} - u^k\|_2 < \epsilon$, one can require reaching a small enough theta $\theta < \theta_{\text{thres}}$. In our studies we often regard a function with $\theta < \pi/360$ ($\frac{1}{2}$ degree) as a numerical eigenfunction.

One may also like to have a local measure. Usually there is no precise pointwise convergence of $(T(u))(x) = \lambda u(x), \forall x$. A good way to see how spatially the function is close to an eigenfunction is by examining the ratio:

$$\Lambda(x) = \frac{T(u)}{u}, \quad \forall u(x) \neq 0.$$

At full convergence we should have $\Lambda(x) \equiv \lambda$. The deviation map from a constant function reveals the areas where the numerical approximation is less accurate. To avoid dividing by values close to 0, one may compute this map only for $u(x) > \delta$, where δ is a small constant. In Fig. 3 we show two examples of this ratio, when one obtains a function close (but not precisely) an eigenfunction and for a case with full convergence.

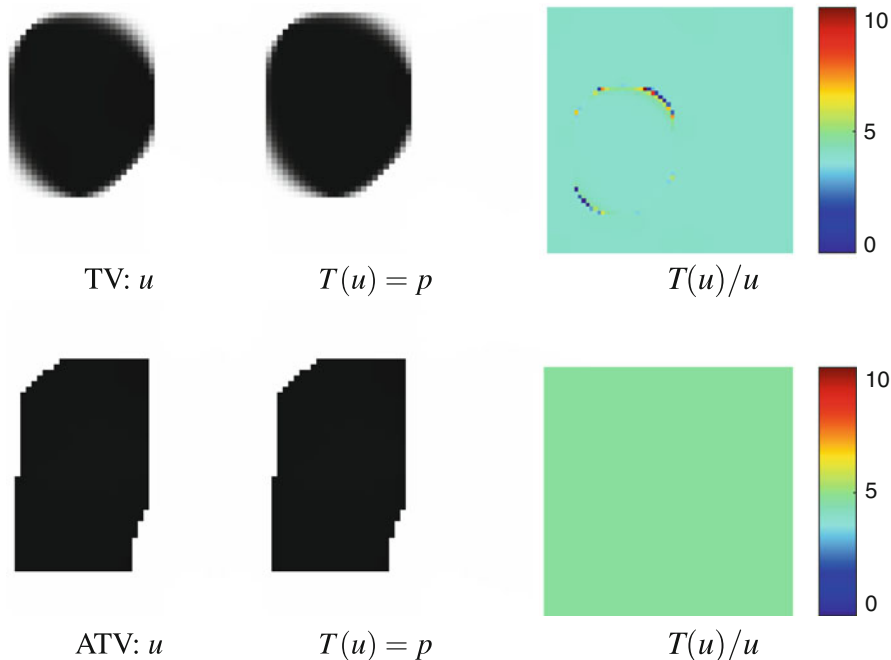


Fig. 3 Local measure $\Lambda(x) = T(u)/u$. At convergence $T(u) = \lambda u$; thus for any $u \neq 0$, we can examine the ratio $\Lambda(x)$, which should be a constant function of value $\lambda, \forall x$. Top row, algorithm did not fully converge yet, u is close to an eigenfunction for isotropic TV, and the ratio (right) exposes areas where there is deviation from a constant. Bottom row, a converged eigenfunction for anisotropic TV. The ratio image is constant, up to numerical precision. (Taken from Aujol et al. 2018)

Numerical Examples

We show some numerical examples of the algorithms presented above. In Fig. 4 some instances along the iteration process of (NG) are shown for the TV and TGV regularizers. At convergence we get structures which are known in the literature to be eigenfunctions induced by these functionals. In Fig. 5 we show an example of the nonlinear power method (BHPG) applied to FFDNet (Zhang et al. 2018), a popular deep neural-network denoiser. We reach an eigenfunction which turns out to be a very good candidate for denoising (reaches PSNR of 44dB, compared to the horse image in the initial condition, which reached only PSNR=30dB). In Fig. 6 two examples of NG and CG flows are shown. Eigenfunctions on graphs are very useful for segmentation, when using graph (or nonlocal) TV for J ; it is seen in Fig. 7 how (AGP) flow solves well the two-moon problem. Starting with a noisy initial condition (blue and red represent positive and negative values), the algorithm converges to an eigenfunction which approximates well the Cheeger cut problem.

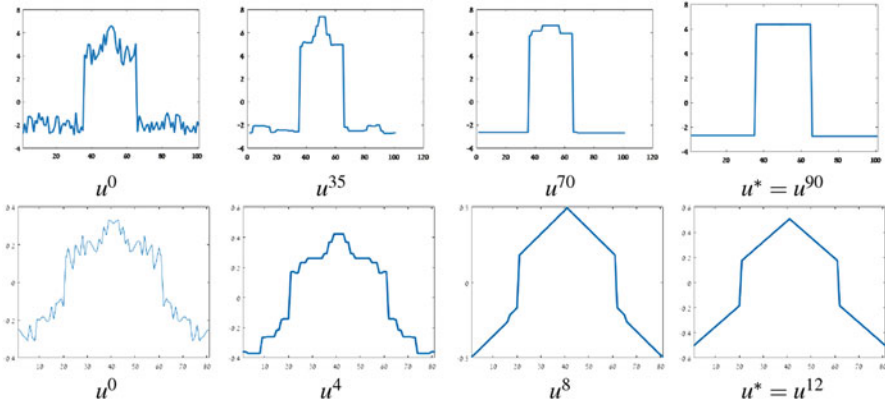


Fig. 4 Two examples of the (NG) flow. Top row $J = TV$, bottom row $J = TGV$ of order 2 (Bredies et al. 2010). (Taken from Nossek and Gilboa 2018)

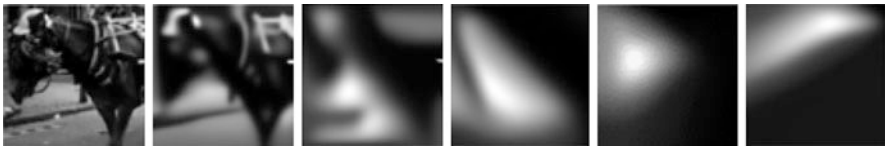


Fig. 5 Nonlinear power method evolution (BHPG) for a denoising neural-network FFDNet (Zhang et al. 2018). Converged eigenfunction ($\lambda = 1$), right, is a highly stable structure for the network. (Taken from Bungert et al. 2020)

In Figs. 8 and 9, we show the resilience of eigenfunctions against noise, esp. when denoised by the matching regularizer J or operator T . In Fig. 8 an eigenfunction of TV was denoised using three classical algorithms. Spectral TV (Gilboa 2014), which is based on the TV regularizer, is most fit to denoise such functions. In Fig. 9 we see a similar trend for EPLL denoiser. Here we have the most stable and unstable eigenfunctions (depending on their eigenvalues) and results of natural images, which are in between, with respect to denoising results. This gives insight on the priors of the denoiser, with respect to the expected spatial structures. Also adversarial examples can be obtained.

Conclusion, Discussion and Open Problems

In this chapter several methods for solving nonlinear eigenvalue problems are presented. Such problems appear in wide and diverse fields of signal and image processing, classification and learning, and nonlinear physics. It is shown how some fundamental concepts of linear eigenvalue problems carry out to the nonlinear case. Specifically, the generalized Rayleigh quotient is a key notion, where eigenfunctions serve as its critical points. A common theme of the presented algorithms is the use of

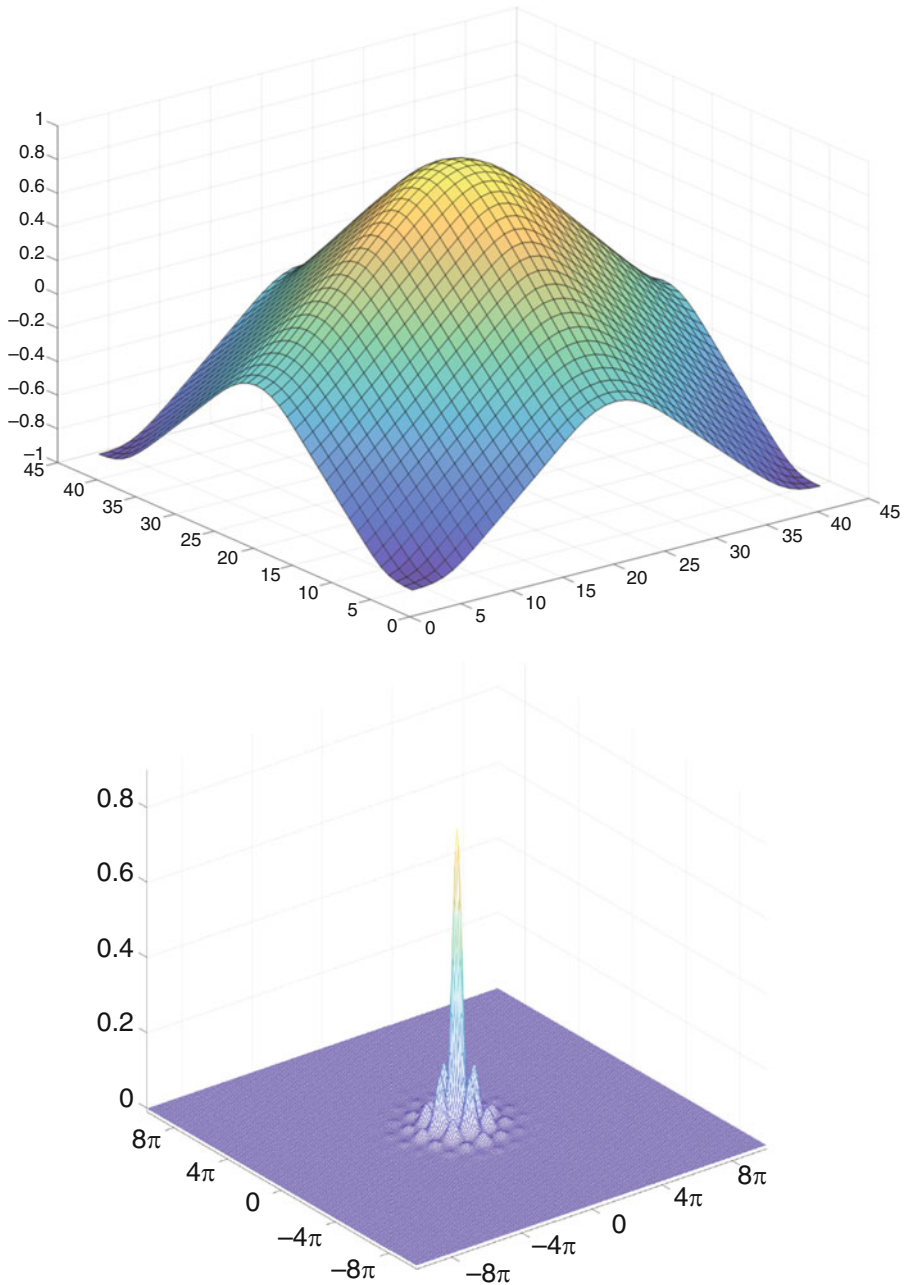


Fig. 6 EF induced by TGV (left, (NG) flow) and EF of the 2D nonlinear Schrodinger equation (35) (right, (CG) flow). (Taken from Nossek and Gilboa 2018 and Cohen and Gilboa 2018)

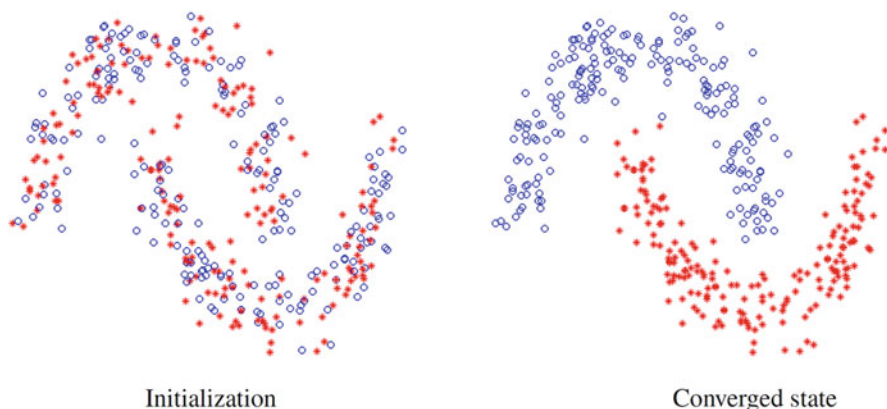


Fig. 7 Results of the flow for TV defined on graphs based on point cloud distances. The processes converge to natural clustering of the data. (Taken from Aujol et al. 2018)

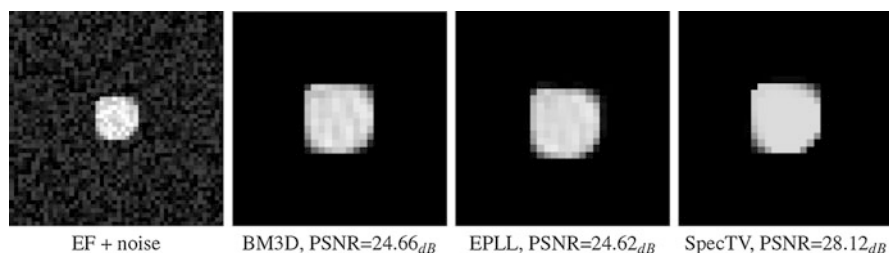


Fig. 8 An eigenfunction obtained by (NG) algorithm for TV. These structures are highly stable in denoising and most suitable for the regularizer (here TV). Here it is shown that for additive white Gaussian noise, spectral TV (Gilboa 2014) recovers well the signal, compared to well-designed classical denoisers BM3D (Dabov et al. 2007) and EPLL (Zoran and Weiss 2011). (Taken from Nossek and Gilboa 2018)

an (often long) iterative process to compute a single eigenfunction. The process can sometimes be understood as a discrete realization of a continuous nonlinear PDE. These nonlinear flows may emerge as gradient descent of a certain energy. However, this energy is always non-convex and has many local minima (each of them is an eigenfunction). Naturally, this implies that the selection of the initial condition is critical to the computation. This is actually true for all iterative processes presented here, even if they are not directly based on a non-convex energy. We would like to highlight several challenges this emerging field is still facing with.

We list below the main intriguing issues and open problems:

1. **Initial condition.** What are the effects of the initial condition to the computation process? Can a link be formulated between the initial condition and the obtained eigenfunction? Is it related to a decomposition of the initial condition into eigenfunctions, in an analogue manner to the linear case? Are there special

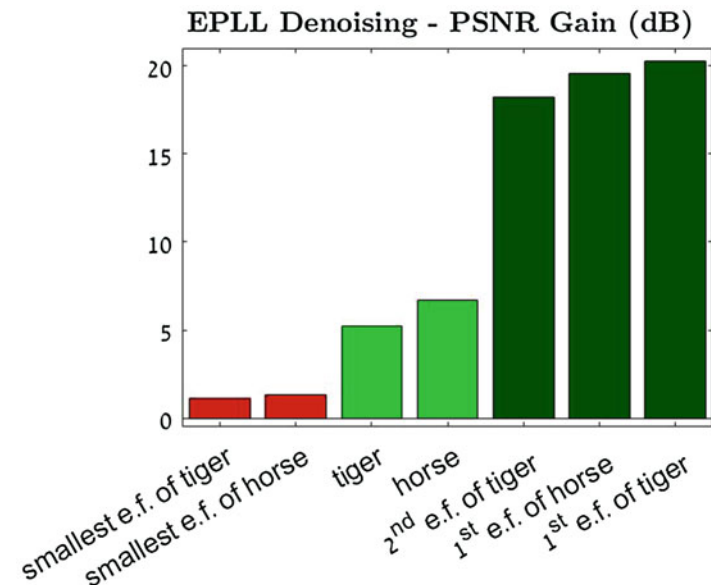


Fig. 9 Nonlinear power method for EPLL denoiser. PSNR gain: eigenfunctions vs. natural images, $var_{noise} = \frac{1}{5} var_{img}$. (Taken from Hait-Fraenkel and Gilboa 2019)

characteristics to the flow when random noise serves as initial condition? Is noise a good choice and in what sense?

2. **Mapping the eigenfunction landscape of a nonlinear operator.** Can one characterize analytically eigenfunctions for a broad family of operators? This was successfully performed for TV (mainly in 2D). For more complex operators and complicated domains or graphs, this is still an open problem. For a given operator, how to design numerically algorithms which span well its eigenfunctions? We have shown that eigenfunctions of large and small eigenvalues can be computed; however reaching middle-range eigenvalues is highly nontrivial without prohibitively large computational efforts (passing through all eigenvalues in ascending/descending order).
3. **Spectral decomposition.** Can a general theory be developed related to the decomposition of a signal into nonlinear eigenfunctions? For the case of one-homogeneous functionals, it was shown how gradient-descent flows can be used for decomposition (see Gilboa 2014, Burger et al. 2016, and Bungert et al. 2019a). A similar phenomenon was observed for the p-Laplacian case in Cohen and Gilboa (2020). Can this be extended to gradient descent of general convex functionals? Can these flows be used to generate multiple eigenfunctions in a much more efficient manner?
4. **Convergence rates.** Until now the algorithms presented here did not deal with convergence rates. They are inherently quite slow; sometimes hundreds or even thousands of iterations are needed in order to numerically converge.

A first analysis of the convergence rate of nonlinear power methods for one-homogeneous functionals is in Bungert et al. (2020). This area surely requires additional focus.

5. **Correspondence to the linear case.** It was shown that the extended definition of the Rayleigh quotient generalizes very well in the nonlinear setting. Are there additional properties related to eigenvalue analysis that can be generalized? For instance, for the power method, we know in the linear case that the method converges to the eigenfunction with the largest eigenvalue (which is part of the initial condition). We see a similar trend in the nonlinear case, where large eigenvalues are reached. Can this be formalized?
6. **Neural networks as operators.** Last but not least, can neural networks benefit from this research field? We have shown in Bungert et al. (2020) that one can treat an entire neural network (intended for denoising) as a single complex nonlinear operator and find some of its eigenfunctions. They represent highly stable and unstable modes (depending on the eigenvalue). Can additional insights be gained by analyzing eigenfunctions of deep neural networks? How can eigenfunctions be defined for classification networks (where the input and output dimensions are very different)? One direction is to develop singular value decomposition into a nonlinear setting, following the earlier work of Benning and Burger (2013). One can also analyze eigenfunctions between layers in the net, the effect of gradient descent (or its stochastic version) on eigenfunctions, and more. For variational networks, the authors of Effland et al. (2020) and Kobler et al. (2020) have shown interesting insights on the learned regularizers can be gained.

Acknowledgments This work was supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 777826, by the Israel Science Foundation (Grant No. 534/19) and by the Ollendorff Minerva Center.

References

- Aujol, J.F., Gilboa, G., Papadakis, N.: Theoretical analysis of flows estimating eigenfunctions of one-homogeneous functionals. *SIAM J. Imaging Sci.* **11**(2), 1416–1440 (2018)
- Bellettini, G., Caselles, V., Novaga, M.: The total variation flow in \mathbb{R}^n . *J. Differ. Equ.* **184**(2), 475–525 (2002)
- Benning, M., Burger, M.: Ground states and singular vectors of convex variational regularization methods. *Methods Appl. Anal.* **20**(4), 295–334 (2013)
- Bozorgnia, F.: Convergence of inverse power method for first eigenvalue of p-laplace operator. *Numer. Funct. Anal. Optim.* **37**(11), 1378–1384 (2016)
- Bozorgnia, F.: Approximation of the second eigenvalue of the p -laplace operator in symmetric domains. *arXiv preprint arXiv:190713390* (2019)
- Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
- Brezis, H.: *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*. North Holland (1973)
- Bungert, L., Burger, M., Chambolle, A., Novaga, M.: Nonlinear spectral decompositions by gradient flows of one-homogeneous functionals. *Anal. PDE* (2019a). To appear

- Bungert, L., Burger, M., Tenbrinck, D.: Computing nonlinear eigenfunctions via gradient flow extinction. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 291–302. Springer (2019b)
- Bungert, L., Hait-Fraenkel, E., Papadakis, N., Gilboa, G.: Nonlinear power method for computing eigenvectors of proximal operators and neural networks. arXiv preprint arXiv:200304595 (2020)
- Burger, M., Gilboa, G., Moeller, M., Eckardt, L., Cremers, D.: Spectral decompositions using one-homogeneous functionals. *SIAM J. Imaging Sci.* **9**(3), 1374–1408 (2016)
- Cohen, I., Gilboa, G.: Energy dissipating flows for solving nonlinear eigenpair problems. *J. Comput. Phys.* **375**, 1138–1158 (2018)
- Cohen, I., Gilboa, G.: Introducing the p-laplacian spectra. *Signal Process.* **167**, 107281 (2020)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Effland, A., Kobler, E., Kunisch, K., et al.: Variational networks: an optimal control approach to early stopping variational methods for image restoration. *J Math Imaging Vis.* **62**, 396–416 (2020)
- Feld, T., Aujol, J.F., Gilboa, G., Papadakis, N.: Rayleigh quotient minimization for absolutely one-homogeneous functionals. *Inverse Probl.* **35**(6), 064003 (2019)
- Gautier, A., Tudisco, F., Hein, M.: The perron–frobenius theorem for multihomogeneous mappings. *SIAM J. Matrix Anal. Appl.* **40**(3), 1179–1205 (2019)
- Gautier, A., Hein, M., Tudisco, F.: Computing the norm of nonnegative matrices and the log-sobolev constant of markov chains. arXiv preprint arXiv:200202447 (2020)
- Gilboa, G.: A spectral approach to total variation. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 36–47. Springer (2013)
- Gilboa, G.: A total variation spectral framework for scale and texture analysis. *SIAM J. Imaging Sci.* **7**(4), 1937–1961 (2014)
- Gilboa, G.: *Nonlinear Eigenproblems in Image Processing and Computer Vision*. Springer, Cham (2018)
- Hait-Fraenkel, E., Gilboa, G.: Numeric solutions of eigenvalue problems for generic nonlinear operators. arXiv preprint arXiv:190912775 (2019)
- Hein, M., Bühler, T.: An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In: *Advances in Neural Information Processing Systems*, pp. 847–855 (2010)
- Kobler, E., Effland, A., Kunisch, K., Pock, T.: Total deep variation: a stable regularizer for inverse problems. arXiv preprint arXiv:200608789 (2020)
- Meyer, Y.: Oscillating patterns in image processing and in some nonlinear evolution equations. The 15th Dean Jacqueline B. Lewis Memorial Lectures. American Mathematical Society, Providence (2001)
- Nosseck, R.Z., Gilboa, G.: Flows generating nonlinear eigenfunctions. *J. Sci. Comput.* **75**(2), 859–888 (2018)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
- Szlam, A., Bresson, X.: Total variation and Cheeger cuts. In: International Conference on Machine Learning (ICML’10), pp. 1039–1046 (2010)
- Vassilis, A., Jean-François, A., Dossal, C.: The differential inclusion modeling fista algorithm and optimality of convergence rate in the case $b \leq 3$. *SIAM J. Optim.* **28**(1), 551–574 (2018)
- Zabusky, N.J., Kruskal, M.D.: Interaction of “solitons” in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.* **15**(6), 240 (1965)
- Zhang, K., Zuo, W., Zhang, L.: FFDNet: toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process.* **27**(9), 4608–4622 (2018)
- Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: International Conference on Computer Vision, pp. 479–486. IEEE (2011)



Xianfeng Gu, Na Lei, and Shing-Tung Yau

Contents

Introduction	1661
Related Works	1662
Optimal Transport Map	1662
Generative Models	1663
Optimal Transport Theory	1665
Monge's Problem	1665
Kantorovich's Approach	1667
Brenier's Approach	1668
McCann's Displacement	1669
Benamou-Brenier Dynamic Fluid	1670
Otto's Calculus	1671
Regularity of Optimal Transport Maps	1673
Computational Algorithm	1676
Semi-discrete Optimal Transport Map	1676
Damping Newton's Method	1680
Monte-Carlo Method	1684
Manifold Distribution Principle	1686
Manifold Learning	1688
ReLu Deep Neural Network	1690
AutoEncoder	1692
Generative Adversarial Networks	1694

X. Gu (✉)

Stony Brook, Stony Brook University, Stony Brook, NY, USA

e-mail: gu@cs.stonybrook.edu

N. Lei

Dalian University of Technology, Dalian, China

e-mail: nalei@dlut.edu.cn

S.-T. Yau

Harvard University, Cambridge, MA, USA

e-mail: yau@math.harvard.edu

Competition vs. Collaboration	1694
Memorization vs. Learning	1696
Mode Collapsing	1697
AE-OT Model	1701
Conclusion	1704
References	1704

Abstract

Optimal transport plays a fundamental role in deep learning. Natural data sets have intrinsic patterns, which can be summarized as the manifold distribution principle: a natural class of data can be treated as a probability distribution on a low-dimensional manifold, embedded in a high-dimensional ambient space. A deep learning system mainly accomplishes two tasks: manifold learning and probability distribution learning.

Given a manifold X , all the probability measures on X form an infinite dimensional manifold, the so-called Wasserstein space. Optimal transport assigns a Riemannian metric on the Wasserstein space, the so-called Wasserstein metric, and defines Otto's calculus, such that variational optimization can be carried out in the Wasserstein space $\mathcal{P}(X)$. A deep learning system learns the distribution by optimizing some functionals in the Wasserstein space $\mathcal{P}(X)$; therefore optimal transport lays down the theoretic foundation for deep learning.

This work introduces the theory of optimal transport and the profound relation between Brenier's theorem and Alexandrov's theorem in differential geometry via Monge-Ampère equation. We give a variational proof for Alexandrov's theorem and convert the proof to a computational algorithm to solve the optimal transport maps. The algorithm is based on computational geometry and can be generalized to general manifold setting.

Optimal transport theory and algorithms have been extensively applied in the models of generative adversarial networks (GANs). In a GAN model, the generator computes the optimal transport map (OT map), while the discriminator computes the Wasserstein distance between the generated data distribution and the real data distribution. The optimal transport theory shows the competition between the generator and the discriminator is completely unnecessary and should be replaced by collaboration. Furthermore, the regularity theory of optimal transport map explains the intrinsic reason for mode collapsing.

A novel generative model is introduced, which uses an autoencoder (AE) for manifold learning and OT map for probability distribution transformation. This AE-OT model improves the theoretical rigor and transparency, as well as the computational stability and efficiency; in particular, it eliminates the mode collapsing.

Keywords

Explainable deep learning · Optimal Transport · Convex Geometry · Generative adversarial networks · Manifold learning · Monge-Ampère Equation

Introduction

Deep learning is the mainstream technique for many machine learning tasks, including image recognition, machine translation, speech recognition, and so on. Despite its great success, the theoretical understanding on how it works remains primitive. Many fundamental problems need to be solved, and many profound questions need to be answered.

In this chapter, we focus on a geometric view of optimal transport (OT) to understand deep learning models, such as generative adversarial networks (GANs). Especially, we aim at answering the following basic questions:

Question 1. What does a deep learning system really learn? The system learns the probability distributions on manifolds. Each natural class of data set can be treated as a point cloud in the high-dimensional ambient space, and the point cloud approximates a special probability measure defined on a low-dimensional manifold. The system learns two things: one is the manifold structure and the other is the distribution on the manifold. The manifold structure is represented by the encoding and decoding maps, which map between the manifold and the latent space. In generative models, such as GANs, the probability distributions are represented by the transport mappings from a predefined white noise (such as a Gaussian distribution, which can be easily generated from a uniform distribution) to the data distribution, either in the latent space or on the data manifold.

Question 2. How does a deep learning system really learn? All the probability distributions on a manifold Σ form an infinite dimensional space $\mathcal{P}(\Sigma)$, the so-called Wasserstein space. A deep learning system performs optimization in the space of $\mathcal{P}(\Sigma)$. For example, the principle of maximum entropy searches for a distribution in $\mathcal{P}(\Sigma)$ by optimizing the entropy functional with some constraints obtained by observations. The optimal transport theory defines a Riemannian metric on the probability distribution space $\mathcal{P}(\Sigma)$, and Otto's calculus, such that the Wasserstein distance between measures can be computed explicitly and the variational optimizations can be carried out by these theoretic tools. For example, the discriminator in the WGAN model computes the Wasserstein distance between the real data distribution and the generated data distribution, and the training process follows the Wasserstein gradient flow on $\mathcal{P}(\Sigma)$.

Question 3. How well does a deep learning system really learn? Current deep learning system designs have fundamental flaws; most generative models suffer from mode collapsing. Namely, they keep forgetting some knowledge already learned at the intermediate stage, or they generate unrealistic samples. This can be explained by the regularity theory of optimal transport maps; basically the transport maps are discontinuous, whereas the deep neural networks can only represent continuous maps; therefore either the map misses some connected components of the support of data distribution or covers all the components but also the gaps among them.

From the above short answers, we can see the importance of the theories of manifold and optimal transport for deep learning. In the following, we will briefly review the most related works in section “[Related Works](#),” introduce the theory of optimal transport in section “[Optimal Transport Theory](#),” explain the computational algorithms for optimal transport in details in section “[Computational Algorithm](#),” and after the preparation, we will explain the manifold distribution principle in deep learning and manifold learning by autoencoder in sections “[Manifold Distribution Principle](#)” and “[Manifold Learning](#)”, respectively; and then we use optimal transport view to analyze GAN model and explain the reason for mode collapse and the novel design to eliminate mode collapse in section “[Generative Adversarial Networks](#)”; finally, we conclude the work in section “[Conclusion](#)”.

Related Works

The literature of optimal transport and generative models is huge. Here, we only review the most directly related works.

Optimal Transport Map

Monge-Kantorovich theory has been applied to solve optimal transport problem via linear programming technique (Kantorovich 1948, 2006). The method was intuitively applied for image registration and warping in early research works. This approach was proposed in Rehman et al. (2009); however due to the expensive computational cost, the method can hardly handle the 3D image registration problem efficiently. Optimal transportation map was also applied for texture mapping purposes in Dornik and Tannenbaum (2010), where the surface is initially mapped to the unit sphere conformally, and then the mapping is optimized by a gradient flow with multiple level of resolutions to accelerate the convergence. Since the exact evaluation of Wasserstein distance is expensive, the heat kernel method was applied to approximate it in Solomon et al. (2014, 2015b). In order to extend the problem into large data sets, Cuturi (2013) added an entropic regularizer into the original linear programming problem, and as a result, the regularized problem can be quickly computed with the Sinkhorn algorithm. Then Solomon et al. (2015a) improved the computational efficiency by the introduction of fast convolution.

Recent research works are more based on Monge-Brenier theory (Brenier 1991). Gu et al. used a geometric variational approach to prove Alexandrov theorem in Gu et al. (2016), which is equivalent to the discrete Brenier theorem. The method leads to a constructive algorithm for computing optimal transportation maps in general settings. In (2011), De Goes et al. proposed to use OT for 2D shape reconstruction and simplification; later on they generalized to use capacity-constrained Voronoi tessellation to deal with blue noise processing problem (De Goes et al. 2012). Mérigot (2011) proposed a multi-scale approach to accelerate the computation for large-scale problems. Most of the early works focus on 2D image registration

and processing; recent works generalized them to deal with 3D surfaces by using computational geometric approaches. By incorporating with conformal mapping methods, optimal transportation maps are applied to obtain area-preserving maps in Su et al. (2016). The methods in Yu et al. (2018) can simultaneously balance the area and the angle distortion. Su et al. generalized the algorithm to three-dimensional cases and presented a volume-preserving map in Su et al. (2016), and then in Su et al. (2017) they further gave a volumetric controllable algorithm by OT maps.

While most of the research works deal with optimal transport problems with Euclidean metric, Wang (2004) and Cui et al. (2019) focused on solving the optimal transportation problems in the spherical domain. The method has also been applied for area-preserving brain mapping in Su et al. (2013), which maps the cortical surface onto the unit sphere conformally and then onto the extended complex plane by the stereographic projection. The method has been improved in Nadeem et al. (2017) by using the conformal welding method.

Recent research works also introduce optimal transportation theory in the optical design field. Reflector design problems were summarized as a group of Monge-Ampère equations in Wang (1996, 2004) and Guan et al. (1998). The correspondence between Monge-Ampère equations and reflector design problems was listed as one of the open problems in Yau (1998) and can further be related to optimal transportation theory. Similar researches in lens design situation were introduced in Gutiérrez, Qingbo and Huang (2009). Numerical methods and simulation results of these optical design problems were proposed in Meyron et al. (2018).

Generative Models

Encoder-decoder architecture A breakthrough for image generation comes from the scheme of variational autoencoders (VAEs) (e.g., Kingma and Welling 2013), where the decoders approximate real data distributions from a Gaussian distribution in a variational approach (e.g., Kingma and Welling 2013 and Rezende et al. 2014). Latter Yuri Burda et al. (2015) lower the requirement of latent distribution and propose the importance weighted autoencoder (IWAE) model through a different lower bound. Bin and David (2019) propose that the latent distribution of VAE may not be Gaussian and improve it by firstly training the original model and then generating new latent code through the extended ancestral process. Another improvement of the VAE is the VQ-VAE model (van den Oord and Vinyals 2017), which requires the encoder to output discrete latent codes by vector quantization, and then the posterior collapse of VAEs can be overcome. By multi-scale hierarchical organization, this idea is further used to generate high-quality images in VQ-VAE-2 (Razavi et al. 2019). In Gelly et al. (2018), the authors adopt the Wasserstein distance in the latent space to measure the distance between the distribution of the latent code and the given one and generate images with better quality. Different from the VAEs, the AE-OT model (An et al. 2020) firstly embed the images into the latent space by an autoencoder, and then an extended semi-discrete OT map is computed to generate new latent code based on the fixed ones. Decoded by

the decoder, new images can be generated. Although the encoder-decoder-based methods are relatively simple to train, the generated images tend to be blurry.

Generative adversarial networks The GAN model (Goodfellow et al. 2014) tries to alternatively update the generator, which maps the noise sampled from a given distribution to real images, and the discriminator differentiates the difference between the generated images and the real ones. If the generated images successfully fool the discriminator, we say the model is well trained. Later, Radford et al. (2016) proposes a deep convolutional neural network (DCGAN) to generate images with better quality. While being a powerful tool in generating realistic samples, GANs can be hard to train and suffer from mode collapse problem (Goodfellow 2016). After delicate analysis, Arjovsky et al. (2017) points out that it is the KL divergence the original GAN used that causes these problems. Then the authors introduce the celebrated WGAN, which makes the whole framework easy to converge. To satisfy the Lipschitz continuity required by WGAN, a lot of methods are proposed, including clipping Arjovsky et al. (2017), gradient penalty (Gulrajani et al. 2017), spectral normalization (Miyato et al. 2018), and so on. Later, Wu et al. (2018) use the Wasserstein divergence objective, which get rid of the Lipschitz approximation problem and gets a better result. Instead L_1 cost adopted by WGAN, Liu et.al (2019) propose the WGAN-QC by taking the L_2 cost into consideration. Though various GANs can generate sharp images, they will theoretically encounter the mode collapse or mode mixture problem (Goodfellow 2016; An et al. 2020).

Hybrid models To solve the blurry image problem of encoder-decoder architecture and the mode collapse/mixture problems of GANs, a natural idea is to compose them together. Larsen et al. (2016) propose to combine the variational autoencoder with a generative adversarial network and thus generate images better than VAEs. Makhzani et al. (2015) matches the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior to distribution by a discriminator and then applies the model into tasks like semi-supervised classification and dimensionality reduction. BiGAN by Jeff Donahue and Krähenbühl (2017) uses the discriminator to differentiate both the generated images and the generated latent code. Further, by utilizing the BigGAN generator (Simonyan et al. 2019), the BigBiGAN (Donahue and Simonyan 2019) extends this method to generate much better results. Here we also treat the BourGAN (Xiao et al. 2018) as a hybrid model, because it firstly embeds the images into latent space by Bourgain theorem and then trains the GAN model by sampling from the latent space using the GMM model.

Conditional GANs are another kind of hybrid models that can also be treated as image-to-image transformation. For example, using an encoder-decoder architecture to build the connection between paired images and then differentiating the decoded images with the real ones by a discriminator, Isola et al. (2017) is able to transform images of different styles. Further, SRGAN (Ledig et al. 2017) uses similar architecture to get super resolution images from their low-resolution versions. The SRGAN model utilizes the content loss and adversarial loss. It uses the paired data, and the visually meaningful features used by SRGAN are extracted from the

pre-trained VGG19 network (Simonyan and Zisserman 2014), which makes it not so reasonable under the scenes where the data sets are not included in those used to train the VGG.

Optimal Transport Based Generative Model In (2019) Lei et al. first gave a geometric interpretation to the generative adversarial networks (GANs). By using the optimal transport view of GAN model, they showed that the discriminator computes the Wasserstein distance via the Kantorovich potential and the generator calculates the transport map. For a large class of transportation costs, the Kantorovich potential can give the optimal transportation map by a close-form formula. This shows the adversarial competition can be replaced by collaboration to improve the efficiency and simplicity. In Lei et al. (2020) the authors pointed out that GANs mainly accomplish two tasks: manifold learning and probability distribution transformation. The latter can be carried out using the classical OT method. Then in An et al. (2020), a new generative model based on extended semi-discrete optimal transport was proposed, which avoids representing discontinuous maps by DNNs and therefore effectively prevents mode collapse and mode mixture (Fig. 23).

Numerical Method In this work, we show that the reason that causes the mode collapse in deep learning is indeed the discontinuity of optimal transport map in general. It is very similar to the situation when using the classic numerical method to solve OT map. For instance, the Brenier potential in OT satisfies the Hamiltonian–Jacobi equation which could be continuous. However, its velocity (corresponding to the OT map) satisfying the conservation law is generally discontinuous. For examples, the Benamou–Brenier method (Benamou and Brenier 1999) and Haker–Tannenbaum–Angenent method (Angenent et al. 2003) compute the optimal transport maps based on fluid dynamics.

Optimal Transport Theory

In this subsection, we will introduce basic concepts and theorems in classic optimal transport theory, focusing on Brenier’s approach, and their generalization to the discrete setting. Details can be found in Villani’s book (Villani 2008).

Monge’s Problem

Suppose $X \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}^d$ are two measurable subsets of d -dimensional Euclidean space \mathbb{R}^d and μ, ν are two probability measures defined on X and Y respectively, with density functions

$$d\mu(x) = f(x) dx, \quad d\nu(y) = g(y) dy.$$

Suppose their total measures are equal, $\mu(X) = \nu(Y)$, namely,

$$\int_X f(x)dx = \int_Y g(y)dy. \quad (1)$$

We only consider maps which preserve the measure.

Definition 1 (Measure-Preserving Map). A map $T : X \rightarrow Y$ is *measure preserving* if for any measurable set $B \subset Y$, the set $T^{-1}(B)$ is μ -measurable and $\mu(T^{-1}(B)) = \nu(B)$, i.e.,

$$\int_{T^{-1}(B)} f(x)dx = \int_B g(y)dy. \quad (2)$$

Measure-preserving condition is denoted as $T_{\#}\mu = \nu$, where $T_{\#}\mu$ is the push forward measure induced by T . Suppose $T : X \rightarrow Y$ is differentiable, $T \in C^1(X)$, then the measure-preserving map satisfies the Jacobian equation:

$$\det DT(x) = \frac{f(x)}{g \circ T(x)}. \quad (3)$$

Definition 2 (Transport Cost). Given a *cost function* $c(x, y) : X \times Y \rightarrow \mathbb{R}_{\geq 0}$, which indicates the cost of moving each unit mass from the source to the target, the total *transport cost* of the map $T : X \rightarrow Y$ is defined to be

$$C(T) := \int_X c(x, T(x))d\mu(x). \quad (4)$$

The Monge's problem of optimal transport arises from finding the measure-preserving map that minimizes the total transport cost.

Problem 1 (Monge's Optimal Transport Problem Bonnotte 2013 (MP)). Given a transport cost function $c : X \times Y \rightarrow \mathbb{R}$, find the measure preserving map $T : X \rightarrow Y$ that minimizes the total transport cost

$$(MP) \quad \min_{T_{\#}\mu=\nu} \int_X c(x, T(x))d\mu(x). \quad (5)$$

Definition 3 (Optimal Transport Map). The solution to the Monge's problem is called the *optimal transport map*, whose total transport cost defines the *Wasserstein distance* between μ and ν .

If $c(x, y) = \frac{1}{2}\|x - y\|^2$, the Wasserstein distance is denoted as $\mathcal{W}_2(\mu, \nu)$, then

$$\mathcal{W}_2^2(\mu, \nu) = \min_{T_{\#}\mu=\nu} \frac{1}{2} \int_X |x - T(x)|^2 d\mu(x). \quad (6)$$

Kantorovich's Approach

Depending on the cost function and the measures, the optimal transport map between (X, μ) and (Y, ν) may not exist. For example, suppose μ is atomic $\mu = \delta(x - x_0)$, and $\nu = \sum_{i=1}^k v_i \delta(y - y_i)$ with $\sum_{i=1}^k v_i = 1$, $k > 1$, then the mass concentrated on x_0 has to be split and sent to different y_i 's. Kantorovich relaxed transport maps to *transport plans* or *transport schemes*. A transport plan is represented by a joint probability measure $\rho : X \times Y \rightarrow \mathbb{R}_{\geq 0}$, such that the marginal probability of ρ equals to μ and ν , respectively. Formally, let the projection maps be $\pi_x(x, y) = x$, $\pi_y(x, y) = y$, and then the joint measure class is defined as

$$\Pi(\mu, \nu) := \{\rho : X \times Y \rightarrow \mathbb{R} : (\pi_x)_\# \rho = \mu, (\pi_y)_\# \rho = \nu\} \quad (7)$$

Problem 2 (Kantorovich). Given a transport cost function $c : X \times Y \rightarrow \mathbb{R}$, find the joint probability measure $\rho : X \times Y \rightarrow \mathbb{R}_{\geq 0}$ with marginals μ and ν that minimizes the total transport cost

$$(KP) \quad \min_{\rho \in \Pi(\mu, \nu)} \int_{X \times Y} c(x, y) d\rho(x, y). \quad (8)$$

Kantorovich's problem can be solved using the linear programming method. Due to the duality of linear programming, the (KP) Eq. 8 can be reformulated as the following duality problem (DP):

Problem 3 (Kantorovich Dual). Given a transport cost function $c : X \times Y \rightarrow \mathbb{R}$, find the function $\varphi \in L^1(X)$ and $\psi \in L^1(Y)$, such that

$$(DP) \quad \max_{\varphi, \psi} \left\{ \int_X \varphi(x) d\mu + \int_Y \psi(y) d\nu : \varphi(x) + \psi(y) \leq c(x, y) \right\} \quad (9)$$

The maximum value of Eq. 9 gives the Wasserstein distance. Most existing Wasserstein GAN models are based on the duality formulation under the L^1 cost function.

Definition 4 (c -transform). The c -transform of $\varphi : X \rightarrow \mathbb{R}$ is defined as $\varphi^c : Y \rightarrow \mathbb{R}$:

$$\varphi^c(y) = \inf_{x \in X} (c(x, y) - \varphi(x)). \quad (10)$$

Assume $c(x, y)$ and φ are with C^1 continuity, and then the necessary condition for c -transform is given by

$$\nabla_x c(x, y(x)) - \nabla \varphi(x) = 0. \quad (11)$$

Then the Kantorovich dual problem can be rewritten as

$$(DP) \quad \mathcal{W}_c(\mu, \nu) = \max_{\varphi} \int_X \varphi(x) d\mu + \int_Y \varphi^c(y) d\nu, \tag{12}$$

where φ is called the *Kantorovich’s potential*.

Brenier’s Approach

Given a strictly C^1 convex function $h : \Omega \rightarrow \mathbb{R}$, where Ω is a convex domain in \mathbb{R}^n , the gradient mapping $x \mapsto \nabla h(x)$ is invertible. The inverse mapping is denoted as $(\nabla h)^{-1}$.

Suppose the cost function $c(x, y) = h(x - y)$ where h is a strictly C^1 convex function, then the solution to Kantorovich’s dual problem Eq. 12 satisfies the c -transform condition Eq. 11; hence we obtain the formula for the optimal transport map T ,

$$T(x) = x - (\nabla h)^{-1}(\nabla \varphi(x)). \tag{13}$$

This leads to the following theorem:

Theorem 1 (Villani 2008). *Given μ and ν on a compact domain $\Omega \subset \mathbb{R}^n$, there exists an optimal transport plan ρ for the cost $c(x, y) = h(x - y)$ with h strictly convex. It is unique and of the form $(id, T_{\#})\mu$, provided μ is absolutely continuous and $\partial\Omega$ is negligible. Moreover, there exists a Kantorovich potential φ , and T can be represented as*

$$T(x) = x - (\nabla h)^{-1}(\nabla \varphi(x)).$$

For quadratic Euclidean distance cost, $h(x) = \frac{1}{2}\langle x, x \rangle$, $(\nabla h)^{-1}(x) = x$, then Eq. 13 becomes

$$T(x) = x - \nabla \varphi(x) = \nabla \left(\frac{1}{2}\langle x, x \rangle - \varphi(x) \right) = \nabla u, \tag{14}$$

where the function $u : X \rightarrow \mathbb{R}$ is called the *Brenier’s potential*. In this case, the Brenier’s potential u and the Kantorovich’s potential φ are related by Eq. 14. Assume the Brenier’s potential is C^2 convex, by Jacobian equation Eq. 3, it satisfies the following *Monge-Ampère equation*:

$$\det \left(\frac{\partial^2 u}{\partial x_i \partial x_j} \right) (x) = \frac{f(x)}{g \circ \nabla u(x)} \tag{15}$$

The existence, uniqueness, and the intrinsic structure of the optimal transport map were proven by Brenier (1991).

Theorem 2 (Brenier 1991). *Suppose X and Y are measurable subsets of the Euclidean space \mathbb{R}^d and the transport cost is the quadratic Euclidean distance $c(x, y) = 1/2\|x - y\|^2$. Furthermore μ is absolutely continuous with respect to Lebesgue measure and μ and ν have finite second-order moments,*

$$\int_X |x|^2 d\mu(x) + \int_Y |y|^2 d\nu(y) < \infty, \quad (16)$$

then there exists a convex function $u : X \rightarrow \mathbb{R}$, the so-called Brenier's potential, its gradient map ∇u gives the solution to the Monge's problem,

$$(\nabla u)_\# \mu = \nu. \quad (17)$$

The Brenier's potential is unique up to a constant; hence the optimal mass transport map is unique.

Therefore, finding the optimal transport map is reduced to solving the Monge-Ampère equation.

Problem 4 (Brenier). Suppose X and Y are subsets of the Euclidean space \mathbb{R}^d and the transport cost is the quadratic Euclidean distance. Furthermore μ is absolutely continuous with respect to Lebesgue measure and μ and ν have finite second-order moments; Find a convex function $u : X \rightarrow \mathbb{R}$ satisfies the Monge-Ampère equation Eq. 15.

For quadratic Euclidean distance cost $c(x, y) = 1/2\|x - y\|^2$ in \mathbb{R}^n , the c -transform and the classical Legendre transform have special relations.

Definition 5 (Legendre Transform). Given a function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$, its Legendre transform is defined as

$$\varphi^*(y) := \sup_x (\langle x, y \rangle - \varphi(x)). \quad (18)$$

We can show the following relation holds for quadratic Euclidean cost:

$$\frac{1}{2}|y|^2 - \varphi^c(y) = \left(\frac{1}{2}|x|^2 - \varphi(x) \right)^*. \quad (19)$$

McCann's Displacement

We consider all the probability measures μ defined on X with finite second order moment; μ is absolutely continuous with respect to Lebesgue measure:

$$\mathcal{P}(X) := \left\{ \mu : \int_X |x|^2 d\mu(x) < \infty, \mu \text{ a.c.} \right\} \tag{20}$$

Then according to Brenier’s theorem, for any pair $\mu, \nu \in \mathcal{P}(X)$, there exists a unique optimal transport map $T : X \rightarrow X, T\#\mu = \nu$; furthermore $T = \nabla u$ for some Brenier potential u , which satisfies the Monge-Ampère equation 15. The transportation cost gives the Wasserstein distance between μ and ν in Eq. 6.

Definition 6. Given a path $\rho : [0, 1] \rightarrow \mathcal{P}(X)$ in the $(\mathcal{P}(X), \mathcal{W}_2)$, if it satisfies the condition

$$\mathcal{W}_2(\rho(s), \rho(t)) = |t - s| \mathcal{W}_2(\rho(0), \rho(1)) \quad \forall s, t \in [0, 1], \tag{21}$$

then we say ρ is a geodesic.

McCann gives the geodesic formula in the distance space $(\mathcal{P}(X), \mathcal{W}_2)$.

Theorem 3 (McCann). Given $\mu, \nu \in (\mathcal{P}(X), \mathcal{W}_2)$ and u is the corresponding Brenier potential, then the geodesic connecting μ and ν is given by

$$\rho(t) := ((1 - t)Id + t\nabla u)\#\mu \quad t \in [0, 1],$$

which is called McCann’s displacement.

Benamou-Brenier Dynamic Fluid

Brenier-Benamou gives another formulation of geodesics using fluid dynamics. Let $X = \mathbb{R}^n$, and consider a flow field in X , represented by the density field $\rho(t, x)$ and the flow velocity field $\mathbf{v}(t, x)$. We denote $\rho(t, \cdot)$ as $\rho_t, \mathbf{v}(t, \cdot)$ as \mathbf{v}_t . We define $\Sigma(\mu, \nu)$ as set of flows $(\rho, \mathbf{v}) = (\rho_t, \mathbf{v}_t), 0 \leq t \leq 1$, satisfying the following conditions:

1. ρ_t is continuous with respect to t and $\rho_t(x)$ is absolutely continuous with respect to the Lebesgue measure in X .
2. $\mathbf{v}(t, x)$ is L^2 integrable with respect to the measure $d\rho_t(x)dt$.

$$\int_0^1 \int_X |\mathbf{v}(t, x)|^2 d\rho_t(x)dt < \infty.$$

3. The union of the support of ρ_t is bounded.

$$\bigcup_{0 \leq t \leq 1} \text{Supp}(\rho_t) \text{ bounded}$$

4. By mass conservation law, the pair (ρ, \mathbf{v}) satisfies the *continuity equation*:

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t \mathbf{v}_t) = 0 \quad (22)$$

in the distributional sense.

5. Furthermore, the flow satisfies the boundary condition $\rho_0 = \mu$ and $\rho_1 = \nu$.

Problem 5 (Benamou-Brenier). Find the flow $(\rho, \mathbf{v}) \in \Sigma(\mu, \nu)$ that minimizes the total kinetic energy:

$$A[\rho, \mathbf{v}] = \int_0^1 \left(\int_X \rho_t(x) |\mathbf{v}_t(x)|^2 dx \right) dt. \quad (23)$$

Benamou-Brenier proves that the kinetic energy of the solution to Eq. 23 equals to the square of the Wasserstein distance in Eq. 6, namely, Benamou-Brenier problem is equivalent to Brenier problem; furthermore the geodesic is given by the solution to the Benamou-Brenier problem:

$$\min \left\{ \frac{1}{2} \int_0^1 \int_X |\mathbf{v}(x, t)|^2 d\rho(x, t) dt : (\rho_t, \mathbf{v}_t) \in \Sigma(\mu, \nu) \right\}.$$

Otto's Calculus

Suppose \mathbf{v} is the optimal flow, given any divergence free field $\nabla \cdot \mathbf{w} = 0$,

$$-\nabla \cdot \rho \left(\mathbf{v} + \varepsilon \frac{\mathbf{w}}{\rho} \right) = -\nabla \cdot \rho \mathbf{v} = \frac{\partial \rho}{\partial t},$$

therefore $\mathbf{v} + \varepsilon \mathbf{w}/\rho \in \Sigma(\mu, \nu)$. By the optimality of \mathbf{v} , we have

$$\int \rho |\mathbf{v}|^2 \leq \int \rho \left| \mathbf{v} + \varepsilon \frac{\mathbf{w}}{\rho} \right|^2,$$

therefore we have

$$\int \langle \mathbf{v}, \mathbf{w} \rangle = 0.$$

Because \mathbf{w} is an arbitrary divergence free vector field, by Hodge decomposition theorem, we have \mathbf{v} is the gradient field of some function φ , $\mathbf{v} = \nabla \varphi$. Benamou-Brenier problem is reduced to

$$\mathcal{W}_2^2(\mu, \nu) = \min_{(\rho_t, u)} \left\{ \int_0^1 \int_X |\nabla u|^2 d\rho_t dt, \rho_0 = \mu, \rho_1 = \nu, -\nabla \cdot (\rho_t \nabla u) = \frac{\partial \rho_t}{\partial t} \right\}.$$

Given two geodesics $\rho_1(t), \rho_2(t) \subset \mathcal{P}(X), \rho_1(0) = \rho_2(0) = \rho$, their tangent vectors at $\rho \in \mathcal{P}(X)$ are

$$\frac{\partial \rho_1}{\partial t} = -\nabla \cdot (\rho_1 \nabla \varphi_1), \quad \frac{\partial \rho_2}{\partial t} = -\nabla \cdot (\rho_2 \nabla \varphi_2),$$

the Riemannian metric is defined as

$$\left\langle \frac{\partial \rho_1}{\partial t}, \frac{\partial \rho_2}{\partial t} \right\rangle_\rho = \int_X \langle \nabla \varphi_1, \nabla \varphi_2 \rangle \rho(x) dx.$$

Otto’s calculus provides a theoretic tool for optimization in $(\mathcal{P}(X), \mathcal{W}_2)$. For example, we can show the Wasserstein gradient flow of entropy is equivalent to the classical heat flow. Given a domain $X \subset \mathbb{R}^d$ with smooth boundary ∂X and a measure $\rho \in \mathcal{P}(X)$, its entropy is defined as

$$\text{Ent}(\rho) := \int_X \rho \log \rho \, dx.$$

Given a path $\rho(t) \subset \mathcal{P}(X)$,

$$\frac{d}{dt} \text{Ent}(\rho(t)) = \int_X \left(\dot{\rho} \log \rho + \rho \frac{\dot{\rho}}{\rho} \right) dx = \int_X (1 + \log \rho) \dot{\rho} \, dx.$$

By continuity equation $\dot{\rho} = -\nabla \cdot (\mathbf{v}\rho)$

$$\int_X \dot{\rho} \, dx = - \int_X \nabla \cdot (\mathbf{v}\rho) \, dx = - \int_{\partial X} \mathbf{v}\rho \, dx = 0.$$

and

$$\nabla \cdot (\rho \log \rho \mathbf{v}) = \log \rho \nabla(\rho \mathbf{v}) + \langle \nabla \log \rho, \rho \mathbf{v} \rangle.$$

we obtain

$$\frac{d}{dt} \text{Ent}(\rho(t)) = \int_X \langle \nabla \log \rho, \mathbf{v} \rangle \rho \, dx$$

This shows the Wasserstein gradient of entropy equals to $\nabla \log \rho$. We plug it into the continuity equation and obtain

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot \left(-\frac{\nabla \rho_t}{\rho_t} \rho_t \right) = \frac{\partial \rho_t}{\partial t} - \Delta \rho_t = 0.$$

This shows that the Wasserstein gradient flow of the entropy is equivalent to the classical heat flow.

Regularity of Optimal Transport Maps

Let Ω and Λ be two bounded smooth open sets in \mathbb{R}^d , and let $d\mu = f dx$ and $d\nu = g dy$ be two probability measures on \mathbb{R}^d such that $f|_{\mathbb{R}^d \setminus \Omega} = 0$ and $g|_{\mathbb{R}^d \setminus \Lambda} = 0$. Assume that f and g are bounded away from zero and infinity on Ω and Λ , respectively.

Convex Target Domain

Definition 7 (Hölder continuous). A real or complex-valued function f on d -dimensional Euclidean space satisfies a Hölder condition, or is Hölder continuous, when there are non-negative real constants $C, \alpha > 1$, such that

$$|f(x) - f(y)| \leq C|x - y|^\alpha$$

for all x and y in the domain of f .

Definition 8 (Hölder Space). The Hölder space $C^{k,\alpha}(\Omega)$, where Ω is an open subset of some Euclidean space and $k \geq 0$ an integer, consists of those functions on Ω having continuous derivatives up to order k and such that the k -th partial derivatives are Hölder continuous with exponent α , where $0 < \alpha \leq 1$.

Consider the optimal transport map $\nabla u : (\Omega, f(x)dx) \rightarrow (\Lambda, g(y)dy)$, the following theorems give the regularity of the Brenier potential u . Caffarelli’s theorem addresses the cases with the cost function $c(x, y) = 1/2|x - y|^2$.

Theorem 4 (Caffarelli 1991). *If Λ is convex, then the Brenier potential u is strictly convex; furthermore*

1. *If $\lambda \leq f, g \leq 1/\lambda$ for some $\lambda > 0$, then $u \in C_{loc}^{1,\alpha}(\Omega)$.*
2. *If $f \in C_{loc}^{k,\alpha}(\Omega)$ and $g \in C_{loc}^{k,\alpha}(\Lambda)$, with $f, g > 0$, then $u \in C_{loc}^{k+2,\alpha}(\Omega)$, ($k \geq 0, \alpha \in (0, 1)$)*

Ma-Trudinger-Wang’s theorem (Ma et al. 2005) handles general cost functions $c(x, y)$. In the following theorem,

$$c_{p,q} := \frac{\partial^2 c(x, y)}{\partial x_p \partial y_q}, c_{ij,p} := \frac{\partial^3 c(x, y)}{\partial x_i \partial x_j \partial y_p}, c_{ij,pq} := \frac{\partial^4 c(x, y)}{\partial x_i \partial x_j \partial y_p \partial y_q},$$

and $(c^{p,q})$ is the inverse matrix of $c_{p,q}$.

Theorem 5 (Ma-Trudinger-Wang). *The potential function u is C^3 smooth if the cost function c is smooth, f, g are positive, $f \in C^2(\Omega), g \in C^2(\Lambda)$, and*

- A1 $\forall x, \xi \in \mathbb{R}^n, \exists !y \in \mathbb{R}^n, \text{ s.t. } \xi = D_x c(x, y)$ (for existence)
- A2 $|D_{xy}^2 c| \neq 0$.
- A3 $\exists c_0 > 0 \text{ s.t. } \forall \xi, \eta \in \mathbb{R}^n, \xi \perp \eta$

$$\sum (c_{ij,rs} - c^{p,q} c_{ij,p} c_{q,rs}) c^{r,k} c^{s,l} \xi_i \xi_j \eta_k \eta_l \geq c_0 |\xi|^2 |\eta|^2.$$

- B1 Λ is c -convex w.r.t. Ω , namely, $\forall x_0 \in \Omega$,

$$\Lambda_{x_0} := D_x c(x_0, \Lambda)$$

is convex.

Non-convex Target Domain

If Λ is not convex, there exist smooth f and g such that $u \notin C^1(\Omega)$, and the optimal transportation map ∇u is discontinuous at singularities.

Definition 9 (subgradient). Given an open set $\Omega \subset \mathbb{R}^d$ and $u : \Omega \rightarrow \mathbb{R}$ a convex function, for $x \in \Omega$, the subgradient (subdifferential) of u at x is defined as

$$\partial u(x) := \{p \in \mathbb{R}^n : u(z) \geq u(x) + \langle p, z - x \rangle \quad \forall z \in \Omega\}.$$

It is obvious that $\partial u(x)$ is a closed convex set. Geometrically, if $p \in u(x)$, then the hyper-plane

$$l_{x,p}(z) := u(x) + \langle p, z - x \rangle$$

touches u from below at x , namely, $l_{x,p} \leq u$ in Ω and $l_{x,p}(x) = u(x)$, $l_{x,p}$ is a supporting plane to u at x .

The Brenier potential u is differentiable at x if its subgradient $\partial u(x)$ is a singleton. We classify the points according to the dimensions of their subgradients and define the sets

$$\Sigma_k(u) := \left\{ x \in \mathbb{R}^d \mid \dim(\partial u(x)) = k \right\}, \quad k = 0, 1, 2, \dots, d.$$

It is obvious that $\Sigma_0(u)$ is the set of regular points, $\Sigma_k(u), k > 0$ are the set of singular points. We also define the *reachable subgradients* at x as

$$\nabla_* u(x) := \left\{ \lim_{k \rightarrow \infty} \nabla u(x_k) \mid x_k \in \Sigma_0, x_k \rightarrow x \right\}.$$

It is well known that the subgradient equals to the convex hull of the reachable subgradient

$$\partial u(x) = \text{Convex Hull}(\nabla_* u(x)).$$

Theorem 6 (Regularity Figalli (2010)). *Let $\Omega, \Lambda \subset \mathbb{R}^d$ be two bounded open sets, and let $f, g : \mathbb{R}^d \rightarrow \mathbb{R}^+$ be two probability densities, which are zero outside Ω, Λ and are bounded away from zero and infinity on Ω, Λ , respectively. Denote by $T = \nabla u : \Omega \rightarrow \Lambda$ the optimal transport map provided by Theorem 2. Then there exist two relatively closed sets $\Sigma_\Omega \subset \Omega$ and $\Sigma_\Lambda \subset \Lambda$ with $|\Sigma_\Omega| = |\Sigma_\Lambda| = 0$ such that $T : \Omega \setminus \Sigma_\Omega \rightarrow \Lambda \setminus \Sigma_\Lambda$ is a homeomorphism of class $C_{loc}^{0,\alpha}$ for some $\alpha > 0$.*

We call Σ_Ω as singular set of the optimal transportation map $\nabla u : \Omega \rightarrow \Lambda$. Figure 1 illustrates the singularity set structure, computed using the algorithm based on Theorem 8. We obtain

$$\Sigma_\Omega = \Omega \setminus \{\Sigma_1 \cup \Sigma_2\}, \quad \Sigma_1 = \bigcup_{k=0}^3 \gamma_k, \quad \Sigma_2 = \{x_0, x_1\}.$$

The subgradient of x_0 , $\partial u(x_0)$ is the entire inner hole of Λ , $\partial u(x_1)$ which is the shaded triangle. For each point on $\gamma_k(t)$, $\partial u(\gamma_k(t))$ is a line segment outside Λ . x_1 is the bifurcation point of γ_1, γ_2 , and γ_3 . The Brenier potential on Σ_1 and Σ_2 is not differentiable, and the optimal transportation map ∇u on them is discontinuous.

Figure 2 shows the singularity structure of an optimal transport map between the uniform distribution inside a solid ball to that of the solid Stanford bunny. Since the target domain is non-convex, the boundary surface has complicated folding structure, which is the singularity set of the map.

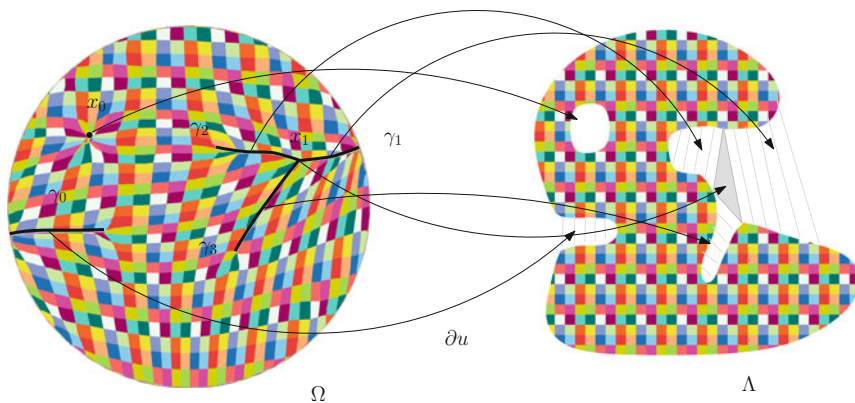


Fig. 1 Singularity structure of an optimal transportation map

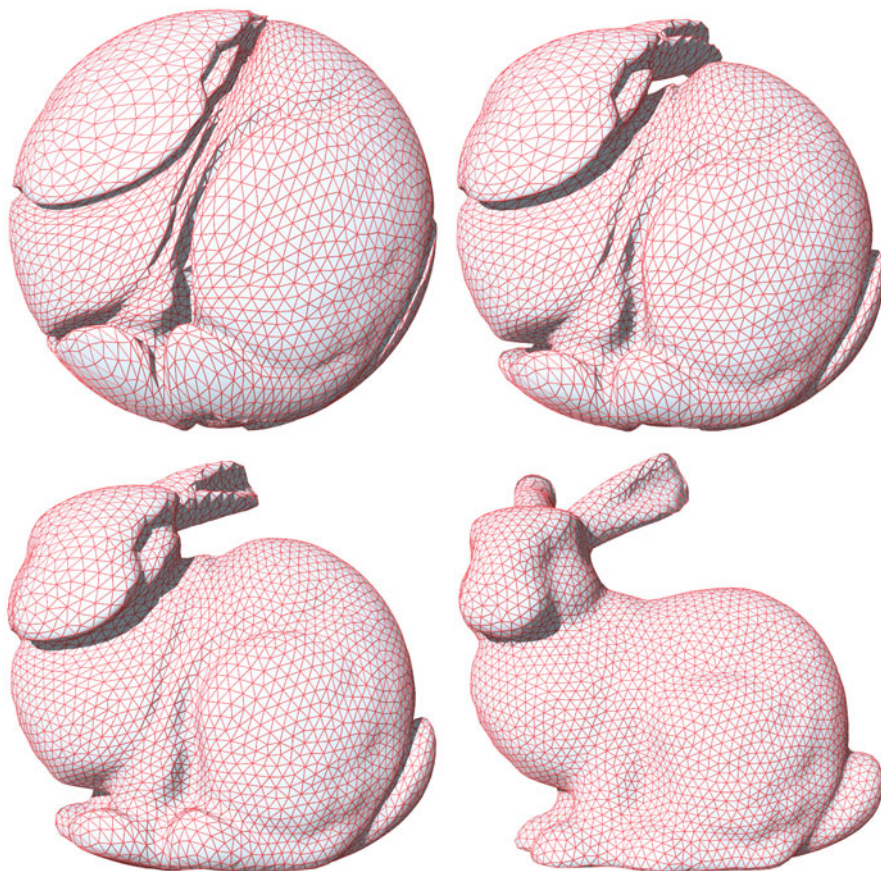


Fig. 2 Singularity structure of an optimal transport map

Computational Algorithm

Semi-discrete Optimal Transport Map

Brenier's theorem can be directly generalized to the discrete situation. The source measure μ is absolutely continuous with respect to Lebesgue measure, defined on a convex compact domain Ω ; the target measure ν is the summation of Dirac measures

$$\nu = \sum_{i=1}^n v_i \delta(y - y_i), \quad (24)$$

where $Y = \{y_1, y_2, \dots, y_n\}$ are training samples. The source and the target measures have equal total mass $\sum_{i=1}^n v_i = \mu(\Omega)$. Each sample y_i corresponds to a supporting plane of the Brenier potential, denoted as

$$\pi_{h,i}(x) := \langle x, y_i \rangle + h_i, \tag{25}$$

where the height h_i is an unknown variable. We represent all the height variables as $\mathbf{h} = (h_1, h_2, \dots, h_n)$.

An *envelope* of a family of hyper-planes in the Euclidean space is a hyper-surface that is tangent to each member of the family at some point, and these points of tangency together form the whole envelope. As shown in Fig. 3, the Brenier potential $u_{\mathbf{h}} : \Omega \rightarrow \mathbb{R}$ is a piecewise linear convex function determined by h , which is the upper envelope of all its supporting planes,

$$u_{\mathbf{h}}(x) = \max_{i=1}^n \{\pi_{h,i}(x)\} = \max_{i=1}^n \{\langle x, y_i \rangle + h_i\}. \tag{26}$$

The graph of Brenier potential is a convex polytope. Each supporting plane $\pi_{h,i}$ corresponds to a facet of the polytope. The projection of the polytope induces a cell decomposition of Ω , each supporting plane $\pi_i(x)$ projects onto a cell $W_i(\mathbf{h})$,

$$\Omega = \bigcup_{i=1}^n W_i(\mathbf{h}) \cap \Omega, \quad W_i(\mathbf{h}) := \{p \in \mathbb{R}^d \mid \nabla u_{\mathbf{h}}(p) = y_i\}. \tag{27}$$

the cell decomposition is a *power diagram*.

The μ -measure of $W_i \cap \Omega$ is denoted as $w_i(\mathbf{h})$,

$$w_i(\mathbf{h}) := \mu(W_i(\mathbf{h}) \cap \Omega) = \int_{W_i(\mathbf{h}) \cap \Omega} d\mu. \tag{28}$$

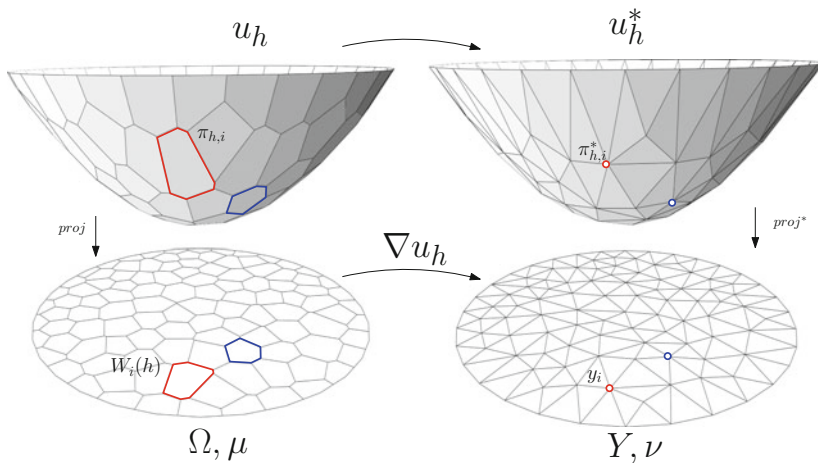


Fig. 3 PL Brenier potential (left) and its Legendre dual (right)

The gradient map $\nabla u_{\mathbf{h}} : \Omega \rightarrow Y$ maps each cell $W_i(\mathbf{h})$ to a single point y_i ,

$$\nabla u_{\mathbf{h}} : W_i(\mathbf{h}) \mapsto y_i, i = 1, 2, \dots, n. \tag{29}$$

Given the target measure ν in Eq. 24, there exists a discrete Brenier potential in Eq. 26, whose projected μ -volume of each facet $w_i(h)$ equals to the given target measure ν_i . This was proved by Alexandrov in convex geometry.

Theorem 7 (Alexandrov2005). *Suppose Ω is a compact convex polytope with non-empty interior in \mathbb{R}^n , $n_1, \dots, n_k \subset \mathbb{R}^{n+1}$ are distinct k unit vectors, the $(n+1)$ -th coordinates are negative, and $v_1, \dots, v_k > 0$ so that $\sum_{i=1}^k v_i = \text{vol}(\Omega)$. Then there exists a convex polytope $P \subset \mathbb{R}^{n+1}$ with exact k codimension-1 faces F_1, \dots, F_k so that n_i is the normal vector to F_i and the intersection between Ω and the projection of F_i is with volume v_i . Furthermore, such P is unique up to vertical translation.*

Alexandrov’s proof for the existence is based on algebraic topology, which is not constructive. Recently, Gu et al. (2016) gave a constructive proof based on the variational approach.

Theorem 8 (Gu-Luo-Yau 2016). *Let μ be a probability measure defined on a compact convex domain Ω in \mathbb{R}^d , $Y = \{y_1, y_2, \dots, y_n\}$ be a set of distinct points in \mathbb{R}^d . Then for any $v_1, v_2, \dots, v_n > 0$ with $\sum_{i=1}^n v_i = \mu(\Omega)$, there exists $\mathbf{h} = (h_1, h_2, \dots, h_n) \in \mathbb{R}^n$, unique up to adding a constant (c, c, \dots, c) , so that $w_i(h) = v_i$, for all i . The vector h is the unique minimum argument of the following convex energy*

$$E(\mathbf{h}) = \int_0^{\mathbf{h}} \sum_{i=1}^n w_i(\eta) d\eta_i - \sum_{i=1}^n h_i v_i, \tag{30}$$

defined on an open convex set

$$\mathcal{H} = \{\mathbf{h} \in \mathbb{R}^n : w_i(h) > 0, i = 1, 2, \dots, n\}. \tag{31}$$

Furthermore, $\nabla u_{\mathbf{h}}$ minimizes the quadratic cost

$$\frac{1}{2} \int_{\Omega} |x - T(x)|^2 d\mu(x) \tag{32}$$

among all transport maps $T_{\#}\mu = \nu$, where the Dirac measure $\nu = \sum_{i=1}^n v_i \delta(y - y_i)$.

The gradient of the above convex energy in Eq. 30 is given by

$$\nabla E(\mathbf{h}) = (w_1(\mathbf{h}) - v_1, w_2(\mathbf{h}) - v_2, \dots, w_n(\mathbf{h}) - v_n)^T \tag{33}$$

The Hessian of the energy is given by

$$\frac{\partial w_i}{\partial h_j} = -\frac{\mu(W_i \cap W_j \cap \Omega)}{|y_i - y_j|}, \quad \frac{\partial w_i}{\partial h_i} = \sum_{j \neq i} \frac{\partial w_i}{\partial h_j} \tag{34}$$

As shown in Fig. 3, the Hessian matrix has explicit geometric interpretation. The left frame shows the discrete Brenier potential u_h ; the right frame shows its Legendre transformation u_h^* using Definition 18. The Legendre transformation can be constructed geometrically: for each supporting plane $\pi_{h,i}$, we construct the dual point $\pi_{h,i}^* = (y_i, -h_i)$; the convex hull of the dual points $\{\pi_{h,1}^*, \pi_{h,2}^*, \dots, \pi_{h,n}^*\}$ is the graph of the Legendre transformation u_h^* . The projection of $Y = \{y_1, y_2, \dots, y_n\}$, which is the *weighted Delaunay triangulation*. As shown in Fig. 4, the power diagram in Eq. 27 and weighted Delaunay triangulation are Poincaré dual to each other: if in the power diagram, $W_i(h)$ and $W_j(h)$ intersect at a $(d - 1)$ -dimensional cell, then in the weighted Delaunay triangulation y_i connects with y_j . The element of the Hessian matrix Eq. 34 is the ratio between the μ -volume of the $(d - 1)$ cell in the power diagram and the length of dual edge in the weighted Delaunay triangulation.

The conventional power diagram can be closely related to the above theorem.

Definition 10. (power distance) Given a point $y_i \in \mathbb{R}^d$ with a power weight ψ_i , the power distance is given by

$$\text{pow}(x, y_i) = |x - y_i|^2 - \psi_i. \tag{35}$$

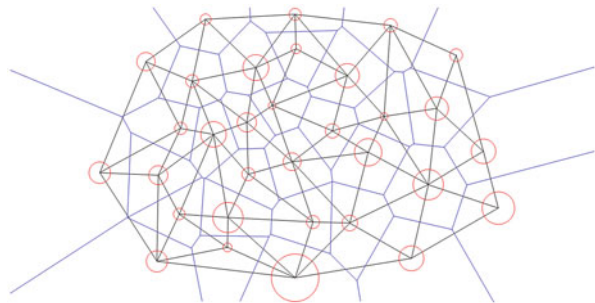
Definition 11. (power diagram) Given weighted points $(y_1, \psi_1), \dots, (y_k, \psi_k)$, the power diagram is the cell decomposition of \mathbb{R}^d

$$\mathbb{R}^d = \cup_{i=1}^k W_i(\psi), \tag{36}$$

where each cell is a convex polytope

$$W_i(\psi) = \{x \in \mathbb{R}^d \mid \text{pow}(x, y_i) \leq \text{pow}(x, y_j), 1 \leq j \leq k\}. \tag{37}$$

Fig. 4 Power diagram (blue) and its dual weighted Delaunay triangulation (black)



The weighted Delaunay triangulation, denoted as $\mathcal{T}(\psi)$, is the Poincaré dual to the power diagram; if $W_i(\psi) \cap W_j(\psi) \neq \emptyset$, then there is an edge connecting y_i and y_j in the weighted Delaunay triangulation. Note that $pow(x, y_i) \leq pow(x, y_j)$ is equivalent to

$$\langle x, y_i \rangle + \frac{1}{2}(\psi_i - |y_i|^2) \geq \langle x, y_j \rangle + \frac{1}{2}(\psi_j - |y_j|^2). \tag{38}$$

Let $h_i = 1/2(\psi_i - |y_i|^2)$ then we re-write definition of $W_i(\psi)$ as

$$W_i(\psi) = \{x \in \mathbb{R}^d \mid \langle x, y_i \rangle + h_i \geq \langle x, y_j \rangle + h_j, \forall j\}. \tag{39}$$

Damping Newton’s Method

Initially, we set $\mathbf{h}^0 = \frac{1}{2}(|y_1|^2, |y_2|^2, \dots, |y_n|^2)$, where y_i represents the coordinates of the i -th sample in the target domain. The initial power diagram and weighted Delaunay triangulation are conventional Voronoi diagram and Delaunay triangulation. This guarantees the initial Brenier potential and its Legendre dual are strictly convex, namely, the initial height vector belongs to the admissible space, $\mathbf{h}^0 \in \mathcal{H}$.

Assume at the k -th step, we have got \mathbf{h}^k , the Brenier potential $u_{\mathbf{h}^k}$, and its Legendre dual $u_{\mathbf{h}^k}^*$, the power diagram $\{W_{\mathbf{h}^k}^i\}_{i=1}^n$. We compute the gradient of Alexandrov energy Eq. (33) and Hessian matrix H as described in Eq. (34). Then we solve the linear system:

$$\nabla E(\mathbf{h}^k) = \text{Hess}(\mathbf{h}^k)\mathbf{d}.$$

Next, we need to determine the step length λ . We initialize λ as one and compute the convex hull of the points

$$\{(y_1, h_1^k + \lambda d_1), (y_2, h_2^k + \lambda d_2), \dots, (y_n, h_n^k + \lambda d_n)\}.$$

If the convex hull misses any point, then $\mathbf{h}^k + \lambda \mathbf{d}$ is outside the admissible space, and the corresponding Brenier potential is not strictly convex. Then we reduce the step length λ by half, $\lambda \leftarrow \frac{1}{2}\lambda$, and repeat the trial. We repeat this procedure and find the minimal l , such that

$$\min_l \mathbf{h}^k + 2^{-l}\mathbf{d} \in \Sigma.$$

By iterating this procedure, we reduce the Alexandrov energy monotonously, until the gradient of the energy is less than a prescribed threshold $\varepsilon > 0$.

Algorithm 1 Geometric Variational method for optimal transportation map

- 1: **Input:** Convex domain Ω with measure μ ; Discrete samples $Y := \{y_1, y_2, \dots, y_n\}$ with measures $\nu_1, \nu_2, \dots, \nu_n$, respectively μ and ν are with equal measures $\mu(\Omega) = \sum_{i=1}^n \nu_i$.
 - 2: **Output:** Optimal transport map $T : \Omega \rightarrow Y$.
 - 3: Initialize $\mathbf{h}^0 = (h_1, h_2, \dots, h_n) \leftarrow 1/2(|y_1|^2, |y_2|^2, \dots, |y_n|^2)$.
 - 4: **while** true **do**
 - 5: Compute the Brenier potential $u_{\mathbf{h}^k}$ and its Legendre dual $u_{\mathbf{h}^k}^*$;
 - 6: Project $u_{\mathbf{h}^k}$ and $u_{\mathbf{h}^k}^*$ to obtain the power diagram and weighted Delaunay triangulation;
 - 7: Compute the gradient $\nabla E(\mathbf{h}^k)$ of Alexandrov energy Eq. (33);
 - 8: **if** $\|\nabla E(\mathbf{h}^k)\|$ is less than ε **then**
 - 9: return $T = \nabla u_{\mathbf{h}^k}$.
 - 10: **end if**
 - 11: Compute the Hessian matrix of Alexandrov energy Eq. (34) and (30);
 - 12: Solve linear system $\nabla E(\mathbf{h}^k) = \text{Hess}(\mathbf{h}^k)\mathbf{d}$;
 - 13: Set the step length $\lambda \leftarrow 1$;
 - 14: **repeat**
 - 15: $\lambda \leftarrow \lambda/2$;
 - 16: Construct the convex hull of $\{(y_i, h_i^k + \lambda d_i)\}_{i=1}^n$;
 - 17: **until** all sample points are on the convex hull;
 - 18: update height vector $\mathbf{h}^{k+1} \leftarrow \mathbf{h}^k + \lambda \mathbf{d}$;
 - 19: **end while**
-

As shown in Fig. 5, given a genus zero surface S with a single boundary, it has an induced Euclidean metric \mathbf{g} , which induces the surface area element $dA_{\mathbf{g}}$. After the normalization, the total surface area is π . The Riemann mapping $\varphi : (S, \mathbf{g}) \rightarrow (\mathbb{D}, du^2 + dv^2)$ maps the surface onto the unit disk and pushes the area element to the disk, denoted as $\varphi_{\#}dA_{\mathbf{g}}$. Since Riemann mapping is conformal, the surface area element can be written as

$$dA_{\mathbf{g}}(u, v) = e^{2\lambda(u, v)} dudv,$$

where $e^{2\lambda(u, v)}$ is the area distortion function and can be treated as the target density function.

On the disk, the Lebesgue measure, or equivalently the Euclidean metric $du^2 + dv^2$, induces the Euclidean area element $dudv$. We compute the optimal transportation $T : (\mathbb{D}, dudv) \rightarrow (\mathbb{D}, \varphi_{\#}dA_{\mathbf{g}})$ using the geometric variational method. The optimal transport mapping result is shown between the two planar images. The composition between the Riemann mapping φ and the inverse of the optimal transport map T^{-1} gives an area-preserving mapping

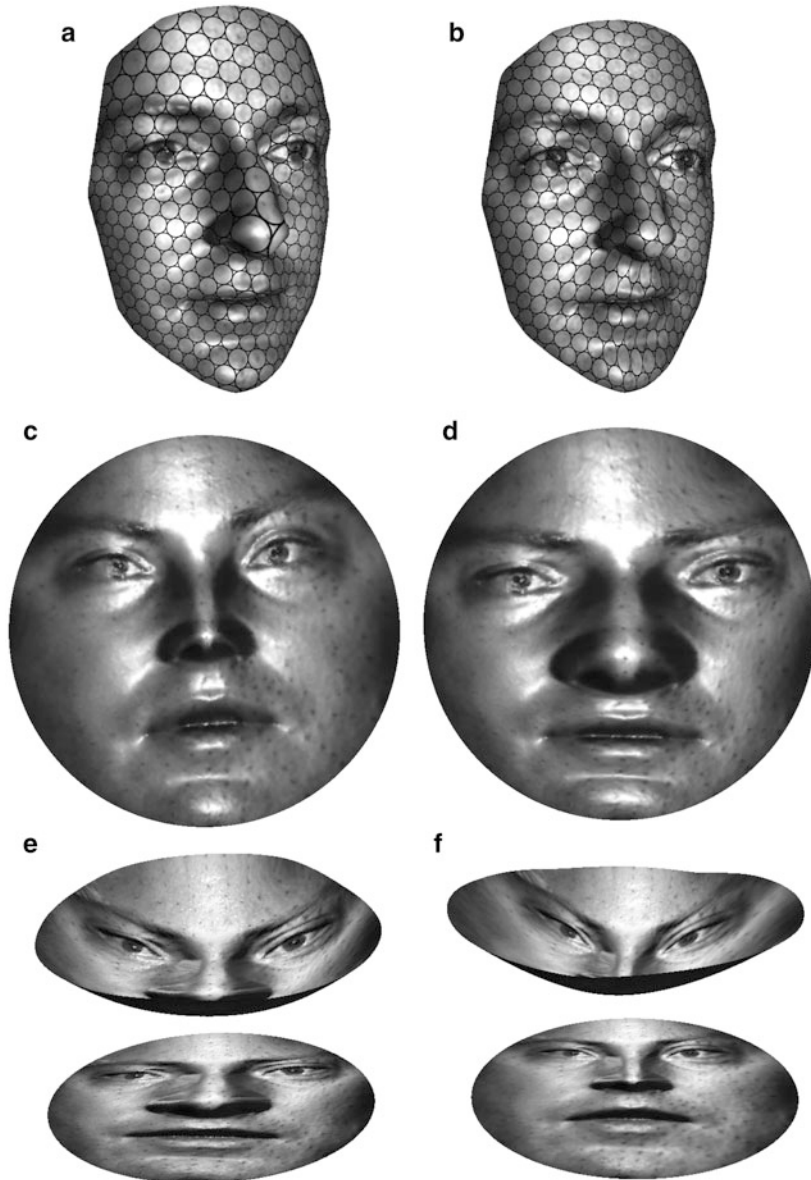


Fig. 5 The optimal transport map for a male face. (a) conformal parameterization (b) area-preserving parameterization (c) conformal mapping (d) optimal transport map (e) Brenier potential (f) Legendre dual

$$T^{-1} \circ \varphi : (S, \mathbf{g}) \rightarrow (\mathbb{D}, dudv), \quad (T^{-1} \circ \varphi)_\# dA_{\mathbf{g}} = dudv.$$

In order to visualize the mapping $T^{-1} \circ \varphi$ is area-preserving, we put circle packing texture on the planar unit disk and pull it back to the original surface as shown in the top right frame Fig. 5, we can see that the small circles are mapped to ellipses with similar areas.

As shown in Fig. 6, we compute the histograms to measure the distortions. The top row shows the histograms of conformal mapping of Fig. 5, and the bottom row shows those of optimal transport map. The left column shows the angle distortion histogram and the right column the area distortion histogram. The angle distortion histogram is calculated as follows: the triangle mesh S in \mathbb{R}^3 and its planar image share the same triangulation; each corner angle in S corresponds to a planar corner angle. We compute the logarithm of the ratio between the corresponding corner angles and construct the histograms. By Fig. 6 left column, it is obvious that the angle distortion histogram of conformal mapping highly concentrates on the zero point; this shows the conformal mapping induces very small angle distortions; in contrast, the optimal transport map induces large angle distortions. The right column shows the area distortion histograms, which are obtained by computing the logarithm of the ratios between corresponding face areas. It can be seen that the optimal transport map induces very small area distortions, whereas the conformal mapping induces large area distortions (Fig. 6).

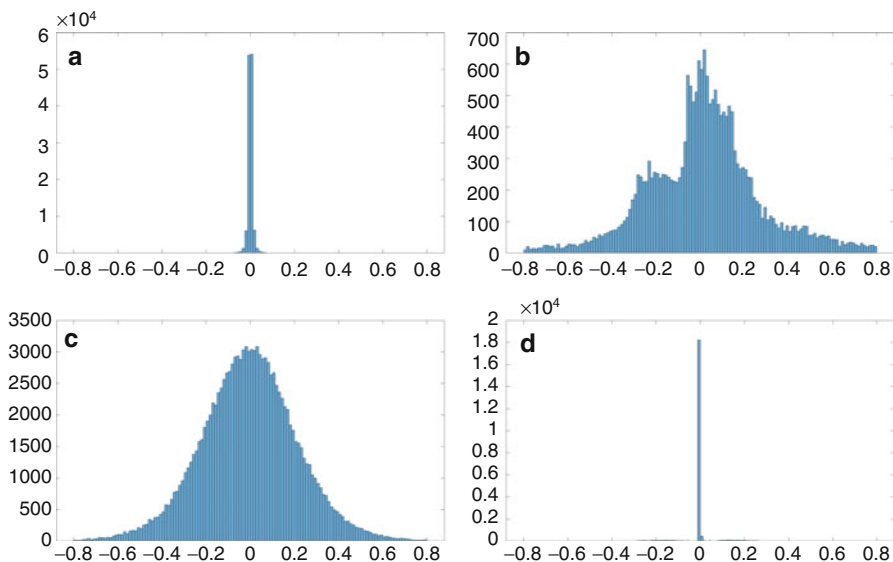


Fig. 6 Angle distortion and area distortion histograms of the male surface in Fig. 5. (a) angle distortion of conformal mapping (b) area distortion of conformal mapping (c) angle distortion of optimal transport map (d) area distortion of optimal, transport map

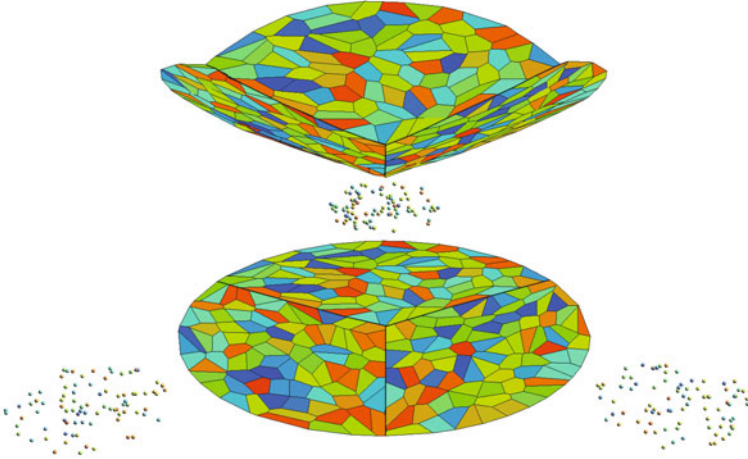


Fig. 7 Singularity set of the Brenier potential function is the discontinuity set of the optimal transportation map

Figure 8 shows the computation process of the Buddha surface model. The conformal mapping is computed first, and then the optimal transport map is obtained by finding the Brenier potential. The intermediate maps are shown in the figure.

Monte-Carlo Method

In practice, our goal is to compute the discrete Brenier potential in Eq. (26) by optimizing the convex energy in Eq. (30). For low dimensional cases, we can directly use Newton's method by computing the gradient Eq. (33) and Hessian matrix Eq. (34). For deep learning applications, direct computation of Hessian matrix is unfeasible; instead we can use gradient descend method or quasi-Newton's method with super-linear convergence. The key of the gradient is to estimate the μ -volume $w_i(\mathbf{h})$. This can be done use Monte-Carlo method: we draw n random samples from the distribution μ and count the number of samples falling in $W_i(\mathbf{h})$, the ratio converge to the μ -volume. This method is purely parallel and can be implemented using GPU. Furthermore, we can use hierarchical method to further improve the efficiency: first we partition the target samples to clusters and compute the optimal transportation map to the mass centers of the clusters; second, for each cluster, we compute the OT map from the corresponding cell to the original target samples within the cluster.

In order to avoid mode collapse, we need to find the singularity sets in Ω . As shown in Fig. 7, the target Dirac measure has two clusters; the source is the uniform distribution on the unit planar disk. The graph of the Brenier potential function is a convex polyhedron with a ridge in the middle. The projection of the ridge on the disk is the singularity set $\Sigma_1(u)$; the optimal mapping is discontinuous on Σ_1 .

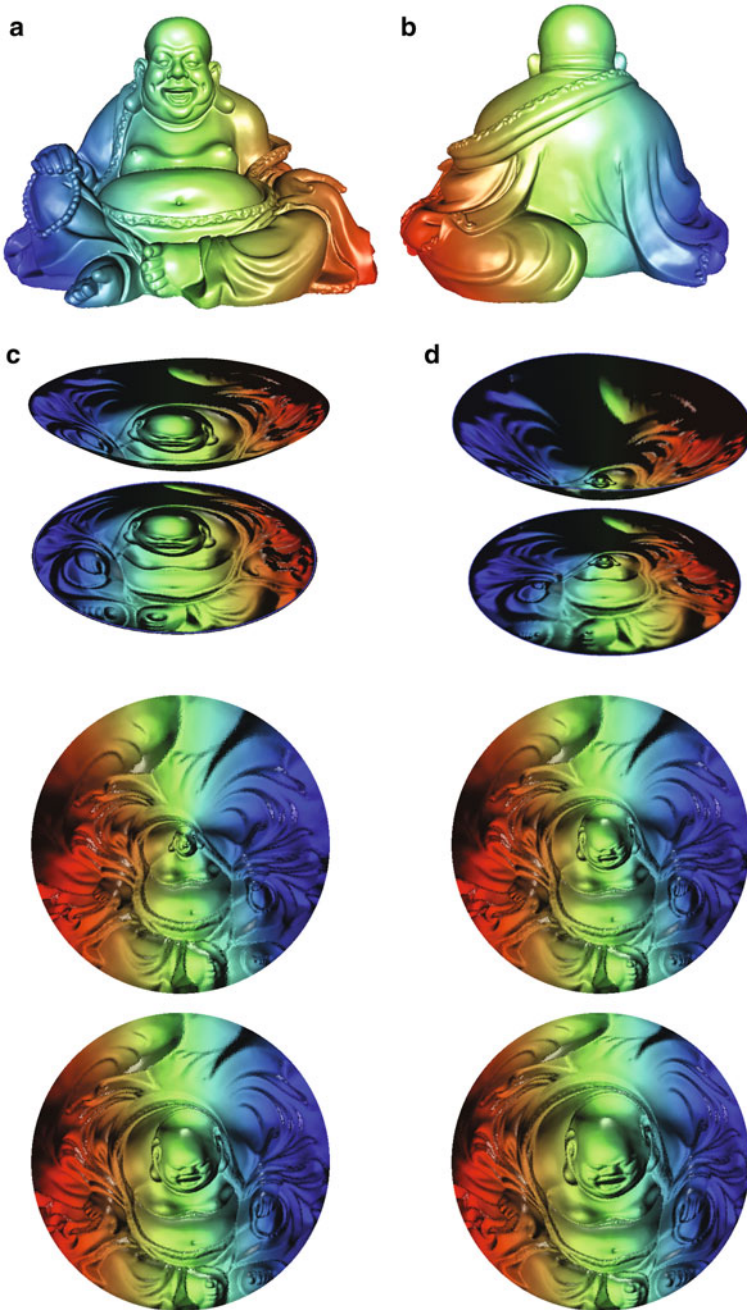


Fig. 8 Buddha surface, the last two rows show the intermediate computational results during the optimization. (a) Buddha surface front side (b) Buddha surface back side (c) Brenier potential (d) Legendre dual

In general cases, if two cells $W_i(\mathbf{h})$ and $W_j(\mathbf{h})$ are adjacent, then we compute the angle between the normals to the corresponding support planes:

$$\theta_{ij} := \frac{\langle y_i, y_j \rangle}{|y_i| \cdot |y_j|}$$

if θ_{ij} is greater than a threshold, then the common facet $W_i(h) \cap W_j(h)$ is in the discontinuity singular set.

Manifold Distribution Principle

We believe the great success of deep learning can be partially explained by the well accepted manifold distribution principle.

Manifold Distribution Principle

A natural class of data can be treated as a probability distribution on a low-dimensional manifold (data manifold) embedded in the high-dimensional ambient space (image space).

Furthermore, the distances among the probability distributions of subclasses on the manifold are far enough to distinguish them.

As shown in Fig. 9, the MNIST data set is a collection of handwritten images. Each image is 28×28 , which can be treated as a single point in the image space $\mathbb{R}^{28 \times 28}$, the MNIST data set is treated as a point cloud. Using Hinton's t-SNE embedding method, we can map the point cloud onto a planar domain, such that each image is mapped to a single point; the mapping is bijective. The images of the same digit are mapped to the same cluster. As shown in the right frame, there are ten clusters on the plane, corresponding to the ten handwritten digits. This shows the MNIST point cloud is close to a two-dimensional surface embedded in the 784 dimensional image space. We recall the concept of manifold Fig. 10:

Definition 12 (Manifold). Suppose M is a topological space, covered by a set of open sets $M \subset \bigcup_{\alpha} U_{\alpha}$. For each open set U_{α} , there is a homeomorphism $\varphi_{\alpha} : U_{\alpha} \rightarrow \mathbb{R}^n$; the pair $(U_{\alpha}, \varphi_{\alpha})$ form a chart. The union of charts form an atlas $\mathcal{A} = \{(U_{\alpha}, \varphi_{\alpha})\}$. If $U_{\alpha} \cap U_{\beta} \neq \emptyset$, then the chart transition map is given by $\varphi_{\alpha\beta} : \varphi_{\alpha}(U_{\alpha} \cap U_{\beta}) \rightarrow \varphi_{\beta}(U_{\alpha} \cap U_{\beta})$,

$$\varphi_{\alpha\beta} := \varphi_{\beta} \circ \varphi_{\alpha}^{-1}.$$

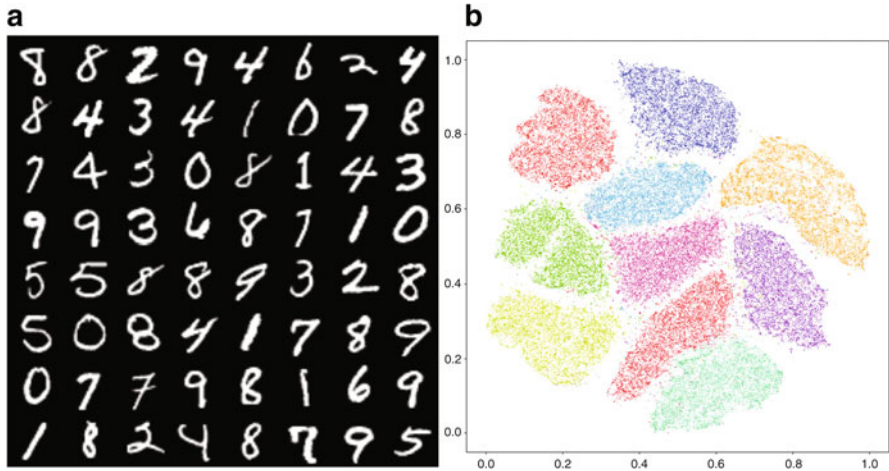


Fig. 9 The MNIST data set is a two dimensional surface in the image space. **(a)** LeCunn’s MNIST handwritten digits samples on manifold **(b)** Hinton’s t-SNE embedding on the latent space

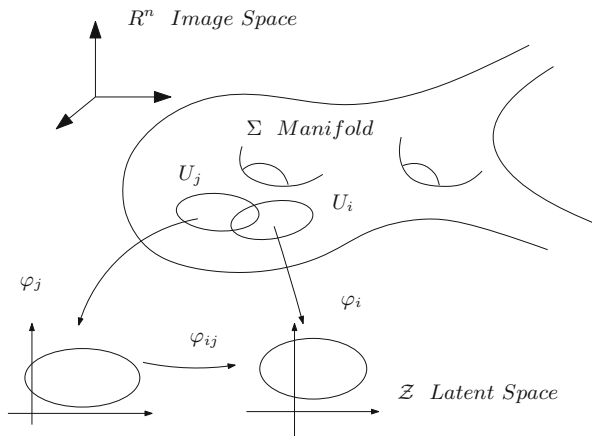
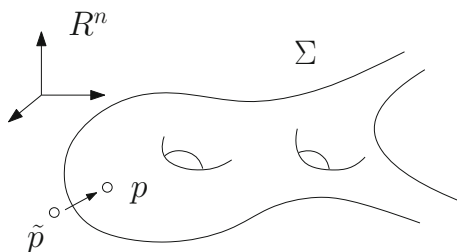


Fig. 10 The concept of manifold

The MNIST data set is treated as the *data manifold* Σ ; the space of all possible images is the *image space* \mathbb{R}^{784} ; the plane is the *latent space* \mathcal{Z} ; the mapping from the data manifold to the latent space $\varphi : \Sigma \rightarrow \mathcal{Z}$ is called the *encoding map*; the inverse mapping $\varphi^{-1} : \mathcal{Z} \rightarrow \Sigma$ is called the *decoding map*. Each handwritten digit image $p \in \Sigma$ is a *training sample* on the data manifold; its image of the encoding map $\varphi(p)$ is called the *latent code* of p . The data set can be treated as a probability distribution μ defined on the data manifold Σ , which is called *data distribution* (Fig. 10).

Fig. 11 Image denoising as projecting to a manifold



Main Tasks In general, deep learning systems have two major tasks:

1. Learn the manifold structure Σ , represented as encoding and decoding maps.
2. Learn the data distribution μ on Σ .

We use the manifold view to explain how the denoising is accomplished by a deep learning system. Traditional methods Fourier transform the noisy image, filter out the high frequency component, and inverse Fourier transform back to the denoised image. Deep learning methods use the clean images to train the neural network, obtain a representation of the manifold, and then project the noisy image to the manifold; the projection image point is the denoised image. As shown in Fig. 11 and the left frame of Fig. 12, we use a deep learning system to learn the data manifold Σ of clean human facial images. A facial image with noise is \tilde{p} , which is not on Σ but close to the manifold. We project \tilde{p} to Σ using the Riemannian metric in the image space \mathbb{R}^n , the closest point on Σ to \tilde{p} is p , and then p is the denoised image.

Traditional method is independent of the content of the image; ML method heavily depends on the content of the image. The prior knowledge is encoded by the manifold. If the wrong manifold is chosen, then the denoising result is of nonsense. As shown in Fig. 12 right frame, we use the cat face manifold to denoise a human face image; the result looks like a cat face.

Manifold Learning

Learning the data manifold structure is equivalent to learning the encoding and decoding maps. The encoding mapping $\varphi : \Sigma \rightarrow \mathcal{Z}$ maps the data manifold to the latent space. It push-forwards μ to the *latent distribution*, denoted as $\varphi_{\#}\mu$. Given the data manifold Σ and the latent space \mathcal{Z} , there are infinite many encoding mappings. In practice, it is crucial to choose the appropriate mapping that preserves the data distribution. We use a low-dimensional example to illustrate the concepts as shown in Fig. 13. The Buddha surface represents the data manifold Σ ; μ is the

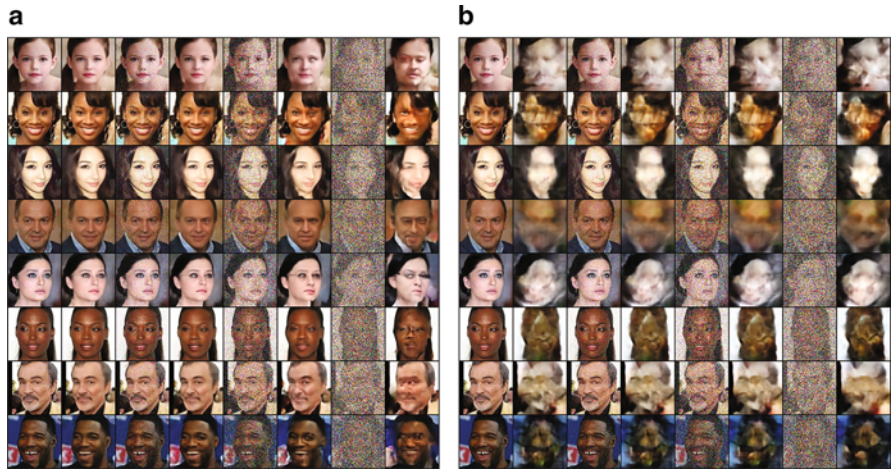


Fig. 12 Human facial image denoising by projection to the data manifold. (a) projection to a human facial photo manifold (b) projection to a cat face image manifold



Fig. 13 Different encoding mappings from the manifold to the planar disk

uniform distribution on Σ . Each row shows one encoding map. In the top row, if we uniformly sample the unit disk in the latent space, the samples are pulled back to the surface by the decoding map, and then the pullback samples on Σ are highly nonuniform. In contrast, in the bottom row, the uniform latent samples are pulled back to uniform samples on the surface. This shows the encoding map in the bottom row preserves the data distribution μ in the latent space.

In practice, many methods have been proposed to compute the encoding/decoding maps, such as VAE (variational autoencoder) (Kingma and Welling 2013; Jain et al. 2017), WAE (Wasserstein autoencoder) (Gelly et al. 2018), adversarial autoencoder (Makhzani et al. 2015), and so on.

ReLU Deep Neural Network

In deep learning, the deep neural networks are used to approximate mappings between Euclidean spaces. One of the most commonly used activation function is the ReLU function, $\sigma(x) = \max\{x, 0\}$. When x is positive, we say the neuron is *activated*. One neuron represents a function $\sigma(\sum_{i=1}^k \lambda_i x_i - b_i)$, where λ_i 's are *weights* and b_i the *bias*. Many neurons are connected to form a network. A ReLU deep neural network (DNN) represents a piecewise linear map.

Definition 13 (ReLU DNN). For any number of hidden layers $k \in \mathbb{N}$, input and output dimensions $w_0, w_{k+1} \in \mathbb{N}$, a $\mathbb{R}^{w_0} \rightarrow \mathbb{R}^{w_{k+1}}$ ReLU DNN is given by specifying a sequence of k natural numbers w_1, w_2, \dots, w_k representing widths of the hidden layers, a set of k affine transformations $T_i : \mathbb{R}^{w_{i-1}} \rightarrow \mathbb{R}^{w_i}$ for $i = 1, \dots, k$ and a linear transformation $T_{k+1} : \mathbb{R}^{w_k} \rightarrow \mathbb{R}^{w_{k+1}}$ corresponding to weights of hidden layers.

The mapping $\varphi_\theta : \mathbb{R}^{w_0} \rightarrow \mathbb{R}^{w_{k+1}}$ represented by this ReLU DNN is

$$\varphi_\theta = T_{k+1} \circ \sigma_k \circ T_k \circ \dots \circ T_2 \circ \sigma_1 \circ T_1, \tag{40}$$

where \circ denotes mapping composition, θ represent all the weight and bias parameters, and σ_i represents the mapping $\sigma_i : \mathbb{R}^{w_{i-1}} \rightarrow \mathbb{R}^{w_i}$ $\sigma_i = (\sigma_i^1, \sigma_i^2, \dots, \sigma_i^{w_i})$,

$$\sigma_i^j = \sigma \left(\sum_{k=1}^{w_{i-1}} \lambda_i^{jk} x_k - b_i^j \right).$$

Definition 14 (Activated Path). Given a point $\mathbf{x} \in \mathcal{X}$ in the input space \mathcal{X} , the *activated path* of \mathbf{x} consists all the activated neurons when $\varphi_\theta(\mathbf{x})$ is evaluated and denoted as $\rho(\mathbf{x})$. Then the activated path defines a set-valued function $\rho : \mathcal{X} \rightarrow 2^{\mathcal{S}}$ (\mathcal{S} is the set of all neurons; $2^{\mathcal{S}}$ are all the subsets of \mathcal{S}).

Fixing the parameter θ , the map φ_θ induces cell decompositions for the input space and the output space.

Definition 15 (Cell Decomposition). Fix a map φ_θ represented by a ReLU DNN, two data points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ are *equivalent*, denoted as $\mathbf{x}_1 \sim \mathbf{x}_2$, if they share the same activated path, $\rho(\mathbf{x}_1) = \rho(\mathbf{x}_2)$. Then each equivalence relation partitions the ambient space \mathcal{X} into cells,

$$\mathcal{D}(\varphi_\theta) : \mathcal{X} = \bigcup_{\alpha} U_{\alpha},$$

each equivalence class corresponds to a cell: $\mathbf{x}_1, \mathbf{x}_2 \in U_{\alpha}$ if and only if $\mathbf{x}_1 \sim \mathbf{x}_2$. $\mathcal{D}(\varphi_\theta)$ is called the cell decomposition induced by the encoding map φ_θ . The number of cells is denoted as $|\mathcal{D}(\varphi_\theta)|$.

Furthermore, φ_θ maps the cell decomposition in the ambient space $\mathcal{D}(\varphi_\theta)$ to a cell decomposition in the latent space. The restriction of φ_θ on each cell is a linear map. The number of cells in $\mathcal{D}(\varphi_\theta)$ describes the capacity of the network, namely, the learning capability of the network.

Definition 16 (Learning Capability). Given a ReLU DNN N with a fixed architecture, the complexity of the network $\mathcal{N}(N)$ is defined as the maximal number of cells of $\mathcal{D}(\varphi_\theta)$,

$$\mathcal{N}(N) := \max_{\theta} |\mathcal{D}(\varphi_\theta)|.$$

We can explicitly estimate the upper bound of the network capacity $\mathcal{N}(N)$. The maximum number of parts one can get when cutting d -dimensional space \mathbb{R}^d with n hyper-planes is denoted as $C(d, n)$, and then by induction, one can easily show that

$$C(d, n) = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{d}. \quad (41)$$

We can easily get the upper bound estimation.

Theorem 9. Given a ReLU DNN $N(w_0, \dots, w_{k+1})$, representing PL mappings $\varphi_\theta : \mathbb{R}^{w_0} \rightarrow \mathbb{R}^{w_{k+1}}$ with k hidden layers of widths $\{w_i\}_{i=1}^k$, then the complexity of N has an upper bound,

$$\mathcal{N}(N) \leq \prod_{i=1}^{k+1} C(w_{i-1}, w_i). \quad (42)$$

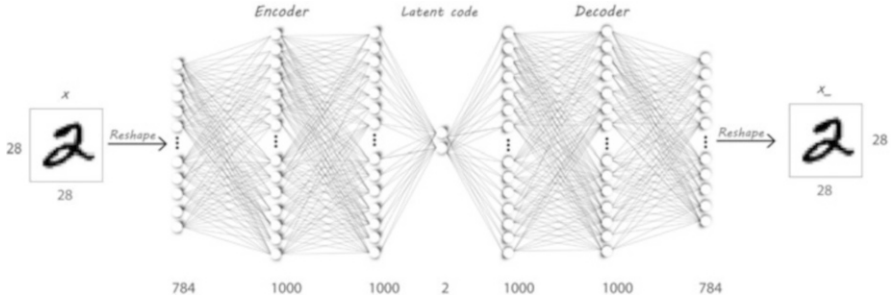


Fig. 14 Autoencoder architecture

AutoEncoder

One of the most popular models for learning the encoding and decoding maps is *AutoEncoder* as shown in Fig. 14. The *AutoEncoder* model consists two symmetric deep neural networks: the first network represents the encoder, and the second network represents the decoder. The numbers of nodes in the input and the output layers equal to the dimension of the ambient space. Between the encoder and decoder, there is a bottleneck layer. The number of nodes in the bottleneck layer equals to the dimension of the latent space.

We denote the ambient space as \mathcal{X} , latent space as \mathcal{Z} , encoding map $\varphi_\theta : \mathcal{X} \rightarrow \mathcal{Z}$, and decoding map $\psi_\theta : \mathcal{Z} \rightarrow \mathcal{X}$. We sample the data manifold $\Sigma \subset \mathcal{X}$ to get training samples $\{x_1, x_2, \dots, x_n\} \subset \Sigma$ and apply the L^2 -norm as the loss function \mathcal{L}_θ . The training process is the optimization

$$\min_{\theta} \mathcal{L}_\theta(x_1, \dots, x_n) = \min_{\theta} \sum_{i=1}^n |x_i - \psi_\theta \circ \varphi_\theta(x_i)|^2. \tag{43}$$

Figure 15 shows one example of surface embedding using an autoencoder. We uniformly sample the Buddha surface Σ in (a) and then train an autoencoder using formula Eq.43; the latent codes of the samples are shown in (b); the decoded surface $\tilde{\Sigma}$ is shown in (c). We can see the reconstructed surface is very similar to the input surface, with user-controlled Hausdorff distance. Figure 16 shows the cell decomposition of the ambient space \mathcal{X} and the latent space \mathcal{Z} induced by the encoding map φ_θ and the decoding map ψ_θ .

In the following, we analyze the accuracy of manifold learning using a surface example. Given the input surface Σ embedded in \mathbb{R}^3 , given any point $p \in \mathbb{R}^3$, the *closest point* on Σ to p is defined as

$$\pi(p, \Sigma) := \operatorname{argmin}_{q \in \Sigma} |p - q|^2.$$

The *medial axis* of the surface Σ is defined as

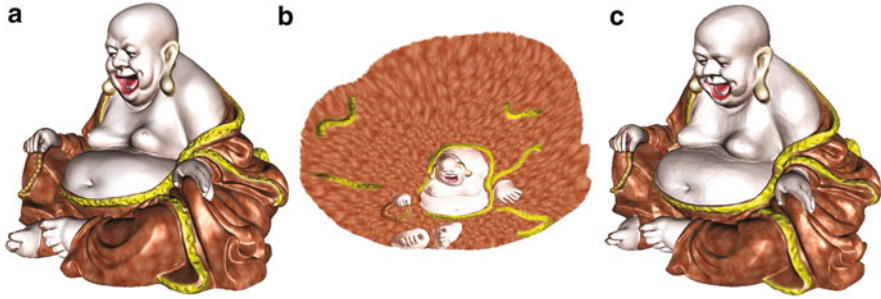


Fig. 15 Manifold embedding computed by an autoencoder. (a) Input manifold $\Sigma \subset X$ (b) latent representation $D = \varphi_\theta(M) \subset Z$ (c) reconstructed manifold $\tilde{\Sigma} = \psi_\theta(D) \subset X$

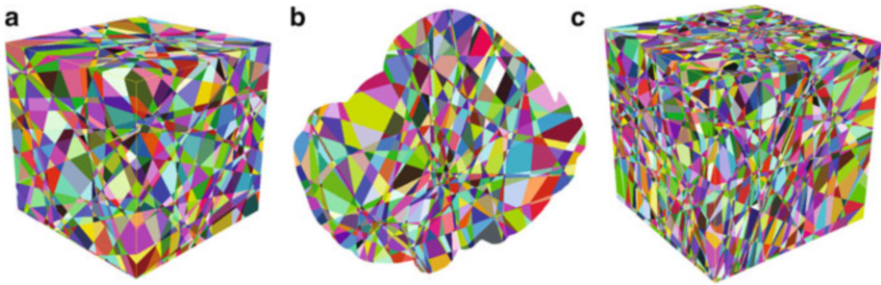


Fig. 16 The cell decompositions induced by the autoencoder. (a) cell decomposition $\mathcal{D}(\varphi_\theta)$ (b) latent space cell decomposition (c) cell decomposition $\mathcal{D}(\psi_\theta \circ \varphi_\theta)$

$$\Gamma(\Sigma) := \{p \in \mathbb{R}^3 : |\pi(p, \Sigma)| > 1\}.$$

where $|\cdot|$ represents the cardinality of the set. For any point $p \in \Sigma$, the *local feature size* of p is the distance from p to the medial axis $\Gamma(\Sigma)$. Suppose the samplings on Σ are $X = \{x_1, x_2, \dots, x_n\}$, such that, for any point $q \in \Sigma$, the geodesic disk $c(q, \delta)$ intersects X is non-empty, and the geodesic distance between any pair of samples is greater than ε , then X is called a (δ, ε) *sampling*. Given such a sampling, we can compute the geodesic Delaunay triangulation of X ; this induces a polyhedral surface $\tilde{\Sigma}$. By geometric approximation theory, suppose Σ is C^2 smooth, we can determine the parameters δ, ε by the injective radius, the principle curvature, and the local feature size, such that $\tilde{\Sigma}$ approximates the original surface Σ with arbitrary precision in terms of Hausdorff distance, Riemannian metric, Laplace-Beltami operator, curvature measures, and so on.

Assume the network capacity for the autoencoder is big enough, the (δ, ε) samples are the training set, and the optimization reduces the loss function to be 0; then the restriction of $\psi_\theta \circ \varphi_\theta$ equals to identity, and the autoencoder recovers $\tilde{\Sigma}$. By construction, the decoded surface approximates the original surface with user desired accuracy. This argument can be generalized to higher dimensional

manifolds. In reality, the data manifold is unknown, and it is hard to figure out its injective radius, curvatures, and local feature size; the optimization of deep networks often gets stuck at the local optima. There are many widely open challenges for learning the manifold structure.

Generative Adversarial Networks

Generative adversarial networks (GAN) are one of the most popular generative models in deep learning. It has many merits, such as it can automatically generate samples; the requirement for the data samples is reduced; and it can model arbitrary data distribution without closed form expression. As shown in Fig. 17, a GAN model includes two deep neural networks, the generator and the discriminator. The generator converts a white noise (user prescribed distribution in the latent space) to generated samples; the discriminator takes both the real data samples and the fake generated samples and verifies whether the current sample is authentic or fake.

Competition vs. Collaboration

The generator and the discriminator compete with each other; the generator improves the quality of the generated samples to confuse the discriminator, and the discriminator improves the discriminating capability and detect the fake samples. Eventually, the system reaches the Nash equilibrium; the discriminator cannot differentiate the generated ones from the real samples, and then the generated samples can be applied to real applications, such as training other recognition systems and so on.

Wasserstein GAN applies optimal transport method as shown in Fig. 18. The generator G computes the optimal transport map $g_\theta : \mathcal{Z} \rightarrow \Sigma$, which transforms

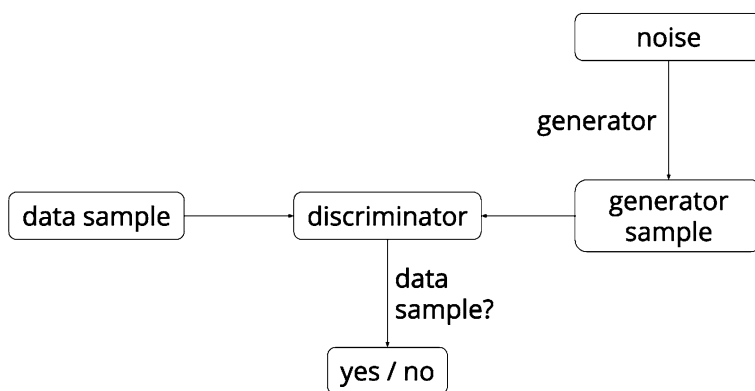


Fig. 17 The framework of a GAN model

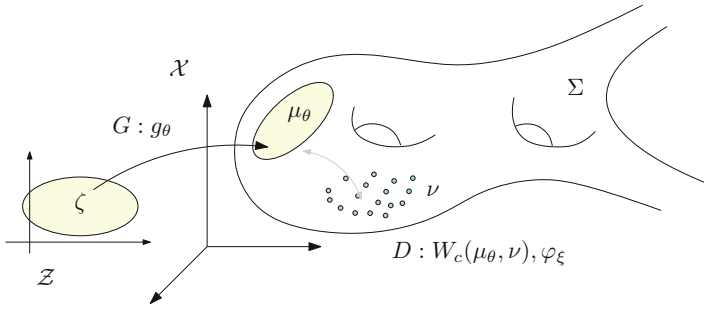


Fig. 18 The framework of a GAN model; \mathcal{Z} is the latent space, ζ the white noise, \mathcal{X} the image space, Σ the data manifold, G generator, D discriminator

the white noise ζ in the latent space \mathcal{Z} to the generated distribution $\mu_\theta = (g_\theta)_\# \zeta$. The discriminator D computes the Kantorovich potential φ_ξ and then computes the Wasserstein distance between μ_θ and the real data distribution ν

$$\mathcal{W}_c(\mu_\theta, \nu) = \max_{\varphi_\xi} \int_X \varphi_\xi(x) d\mu_\theta(x) + \int_Y \varphi_\xi^c(y) d\nu(y),$$

where X and Y should be the data manifold Σ ; in practice, they are replaced by the image space \mathcal{X} in Arjovsky et al. (2017). The whole training process of WGAN model is a min-max optimization:

$$\min_\theta \max_\xi \int_X \varphi_\xi(x) d\mu_\theta(x) + \int_Y \varphi_\xi^c(y) d\nu(y).$$

One can choose L^1 -cost, then $c(x, y) = |x - y|$, $\varphi^c = -\varphi$, given φ is 1-Lipsitz, then the WGAN model optimizes

$$\min_\theta \max_\xi \int_X \varphi_\xi \circ g_\theta(z) d\zeta(z) - \int_Y \varphi_\xi(y) d\nu(y),$$

namely,

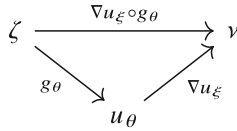
$$\min_\theta \max_\xi \mathbb{E}_{z \sim \zeta} (\varphi_\xi \circ g_\theta(z)) - \mathbb{E}_{y \sim \nu} (\varphi_\xi(y)),$$

with the constraint that φ_ξ is 1-Lipsitz.

If we use L^2 cost, then the discriminator computes the Kantorovich potential φ_ξ for the purpose of Wasserstein distance $\mathcal{W}_2(\mu_\theta, \nu)$; then the Brenier potential u_ξ and the optimal transport map T_ξ can be derived directly

$$u_\xi = \frac{1}{2}|x|^2 - \varphi_\xi(x), \quad T_\xi = \nabla u_\xi.$$

T_ξ transforms the generated distribution μ_θ to the real data distribution ν . The generator g_θ transforms ζ to μ_θ , and then the composition $\nabla u_\xi \circ g_\theta$ maps the latent white noise ζ to the data distribution ν , as shown in the following commutative diagram:



The generator seeks a measure preserving map to transform ζ to ν . In each optimization step, the generator finds the current g_θ , which gives a transport map from ζ to μ_ξ , and the discriminator computes u_ξ , which transport μ_ξ to ν . The composition $\nabla u_\xi \circ g_\theta$ gives a transport map from ζ to ν . Therefore, we can use $\nabla u_\xi \circ g_\theta$ to update the generator g_θ ; this will improve the convergence rate. Currently, the generator and the discriminator do not share intermediate computational results, which make the system highly inefficient. The competition between the generator and the discriminator should be replaced by collaboration.

Memorization vs. Learning

In general, deep neural networks have huge amount of parameters, such that their capacities are big enough to memorize all the training samples. So the following question is naturally raised:

Question 4. Memorization vs. Learning: Does a deep learning system really learn something or just memorize all the training samples?

Generally speaking, in deep learning applications, the real data distribution ν is approximated by the empirical distribution: $\hat{\nu} = \frac{1}{n} \sum_{i=1}^n \delta(y - y_i)$, where $\{y_1, y_2, \dots, y_n\}$ are the training samples, either the raw samples on the data manifold or the latent codes in the latent space. If we use the quadratic Euclidean distance as the cost function, then both the generator and the discriminator compute the optimal transport maps or equivalently the Brenier potentials. From the formula of the semi-discrete Brenier potential,

$$u_h = \max_{i=1}^n \{ \langle x, y_i \rangle - h_i \}$$

we can tell that the system really memorizes all the training samples $\{y_i\}$ but also learns the probability for each sample represented by $\{h_i\}$, which are obtained by nonlinear optimization.

Hence deep learning systems both memorize the training samples and learn the probability measure.

Mode Collapsing

GANs are sensitive to hyper-parameters and notoriously difficult to train. The training process is highly unstable and often diverge. GANs suffer from *mode collapsing*: the generated distributions often miss some modes in the training data set. For example, if a GAN model is trained to learn the MNIST data sets, which has multiple modes representing the ten handwritten digits, then the GAN model may only learn 6 of them and forget the other 4 modes, or it captures some modes in the intermediate stage but forgets part of them in the final stage. GANs also suffer from *mode mixture*: they generate unrealistic samples mixing different modes. As shown in Fig. 19, VAE (Kingma and Welling 2013) or WGAN (Arjovsky et al. 2017) models suffer from mode mixture; they generate unrecognizable handwritten digit images, which look like the interpolation/mixture of some digits. Figure 20 shows mode collapsing on CelebA data set using WGAN-GP (Gulrajani et al. 2017) and WGAN-div (Thoma et al. 2018) models.

Mode collapsing can be explained using the regularity theory of optimal transport maps. As shown in Fig. 21, we use Monte-Carlo method to compute the optimal transport map between the uniform distribution defined on a rectangle and that on a dumb bell shape. Even the target domain is simply connected, because it is concave; the OT map is discontinuous on the singular sets γ_1 and γ_2 as shown in the left frame.

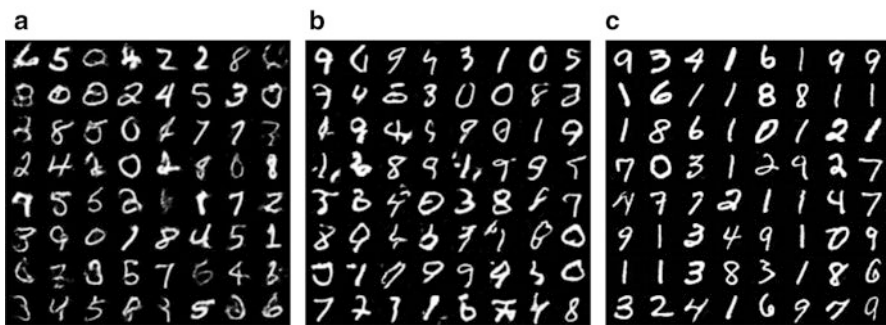


Fig. 19 Comparison between conventional models VAE and WGAN with our model AE-OT using MNIST data set. (a) VAE (b) WGAN (c) Our model, AE-OT



Fig. 20 Mode collapsing in WGAN-GP and WGAN-div model on CelebA data set. (a) WGAN-GP (b) WGAN-div

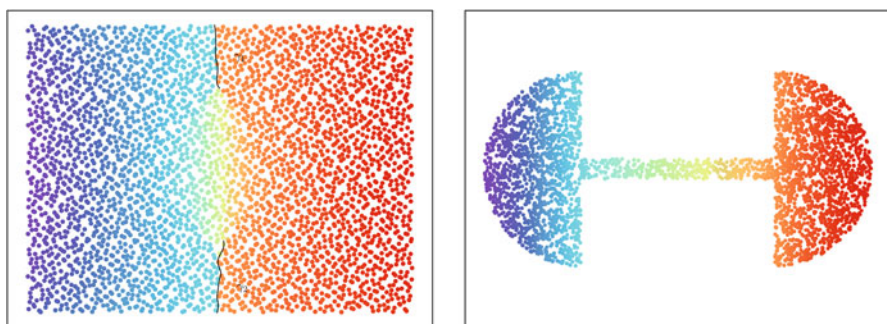


Fig. 21 Discontinuous optimal transport map, produced by a GPU implementation of the algorithm based on regularity theorem. γ_1 and γ_2 are two singularity sets

As we analyzed before, deep neural networks can only represent continuous mappings, but the optimal transport map is discontinuous given the target support is concave; this intrinsic conflict causes mode collapse and mode mixture.

If the target measure ν has multiple modes, namely its support has multiple connected components, then the continuous map may cover one connected component and miss the other modes; this induces mode collapse; or the continuous map covers all the modes but also the gaps among the modes, and then the samples generated in the gap area will mix samples from different modes; this induces mode mixture.

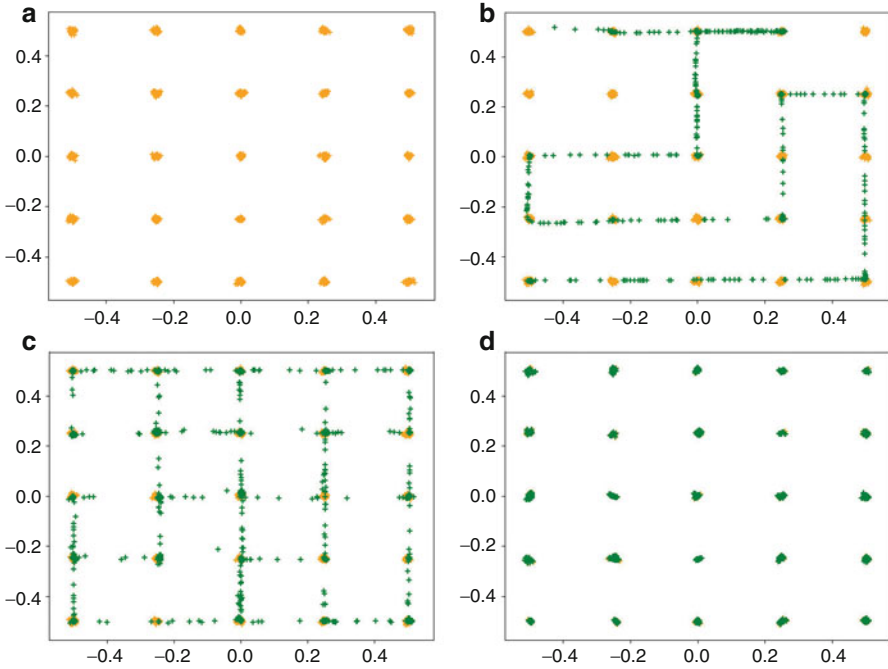


Fig. 22 Comparison between conventional models with AE-OT. (a) original (b) GAN (c) pacgan (d) Our model, AE-OT

As shown in Fig. 22, each orange spot represents a mode in frame (a); the GAN model (Goodfellow et al. 2014) misses some modes and also covers the gaps among the modes in frame (b); the pacgan model (Lin et al. 2018) covers all the modes but also covers the gaps among them. Hence GAN model and pacgan model suffer from both mode collapse and mode mixture.

In order to verify our hypothesis that the transport map is discontinuous on the singularity sets in real applications, we design and perform an experiment using human facial image data set celebA. As shown in Fig. 23, we use an autoencoder to encode the data manifold Σ to the latent space, $\varphi : \Sigma \rightarrow \mathcal{Z}$; φ push forwards the data distribution μ to the latent code distribution $\varphi_{\#}\mu$; then in the latent space, we compute an optimal transport map from a uniform distribution on the unit ball to the latent code distribution $\varphi_{\#}\mu$; we draw line segments in the unit ball, which are mapped to curves on the data manifold; each curve is an interpolation in the facial image set. As shown in Fig. 24, each row is an interpolation curve on the human facial image manifold.

As shown in Fig. 23, there are singularity sets in the unit ball, and a blue line segment intersects the singularity sets at p ; then $T(p)$ is outside the latent code set $\varphi(\Sigma)$; the decoded image $\varphi^{-1}(T(p))$ is outside the data manifold Σ . In this way, we can detect the boundary of the data manifold Σ . An image on the human facial

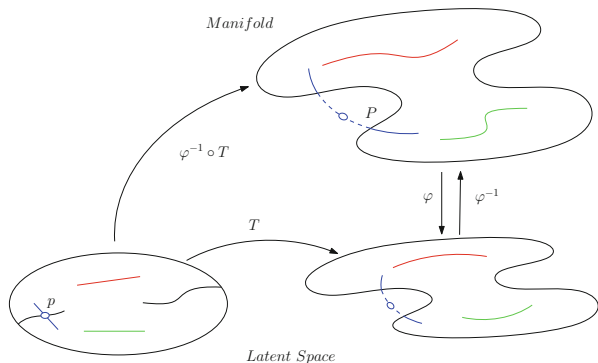


Fig. 23 Singularity set detection



Fig. 24 Interpolation curves on facial photo manifold

image manifold means a human face, which is physically “allowable,” satisfies all the anatomical and biological laws but with zero probability to appear in reality. As shown in Fig. 25, we start from a boy image with brown eyes and end at a girl image with blue eyes. In the middle of the interpolation, we generate a facial image with one blue eye and one brown eye. This type of human faces exist in real world, but the probability to encounter such a person is almost zero in practice. All the training facial images are either brown eyes or blue eyes; the generated facial image with different eye colors is on the boundary of the data manifold. This demonstrates that the existence of singularity set Γ and the transport map T is discontinuous at the Γ .

Fig. 25 Facial images generated by an AE-OT model, the central image shows the boundary of the facial photo manifold



AE-OT Model

In order to eliminate mode collapse, improve the stability, and make the whole model more understandable, we propose a novel generative model: AE-OT model. As shown in Fig. 26, the model consists two parts: AE and OT. The AE network is an autoencoder, which focuses on manifold learning and computes the encoding map $f_\theta : \Sigma \rightarrow \mathcal{Z}$ and the decoding map $g_\xi : \Sigma \rightarrow \mathcal{Z}$; the OT module is in charge of probability distribution transformation and finds the optimal transport map using our geometric variational approach. The OT module can be implemented either using a deep neural network and optimized by training or directly using geometric method, such as Monte Carlo OT algorithm on GPU.

The mode collapses in conventional generative models are mainly caused by the step of computing transport map, because the transport map is discontinuous, but DNNs can only represent continuous maps. The AE-OT model conquers this fundamental difficulty in the following way: observe Fig. 27, in the latent space the latent code distribution has multiple clusters; the support rectangle of the white noise is partitioned into 10 cells as well; each cell is mapped to a cluster with the same color. Therefore, the optimal transport map between the noise and the latent code is discontinuous across the cell boundaries. Instead of computing the OT map itself, the AE-OT model computes the Brenier potential (lower-left corner), which is continuous (but not globally differentiable) and representable by neural networks. Since the OT map covers all the clusters of the latent code distribution, and skips all the gaps among the clusters, no mode collapse or mode mixture can happen.

Fig. 26 The framework of AE-OT model

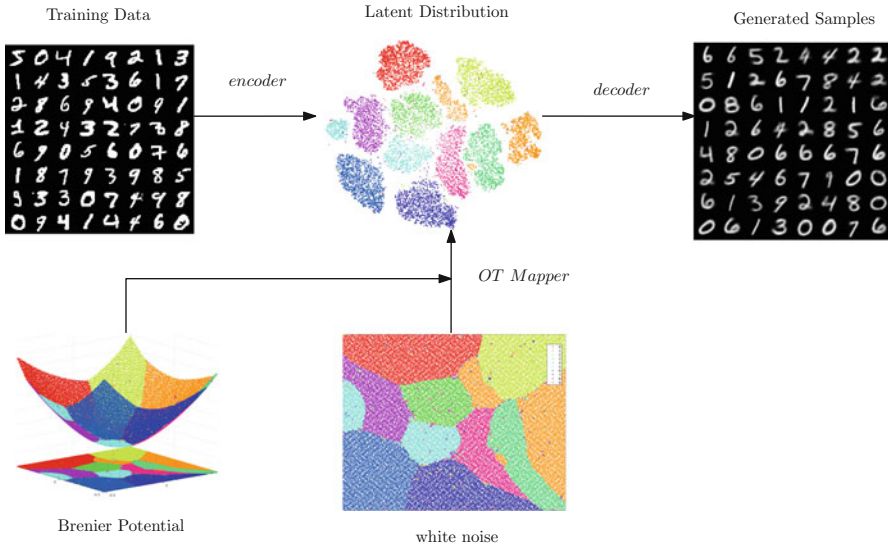
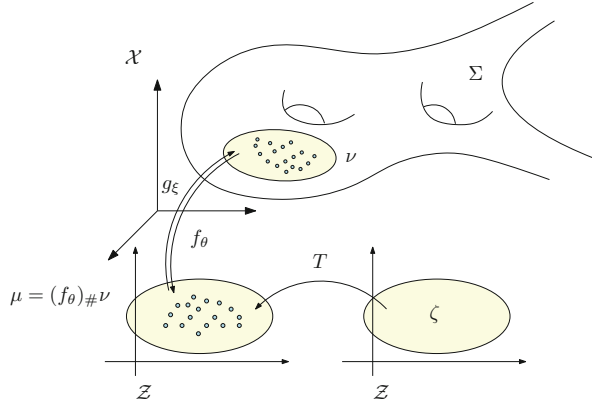


Fig. 27 AE-OT model for MNIST data set

Furthermore, the AE-OT model has the merits: solving Monge-Ampère equation is reduced to a convex optimization, which has a unique solution due to the Brenier Theorem 2. The optimization won't be trapped in a local optimum; the Hessian matrix of the energy has an explicit formulation. The Newton's method can be applied with second-order convergence; or the quasi-Newton's method can be used with super-linear convergence, whereas conventional gradient descend method has linear convergence. The approximation accuracy can be fully controlled by the density of the sampling using Monte-Carlo method; the algorithm can be refined to be hierarchical and self-adaptive to further improve the efficiency; the parallel algorithm can be implemented using GPU. By comparing Figs. 20 and 28,

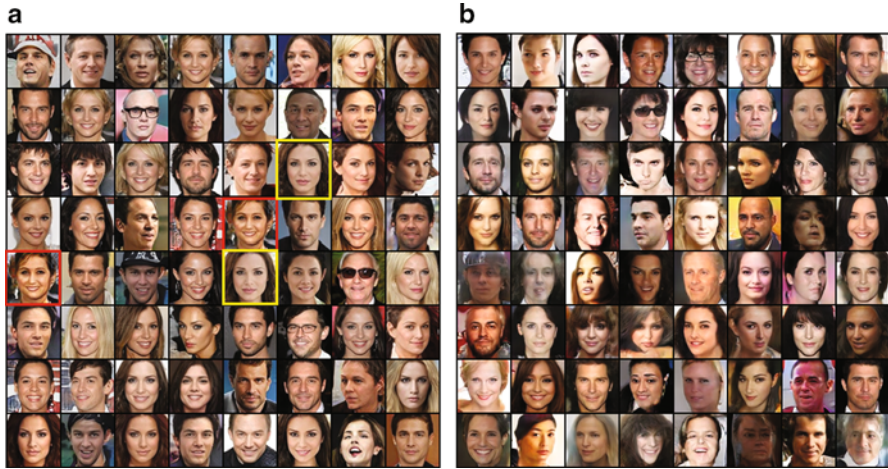


Fig. 28 Comparison between CRGAN (Mescheder et al. 2018) and our model. (a) CRGAN – mode collapsing (b) Our model



Fig. 29 Human facial images generated by our model

we can see that the AE-OT model greatly reduces the mode collapse and mode mixture. Figure 29 shows the generated facial images by training our model on the CelebAHQ data set.

Conclusion

This work focuses on a geometric view of optimal transport to understand deep learning models, such as generative adversarial networks (GANs). By manifold distribution principle, deep learning systems learn probability distributions on manifolds; therefore they have two major tasks: one is manifold learning, and the other is probability measure learning.

Manifold learning is reduced to construct encoding and decoding maps between the data manifold and the latent space. The probability distribution learning can be achieved by optimal transport methods. The Brenier theory in optimal transport has intrinsic relation with Alexandrov theorem in convex geometry via Monge-Ampère equation. This leads to a geometric variational algorithm to compute optimal transport maps. By applying OT theory, we analyze the conventional generative models and find that the generator and discriminator in a GAN model should collaborate instead of compete with each other; the GAN model both memorizes all the training samples and learns the probability measure; furthermore, the regularity theory of Monge-Ampère equation explains the intrinsic reason for mode collapse. In order to eliminate mode collapse, a novel AE-OT model is introduced, which computes the continuous Brenier potential instead of the discontinuous transport maps.

Optimal transport theory and Riemannian geometry lay down the theoretic foundation of deep learning. In the future, we will explore further to use modern geometry theories to understand deep learning algorithms and design novel models.

References

- Alexandrov, A.D.: *Convex polyhedra* Translated from the 1950 Russian edition by N.S. Dairbekov, S.S. Kutateladze, A.B. Sossinsky. Springer Monographs in Mathematics (2005)
- An, D., Guo, Y., Lei, N., Luo, Z., Yau, S.-T., Gu, X.: Ae-ot: A new generative model based on extended semi-discrete optimal transport. In: *International Conference on Learning Representations* (2020)
- Angenent, S., Haker, S., Tannenbaum, A.: Minimizing flows for the monge-kantorovich problem. *SIAM J. Math. Ann.* **35**(1), 61–97 (2003)
- Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: *ICML*, pp. 214–223 (2017)
- Benamou, J.-D., Brenier, Y.: A numerical method for the optimal time-continuous mass transport problem and related problems. In: Caffarelli, L.A., Milman, M. (eds.) *Monge Ampère Equation: Applications to Geometry and Optimization* (Deerfield Beach, FL), volume 226 of *Contemporary Mathematics*, pp. 1–11, Providence (1999) American Mathematics Society
- Bonnotte, N.: From knothe’s rearrangement to Brenier’s optimal transport map. *SIAM J. Math. Anal.* **45**(1), 64–87 (2013)
- Brenier, Y.: Polar factorization and monotone rearrangement of vector-valued functions. *Commun. Pure Appl. Math.* **44**(4), 375–417 (1991)
- Caffarelli, L.A.: Some regularity properties of solutions of monge–ampère equation. *Commun. Pure Appl. Math.* **44**(8–9), 965–969 (1991)
- Cui, L., Qi, X., Wen, C., Lei, N., Li, X., Zhang, M., Gu, X.: Spherical optimal transportation. *Comput. Aided Des.* **115**, 181–193 (2019)

- Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 26*, pp. 2292–2300. Curran Associates, Inc. (2013)
- Dai, B., Wipf, D.: Diagnosing and enhancing VAE models. In: *International Conference on Learning Representations* (2019)
- De Goes, F., Breeden, K., Ostromoukhov, V., Desbrun, M.: Blue noise through optimal transport. *ACM Trans. Graph.* **31**(6), 171 (2012)
- De Goes, F., Cohen-Steiner, D., Alliez, P., Desbrun, M.: An optimal transport approach to robust reconstruction and simplification of 2D shapes. In: *Computer Graphics Forum*, vol. 30, pp. 1593–1602. Wiley Online Library (2011)
- Dominitz, A., Tannenbaum, A.: Texture mapping via optimal mass transport. *IEEE Trans. Vis. Comput. Graph.* **16**(3), 419–433 (2010)
- Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: <https://arxiv.org/abs/1907.02544> (2019)
- Figalli, A.: Regularity properties of optimal maps between nonconvex domains in the plane. *Communications in Partial Differential Equations*, **35**(3), 465–479 (2010)
- Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *NIPS*, pp. 2672–2680 (2014)
- Gu, D.X., Luo, F., Sun, J., Yau, S.-T.: Variational principles for minkowski type problems, discrete optimal transport, and discrete monge–ampère equations. *Asian J. Math.* **20**, 383–398 (2016)
- Guan, P., Wang, X.-J., et al.: On a monge-ampere equation arising in geometric optics. *J. Diff. Geom.* **48**(48), 205–223 (1998)
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: *NIPS*, pp. 5769–5779 (2017)
- Gutiérrez, C.E., Huang, Q.: The refractor problem in reshaping light beams. *Arch. Ration. Mech. Anal.* **193**(2), 423–443 (2009)
- Gelly, S., Schoelkopf, B., Tolstikhin, I., Bousquet, O.: Wasserstein auto-encoders. In: *ICLR* (2018)
- Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2017)
- Jain, U., Zhang, Z., Schwing, A.G.: Creativity: Generating diverse questions using variational autoencoders. In: *CVPR*, pp. 5415–5424 (2017)
- Jeff Donahue, T.D., Krähenbühl, P.: Adversarial feature learning. In: *International Conference on Learning Representations* (2017)
- Kantorovich, L.V.: On a problem of monge. *J. Math. Sci.* **133**(4), 1383–1383 (2006)
- Kantorovich, L.V.: On a problem of monge. *Uspekhi Mat. Nauk.* **3**, 225–226 (1948)
- Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
- Lindbo Larsen, A.B., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric (2016)
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2017)
- Lei, N., An, D., Guo, Y., Su, K., Liu, S., Luo, Z., Yau, S.-T., Gu, X.: A geometric understanding of deep learning. *Engineering* **6**(3), 361–374 (2020)
- Lei, N., Su, K., Cui, L., Yau, S.-T., Gu, X.D.: A geometric view of optimal transportation and generative model. *Comput. Aided Geom. Des.* **68**, 1–21 (2019)
- Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. In: *Advances in Neural Information Processing Systems*, pp. 1505–1514 (2018)
- Liu, H., Gu, X., Samaras, D.: Wasserstein gan with quadratic transport cost. In: *ICCV* (2019)
- Ma, X.N., Trudinger, N.S., Wang, X.J.: Regularity of potential functions of the optimal transportation problem. *Arch. Ration. Mech. Anal.* **177**(2), 151–183 (2005)
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)

- Mérogot, Q.: A multiscale approach to optimal transport. In: *Computer Graphics Forum*, vol. 30, pp. 1583–1592. Wiley Online Library (2011)
- Mescheder, L.M., Nowozin, S., Geiger, A.: Which training methods for gans do actually converge? In: *International Conference on Machine Learning (ICML)* (2018)
- Meyron, J., Mérogot, Q., Thibert, B.: Light in power: a general and parameter-free algorithm for caustic design. In: *SIGGRAPH Asia 2018 Technical Papers*, p. 224. ACM (2018)
- Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: *ICLR* (2018)
- Nadeem, S., Su, Z., Zeng, W., Kaufman, A.E., Gu, X.: Spherical parameterization balancing angle and area distortions. *IEEE Trans. Vis. Comput. Graph.* **23**(6), 1663–1676 (2017)
- Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: *ICLR* (2016)
- Razavi, A., Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: *ICLR 2019 Workshop DeepGenStruct* (2019)
- Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
- Salakhutdinov, R., Burda, Y., Grosse, R.: Importance weighted autoencoders. In: *ICML* (2015)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
- Simonyany, K., Brock, A., Donahuey, J.: Large scale gan training for high fidelity natural image synthesis. In: *International Conference on Learning Representations* (2019)
- Solomon, J., de Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* **34**, 1–11 (2015a)
- Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* **34**(4), 66 (2015b)
- Solomon, J., Rustamov, R., Guibas, L., Butscher, A.: Earth mover’s distances on discrete surfaces. *ACM Trans. Graph.* **33**(4), 67 (2014)
- Su, K., Chen, W., Lei, N., Cui, L., Jiang, J., Gu, X.D.: Measure controllable volumetric mesh parameterization. *Comput. Aided Des.* **78**(C), 188–198 (2016)
- Su, K., Chen, W., Lei, N., Zhang, J., Qian, K., Gu, X.: Volume preserving mesh parameterization based on optimal mass transportation. *Comput. Aided Des.* 82:42–56 (2017)
- Su, K., Cui, L., Qian, K., Lei, N., Zhang, J., Zhang, M., Gu, X.D.: Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. *Comput. Aided Geom. Des.* **46**(C):76–91 (2016)
- Su, Z., Zeng, W., Shi, R., Wang, Y., Sun, J., Gu, X.: Area preserving brain mapping. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2235–2242 (2013)
- Thoma, J., Acharya, D., Van Gool, L., Wu, J., Huang, Z.: Wasserstein divergence for gans. In: *ECCV* (2018)
- ur Rehman, T., Haber, E., Pryor, G., Melonakos, J., Tannenbaum, A.: 3D nonrigid registration via optimal mass transport on the GPU. *Med. Image Anal.* **13**(6), 931–940 (2009)
- van den Oord, K.K.A., Vinyals, O.: Neural discrete representation learning. In: *NeurIPS* (2017)
- Villani, C.: *Optimal transport: Old and new*, vol. 338. Springer Science & Business Media (2008)
- Wang, X.-J.: On the design of a reflector antenna. *Inverse Prob.* **12**(3), 351 (1996)
- Wang, X.-J.: On the design of a reflector antenna II. *Calc. Var. Partial Differ. Equ.* **20**(3), 329–341 (2004)
- Xiao, C., Zhong, P., Zheng, C.: Bourgan: Generative networks with metric embeddings. In: *NeurIPS* (2018)
- Yau, S.-T.: *SS Chern: A great geometer of the twentieth century*. International PressCo (1998)
- Yu, X., Lei, N., Zheng, X., Gu, X.: Surface parameterization based on polar factorization. *J. Comput. Appl. Math.* **329**(C), 24–36 (2018)



Image Reconstruction in Dynamic Inverse Problems with Temporal Models

48

Andreas Hauptmann, Ozan Öktem, and Carola Schönlieb

Contents

Introduction	1708
Outline of Survey	1710
Spatiotemporal Inverse Problems	1711
Reconstruction Without Explicit Temporal Models	1712
Reconstruction Using a Motion Model	1714
Reconstruction Using a Deformable Template	1715
Motion Models Based on Partial Differential Equations	1718
Physical Motion Constraints	1718
Deformable Templates Given by Diffeomorphisms	1722
Flow of Diffeomorphisms and Intensities	1723
Deformable Templates by Metamorphosis	1724
Spatiotemporal Reconstruction with LDDMM	1725
Data-Driven Approaches	1727
Data-Driven Reconstruction Without Temporal Modelling	1729

A. Hauptmann
Research Unit of Mathematical Sciences, University of Oulu, Oulu, Finland

Department of Computer Science, University College London, London, UK
e-mail: andreas.hauptmann@oulu.fi

O. Öktem (✉)
Department of Information Technology, Division of Scientific Computing, Uppsala University,
Uppsala, Sweden

Department of Mathematics, KTH – Royal Institute of Technology, Stockholm, Sweden
e-mail: ozan@kth.se

C.-B. Schönlieb
Department of Applied Mathematics and Theoretical Physics, University of Cambridge,
Cambridge, UK
e-mail: cbs31@cam.ac.uk

Learning Deformation Operators	1730
Learning Motion Models	1732
Outlook and Conclusions	1733
References	1734

Abstract

This paper surveys variational approaches for image reconstruction in dynamic inverse problems. Emphasis is on variational methods that rely on parametrized temporal models. These are encoded here as diffeomorphic deformations with time-dependent parameters or as motion-constrained reconstructions where the motion model is given by a differential equation. The survey also includes recent developments in integrating deep learning for solving these computationally demanding variational methods. Examples are given for 2D dynamic tomography, but methods apply to general inverse problems.

Keywords

Image registration · Indirect registration · Inverse problems · Regularization · Tomography · Image reconstruction · Deep learning

Introduction

Dynamic inverse problems in imaging refer to the case when the object being imaged undergoes a temporal evolution during the data acquisition. The resulting data in such an inverse problem is a time (or quasi-time) series and due to limited sampling speed typically highly undersampled. Failing to account for the dynamic nature of the imaged object will lead to severe degradation in image quality, and hence there is a strong need for advanced modeling of the involved dynamics by incorporating temporal models in the reconstruction task.

The need for dynamic imaging arises, for instance, in various tomographic imaging studies in medicine, such as imaging moving organs (respiratory and cardiac motion) with computed tomography (CT) (Kwong et al. 2015), positron emission tomography (PET), or magnetic resonance imaging (MRI) (Lustig et al. 2006), and in functional imaging studies by means of dynamic PET (Rahmim et al. 2019) or functional MRI (Glover 2011). In functional imaging studies, the dynamic information is crucial for the diagnostic value to assess functionality of organs or tracking an injected tracer. Spatiotemporal imaging also arises in life sciences (Mokso et al. 2014) where it is crucial to understand dynamics and interactions of organisms. Lastly, applications in material sciences (De Schryver et al. 2018; Ruhlandt et al. 2017) and process monitoring (Chen et al. 2018) rely on the capabilities of dynamic image reconstruction.

Mathematically, solving dynamic inverse problems in imaging or spatiotemporal image reconstruction aims to recover a time-dependent image from a measured time

series. Since the measured time series is typically highly undersampled in each time instance, the reconstruction task is ill-posed, and additional prior knowledge is needed to recover a meaningful spatiotemporal image. One such prior assumption can be made on the type of dynamics in the studied object, which can regularize the reconstruction task by penalizing unrealistic motion.

There are various approaches in the literature for solving dynamic inverse problems. In this paper, we focus on variational models for this task which occupy a relatively large space in this context in the literature. Here, we identify two subgroups: those variational approaches which incorporate prior temporal information in the regularizer without a physical motion model but as a smoothness prior, e.g., as in Niemi et al. (2015) for slowly evolving images, and those variational approaches which incorporate prior temporal information in the model by motion constraints characterized either by an evolutionary PDE for the reconstruction or by a registration approach with a time-dependent deformation operator that is applied to a template.

The former, variational methods with a temporal smoothness prior, are applicable to a wide range of dynamic inverse problems as outlined in Schmitt and Louis (2002) and Schmitt et al. (2002). Indeed, the absence of an explicit motion constraint makes these methods more generally applicable. Some imaging-related applications are Feng et al. (2014), Lustig et al. (2006), and Steeden et al. (2018) for spatiotemporal compressed sensing in dynamic MRI. Here, the temporal regularity is enforced by a sparsifying transform (or total variation). Further examples are μ CT imaging of dynamic processes (Bubba et al. 2017; Niemi et al. 2015) and process monitoring with electrical resistance tomography (Chen et al. 2018).

The latter, variational methods featuring explicit motion models, can be divided in two categories. The first ones model the motion as an evolutionary PDE (Burger et al. 2017, 2018; Dirks 2015; Frerking 2016) using optical flow (Horn and Schunck 1981) or a continuity equation (Burger et al. 2018; Lang et al. 2019a), either as a constraint or in the form of a penalty term in the variational reconstruction model. Some prominent applications of this approach are in dynamic photoacoustic tomography (Lucka et al. 2018) and 3D computed tomography (Djurabekova et al. 2019), just to name a few. The second one parametrizes the dynamics in the form of a time-dependent diffeomorphic deformation operator (Younes 2019). Examples for such deformation models are LDDMM (Beg et al. 2005; Miller et al. 2006; Trouvé and Younes 2015) and metamorphosis (Younes 2019, Chapter 13). Dynamic image reconstruction is then modeled as an indirect registration task, as in Gris et al. (2020) with metamorphosis or Chen et al. (2019) and Lang et al. (2019b) using LDDMM. See also Yang et al. (2013) and Chen and Öktem (2018) for surveys on this topic.

Recently, deep neural network approaches have also entered the picture as a mean to approximate the solution to the computationally demanding variational approaches discussed above. Examples for these are Schlemper et al. (2017), Hauptmann et al. (2019), and Kofler et al. (2019) for dynamic image reconstruction without incorporating physical motion models and Qin et al. (2018), Liu et al. (2019), and Pouchol et al. (2019) for learned indirect registration approaches.

Outline of Survey

The survey focuses on variational methods for recovering a tomographic image that undergoes temporal evolution.

Section “[Spatiotemporal Inverse Problems](#)” is an overview of various approaches for reconstruction in such a setting. It starts with a mathematical formalization of a spatiotemporal inverse problem that is given as the task of solving an (time dependent) operator equation. This is followed by specifying various variational approaches for reconstruction that differ according to how the temporal model is specified. Section “[Reconstruction Without Explicit Temporal Models](#)” outlines a setup of a variational approach for reconstruction in a setting when one lacks an explicit temporal model resulting in (4). Such an approach is however not further explored in this survey; instead, focus is on a setting where there is an explicit temporal model and here the survey considers two variants.

In the first (section “[Reconstruction Using a Motion Model](#)”), the temporal model is given as the solution to an operator equation with a time-dependent parameter as in (7). The resulting variational model for reconstruction can be expressed as in (13). Section “[Motion Models Based on Partial Differential Equations](#)” further develops this formulation by considering partial differential equation (PDE)-based formulations.

In the second (section “[Reconstruction Using a Deformable Template](#)”), the temporal model is given by applying a parametrized deformation operator to a template in which the parameter is time dependent. This results in a temporal model of the form (15) that can be incorporated into a variational approach for reconstruction as in (17). This is followed by an outline of two approaches when data is time discretized. Section “[Deformable Templates Given by Diffeomorphisms](#)” builds on these approaches by considering explicit diffeomorphic deformation operators given by solving a flow equation.

As already stated, section “[Motion Models Based on Partial Differential Equations](#)” outlines how PDE-based motion models can be used for spatiotemporal reconstruction through (13). Likewise, section “[Deformable Templates Given by Diffeomorphisms](#)” outlines approaches based on (17) in which the deformation operator is given by solving an ordinary differential equation (ODE).

Section “[Data-Driven Approaches](#)” reviews data-driven approaches that have been developed for improving upon the computational feasibility of the variational models in sections “[Deformable Templates Given by Diffeomorphisms](#)” and “[Motion Models Based on Partial Differential Equations](#)”. In particular, section “[Data-Driven Reconstruction Without Temporal Modelling](#)” outlines data-driven methods that can be viewed as building on section “[Reconstruction Without Explicit Temporal Models](#)”. Similarly, one can see section “[Learning Motion Models](#)” as a data-driven extension of sections “[Reconstruction Using a Motion Model](#)” and “[Motion Models Based on Partial Differential Equations](#)” and section “[Learning Deformation Operators](#)” as a data-driven extension of the methods

in sections “[Reconstruction Using a Deformable Template](#)” and “[Deformable Templates Given by Diffeomorphisms](#)”.

The survey ends with an outlook and conclusions (section “[Outlook and Conclusions](#)”).

Spatiotemporal Inverse Problems

The starting point is to mathematically formalize the notion of a spatiotemporal inverse problem, which refers to the task of recovering a time-dependent image from (time-dependent) noisy indirect observations (Schmitt and Louis 2002).

Image: The time-dependent image is formally represented by a function $f: [0, T] \times \Omega \rightarrow \mathbb{R}^k$ where k is the number of image channels ($k = 1$ for gray scale images) and $\Omega \subset \mathbb{R}^d$ is the image domain.

We henceforth assume $f(t, \cdot) \in X$ where X (reconstruction space) is some vector space of \mathbb{R}^k -valued functions on $\Omega \subset \mathbb{R}^d$ that, unless otherwise stated, is a Hilbert space under the L^2 -inner product.

Data: Data is represented by a time-dependent function $g: [0, T] \times M \rightarrow \mathbb{R}^l$ where M is some manifold that is defined by the acquisition geometry and l is the number of data channels. Likewise, we assume that $g(t, \cdot) \in Y$ where Y (data space) is some vector space of \mathbb{R}^l -valued functions on M that, unless otherwise stated, is a Hilbert space under the L^2 -inner product. Actual measured data represents a digitization of this function by sampling on $[0, T] \times M$.

Spatiotemporal inverse problem: This is the task of recovering a temporal image $t \mapsto f(t, \cdot) \in X$ from time series data $t \mapsto g(t, \cdot) \in Y$ where

$$g(t, \cdot) = \mathcal{A}(t, f(t, \cdot))(t, \cdot) + e(t, \cdot) \quad \text{on } M \text{ for } t \in [0, T]. \quad (1)$$

Note here that $\mathcal{A}(t, \cdot): X \rightarrow Y$ is a (possibly time-dependent) forward operator. It models how an image $f(t, \cdot)$ at time t gives rise to data $g(t, \cdot)$ at time t in the absence of noise or measurement errors. The observation noise in data is accounted for by $e(t, \cdot) \in Y$, which can be seen as a single random realization of a Y -valued random variable that models measurement noise.

Remark 1. The formulation in (1) also covers cases when noise in data depends on the signal strength, like Poisson noise. Simply assume $e(t, \cdot)$ in (1) is a sample of the random variable $\mathbf{e}(t, \cdot) := \mathbf{g}(t, \cdot) - \mathcal{A}(t, f(t, \cdot))$ where $\mathbf{g}(t, \cdot)$ is the Y -valued random variable generating data.

Special cases of (1) arise depending on how the time dependency enters into the problem. In particular, the following three components can depend on time independently of each other:

- (a) Forward operator: The forward model may depend intrinsically on time.
- (b) Data acquisition geometry: The way the forward operator is sampled has a specific time dependency.
- (c) Image: The image to be recovered depends on time.

Next, an important special case is when data in (1) is observed at discrete time instances $0 \leq t_0 < \dots < t_n \leq T$; see also Schmitt and Louis (2002). Then, (1) reduces to the task of recovering images $f_j \in X$ from data $g_j \in Y$ where

$$g_j = \mathcal{A}_j(f_j) + e_j \quad \text{for } j = 1, \dots, n. \quad (2)$$

In the above, we have made use of the following notation for $j = 1, \dots, n$:

$$\begin{aligned} g_j &:= g(t_j, \cdot) \in Y & f_j &:= f(t_j, \cdot) \in X \\ e_j &:= e(t_j, \cdot) \in Y & \mathcal{A}_j &:= \mathcal{A}(t_j, \cdot): X \rightarrow Y. \end{aligned} \quad (3)$$

Reconstruction Without Explicit Temporal Models

The inverse problem in (1) is almost always ill-posed, so solving it requires regularization regarding both the spatial and temporal variation of the image. A variational approach for reconstructing the image trajectory $t \mapsto f(t, \cdot)$ that does not use any explicit temporal model reads as

$$\arg \min_{t \mapsto f(t, \cdot) \in X} \int_0^T \left[\mathcal{L} \left(\mathcal{A}(t, f(t, \cdot)), g(t, \cdot) \right) + \mathcal{G}_\theta(t, f(t, \cdot)) \right] dt. \quad (4)$$

Here, $\mathcal{L}: Y \times Y \rightarrow \mathbb{R}$ is the data fidelity term (data-fit), which is ideally chosen as an appropriate affine transform of the negative log-likelihood of data (Bertero et al. 2008). The term $\mathcal{G}_\theta: X \rightarrow \mathbb{R}$ is a parametrized regularizer that accounts for a priori knowledge about the image. It is common to separately regularize the spatial and temporal components, e.g., by considering

$$\mathcal{G}_\theta(t, f(t, \cdot)) := \mathcal{S}_\gamma(f(t, \cdot)) + \mathcal{T}_\tau(\partial_t f(t, \cdot)) \quad \text{for } \theta = (\gamma, \tau).$$

In the above, $\mathcal{S}_\gamma: X \rightarrow \mathbb{R}$ is a spatial regularizer, and $\mathcal{T}_\tau: X \rightarrow \mathbb{R}$ is a temporal regularizer. The spatial regularizer is commonly of the form $\mathcal{S}_\gamma := \gamma \mathcal{S}$ where $\gamma > 0$ and $\mathcal{S}: X \rightarrow \mathbb{R}$ is some “energy” functional. There is a well-developed theory for how to choose the latter in order to promote solutions of an inverse problem with specific type of regularity, e.g., a suitable choice for $\mathcal{H}^1(\Omega)$ -regularity is

$$\mathcal{S}(f) := \int_\Omega |\nabla f(x)|^2 dx. \quad (5)$$

On the other hand, if the image has edges that need to be preserved, then $BV(\Omega)$ -regularity is more natural and a total variation (TV)-regularizer is a better choice (Rudin et al. 1992). This regularizer is for $f \in W^{1,1}(\Omega)$ expressible as

$$\mathcal{S}(f) := \int_{\Omega} |\nabla f(x)| dx. \quad (6)$$

Other choices may include higher order terms to the total variation functional, like in total generalized variation; see Benning and Burger (2018) and Scherzer et al. (2009) for a survey.

The choice of temporal regularizer is much less explored. This functional accounts for a priori temporal regularity. Similarly to (5) one can here think of a smoothness prior (Niemi et al. 2015) for slowly evolving images

$$\mathcal{T}(\partial_t f) := \int_{\Omega} |\partial_t f(x)|^2 dx, \quad (7)$$

or a total variation type of penalty (Feng et al. 2014) for changes that are small or occur stepwise (image changes stepwise). The regularizer (7) acts pointwise in time, and full temporal dependency is obtained by integrating over time in (4).

Methods for solving (1) based on (4) can be used when there is no explicit temporal model that connects images and data across time. Hence, such methods are applicable to a wide range of dynamic inverse problems as outlined in Schmitt and Louis (2002) and Schmitt et al. (2002). More specific imaging-related applications are Feng et al. (2014), Lustig et al. (2006), and Steeden et al. (2018) for spatiotemporal compressed sensing in dynamic MRI. Here, the temporal regularity is enforced by a sparsifying transform (or total variation). Further examples are μ CT imaging of dynamic processes (Bubba et al. 2017; Niemi et al. 2015) and process monitoring with electrical resistance tomography (Chen et al. 2018).

Remark 2. When data is time discretized, then one also has the option to consider reconstructing images at each time step independently. An example of this is to recover the image at t_j by using a variational regularization method, i.e., as $f_j \approx \widehat{f}_j$ where

$$\widehat{f}_j := \arg \min_{f \in X} \left\{ \mathcal{L}(\mathcal{A}_j(f), g_j) + \mathcal{S}_{\gamma_j}(f) \right\} \quad \text{for } j = 1, \dots, n. \quad (8)$$

Our emphasis will henceforth be on methods for solving (1) that utilize more explicit temporal models.

Reconstruction Using a Motion Model

The idea here is to assume that a solution $t \mapsto f(t, \cdot) \in X$ to (1) has a time evolution that can be modeled by a *motion model*. Restating this assumption mathematically, we assume there is an operator $\Psi : [0, T] \times X \rightarrow X$ (motion model) such that

$$\Psi(t, f(t, \cdot)) = 0 \quad \text{on } \Omega \text{ whenever } t \mapsto f(t, \cdot) \text{ solves (1).} \quad (9)$$

Hence, (1) can be rephrased as the task of recovering the image trajectory $t \mapsto f(t, \cdot) \in X$ along with its motion model $\Psi : [0, T] \times X \rightarrow X$ from time series data $t \mapsto g(t, \cdot) \in Y$ where

$$\begin{aligned} g(t, \cdot) &= \mathcal{A}(t, f(t, \cdot))(t, \cdot) + e(t, \cdot) \text{ on } M \\ \text{s.t. } \Psi(t, f(t, \cdot)) &= 0 \text{ on } \Omega. \end{aligned} \quad \text{for } t \in [0, T]. \quad (10)$$

Parametrized Motion Models

An important special case is when the motion model depends only on time through a time-dependent parameter, i.e., there is $\Psi_\theta : X \rightarrow X$ for $\theta \in \Theta$ such that

$$\Psi_{\theta_t}(f(t, \cdot)) = 0 \quad \text{on } \Omega \text{ whenever } t \mapsto f(t, \cdot) \text{ solves (1),} \quad (11)$$

for some $t \mapsto \theta_t$. Then, (1) can be rephrased as the task to recover $t \mapsto f(t, \cdot) \in X$ along with motion parameter $t \mapsto \theta_t \in \Theta$ from time series data $t \mapsto g(t, \cdot) \in Y$ where

$$\begin{aligned} g(t, \cdot) &= \mathcal{A}(t, f(t, \cdot))(t, \cdot) + e(t, \cdot) \text{ on } M \\ \text{s.t. } \Psi_{\theta_t}(f(t, \cdot)) &= 0 \text{ on } \Omega. \end{aligned} \quad \text{for } t \in [0, T]. \quad (12)$$

The assumption in (11) may act as a regularization since it introduces a model for how images vary across time. In particular, the inverse problem in (12) is challenging but still easier to handle than the one in (1). However, solving (12) will still most likely require regularization. Approaches surveyed in section “[Motion Models Based on Partial Differential Equations](#)” represent different ways for doing this based on the setting where $\Psi_\theta : X \rightarrow X$ is given as a differential operator (involving differentiation in both temporal and spatial variables). Then parameter set Θ is a vector space of vector fields $\theta : \Omega \rightarrow \mathbb{R}^d$ with sufficient regularity, so θ_t corresponds to a velocity field. With these assumptions, (11) is a differential equation that constrains the temporal evolution of the solution to (1), and (12) corresponds to reconstructing the image jointly with its motion model.

General Variational Formulation

It is quite natural to adopt a variational approach for solving (12), cf. Burger et al. (2018). In fact, many of the state-of-the-art methods are of the form

$$\begin{aligned} & \arg \min_{\substack{f(t, \cdot) \in X \\ \theta_t \in \Theta}} \left\{ \int_0^T \left[\mathcal{L} \left(\mathcal{A} \left(t, f(t, \cdot) \right), g(t, \cdot) \right) + \mathcal{T}_\tau(t, \theta_t) + \mathcal{S}_\gamma(f(t, \cdot)) \right] dt \right\}. \\ & \text{s.t. } \Psi_{\theta_t}(f(t, \cdot)) = 0, \quad \text{for } t \in [0, T]. \end{aligned} \tag{13}$$

Just as for (4), one here needs to choose $\mathcal{S}_\gamma: X \rightarrow \mathbb{R}$ (spatial regularizer) and $\mathcal{T}_\tau(t, \cdot): X \rightarrow \mathbb{R}$ (temporal regularizer), whereas $\mathcal{L}: Y \times Y \rightarrow \mathbb{R}$ is derived from a statistical model for the noise in data.

In practice, the hard constrained formulation might be too restrictive, and we rather aim to solve a penalized version, where the motion constraint is incorporated as a regularizer; see section “[Motion Models Based on Partial Differential Equations](#)” for further details. Next, for data that is time discretized, the formulation in (13) reduces to a series of reconstruction and registration problems that are solved simultaneously. Practically, the optimization is usually performed in an alternating way, where first a dynamic reconstruction $f(t, \cdot)$ for $t \in [0, T]$ is obtained, followed by an update of the motion parameters $t \mapsto \theta_t$. This alternating minimization procedure is then iterated until a convergence criterion is fulfilled (Burger et al. 2018). Interpreted in a Bayesian setting, this approach compares to smoothing (Burger et al. 2017).

Reconstruction Using a Deformable Template

The idea here is that when solving (1), the temporal model for $t \mapsto f(t, \cdot) \in X$ is given by deforming a fixed (time-independent) template $f_0 \in X$ using a time-dependent parametrization of a deformation operator.

Deformation Operators

To formalize the underlying assumption in reconstruction with a deformable template, we assume there is a fixed family $\{\mathcal{W}_\theta\}_{\theta \in \Theta}$ of mappings (deformation operators)

$$\mathcal{W}_\theta: X \rightarrow X \quad \text{for } \theta \in \Theta. \tag{14}$$

Next, we assume that

$$f(t, \cdot) = \mathcal{W}_{\theta_t}(f_0) \quad \text{on } \Omega \text{ whenever } t \mapsto f(t, \cdot) \text{ solves (1),} \tag{15}$$

for some $t \mapsto \theta_t \in \Theta$ and $f_0 \in X$. Then, (1) can be rephrased as the inverse problem of recovering $f_0 \in X$ and $t \mapsto \theta_t \in \Theta$ from time series data $g(t, \cdot) \in Y$ where

$$g(t, \cdot) = \mathcal{A} \left(t, \mathcal{W}_{\theta_t}(f_0) \right) + e(t, \cdot) \quad \text{on } M \text{ for } t \in [0, T]. \tag{16}$$

The assumption in (15) may act as a regularization since it introduces a model for how images vary across time. In particular, the inverse problem in (16) is challenging but still easier to handle than the one in (1). However, solving (16) will still most likely require regularization. Variational approaches are suitable for this purpose, but these typically involve optimization over the parameter set Θ so it is desirable to ensure Θ has a vector space structure. Section “[Deformable Templates Given by Diffeomorphisms](#)” surveys different approaches for solving (16) based on the setting where the deformation operator is a diffeomorphic deformation.

Remark 3. Comparing assumption (15) with (9), we see that they are equivalent if

$$\Psi(t, \mathcal{W}_{\theta_t}(f_0)) = 0 \quad \text{holds on } \Omega \text{ for } t \in [0, T].$$

Hence, it is sometimes possible to view a motion model as deforming a template using a deformation operator with time-dependent parametrization. Likewise, a deformation operator with a time-dependent deformation acting on a template gives rise to a motion model.

General Variational Formulation

Following Chen et al. (2019), a variational approach for solving (16) can be formulated as

$$\arg \min_{\substack{f_0 \in X \\ t \mapsto \theta_t \in \Theta}} \left\{ \int_0^T \left[\mathcal{L} \left(\mathcal{A}(t, \mathcal{W}_{\theta_t}(f_0)), g(t, \cdot) \right) + \mathcal{J}_\tau(t, \theta_t) + \mathcal{S}_\gamma(\mathcal{W}_{\theta_t}(f_0)) \right] dt \right\}. \quad (17)$$

This is very similar to (4) with $\mathcal{L}: Y \times Y \rightarrow \mathbb{R}$ denoting the data fidelity term and the regularization term are a sum of a spatial and temporal regularizer:

$$\mathcal{S}_\gamma: X \rightarrow \mathbb{R} \quad \text{and} \quad \mathcal{J}_\tau(t, \cdot): \Theta \rightarrow \mathbb{R}.$$

The choice of the spatial regularizer \mathcal{S}_γ is a well-explored topic as outlined in section “[Reconstruction Without Explicit Temporal Models](#)”. In contrast, how to choose an appropriate temporal regularizer \mathcal{J}_τ is less explored and closely linked to assumptions on $t \mapsto \theta_t$, which governs the time evolution of the image; see, e.g., section “[Spatiotemporal Reconstruction with LDDMM](#)” for an example.

Time Discretized Data

There are different strategies for solving (16) when data is time discretized. They differ depending on how the time discretized version is formulated and in particular on how the initial template f_0 is used for building up the images f_j by means of a deformable template model.

Independent trajectory: The time discretized version of (16) is formulated as the task of recovering $f_0 \in X$ and $\theta_j \in \Theta$ from data $g_j \in Y$ where

$$g_j = \mathcal{A}_j (W_{\theta_j}(f_0)) + e_j \quad \text{for } j = 1, \dots, n. \quad (18)$$

In the above, $W_{\theta_j}: X \rightarrow X$ registers the initial template image $f_0 \in X$ against a target image $f_j \in X$ that is indirectly observed through data $g_j \in Y$. In particular, the trajectory $t \mapsto f(t, \cdot)$ is made up of images $f(t_j, \cdot) = f_j := W_{\theta_j}(f_0)$ that are generated independently from each other by deforming the initial template f_0 .

One approach for solving (18) is to compute $\widehat{f}_j := W_{\widehat{\theta}_j}(\widehat{f}_0)$ where

$$\left(\widehat{f}_0, \widehat{\theta}_1, \dots, \widehat{\theta}_n \right) \in \arg \min_{\substack{f_0 \in X \\ \theta_1, \dots, \theta_n \in \Theta}} \left\{ \sum_{j=1}^n \left[\mathcal{L} \left(\mathcal{A}_j (W_{\theta_j}(f_0)), g_j \right) + \mathcal{T}_\tau(\theta_j) + \mathcal{S}_\gamma (W_{\theta_j}(f_0)) \right] \right\}. \quad (19)$$

Note that the choice of $\mathcal{T}: \Theta \rightarrow \mathbb{R}$ may introduce a dependency between \widehat{f}_j and \widehat{f}_k for $j \neq k$ even though f_j and f_k only depend on each other through the template f_0 .

Single trajectory: Here the template f_0 is only used once to generate the image at t_1 ; the sequence of images at t_2, \dots, t_n that make up the trajectory $t \mapsto f(t, \cdot)$ are generated sequentially. The time discretized version of (16) now reduces to the task of recovering $f_0 \in X$ and $\theta_j \in \Theta$ from data $g_j \in Y$ where

$$g_j = \mathcal{A}_j (W_{\theta_j}(f_{j-1})) + e_j \quad \text{for } j = 1, \dots, n. \quad (20)$$

In contrast to (18), $W_{\theta_j}: X \rightarrow X$ is used here to deform $f_{j-1} \in X$ (image at time step t_{j-1}) to the target image $f_j \in X$ that is indirectly observed through data $g_j \in Y$. Note that one can rewrite (20) as

$$g_j = \mathcal{A}_j ((W_{\theta_j} \circ \dots \circ W_{\theta_1})(f_0)) + e_j \quad \text{for } j = 1, \dots, n. \quad (21)$$

One can attempt at solving (20) by the following intertwined scheme:

$$\left\{ \begin{array}{l} \widehat{f}_0 = \arg \min_{f \in X} \left\{ \mathcal{L} \left(\mathcal{A}_1(f), g_1 \right) + \mathcal{G}(f) \right\} \\ \widehat{\theta}_j \in \arg \min_{\theta \in \Theta} \left\{ \mathcal{L} \left(\mathcal{A}_j (W_\theta(\widehat{f}_{j-1})), g_j \right) \right. \\ \qquad \qquad \qquad \left. + \mathcal{T}_\tau(\theta) + \mathcal{S}_\gamma (W_\theta(\widehat{f}_{j-1})) \right\} \\ \widehat{f}_j := W_{\widehat{\theta}_j}(\widehat{f}_{j-1}) \end{array} \right. \quad \text{for } j = 1, \dots, n. \quad (22)$$

Note that recursive time-stepping schemes of the above type can be related to filtering approaches in a Bayesian setting (see, for instance, Hakkarainen et al. (2019) for an application to dynamic X-ray tomography).

Motion Models Based on Partial Differential Equations

In some applications, it is reasonable to assume that the underlying motion is governed by a physical phenomena that can be described by a suitable equation, like a PDE. Such an equation can then be used to constrain the motion of the reconstructed target image. Focus here is therefore on joint reconstruction and motion estimation as formulated in (13). It has been shown that a joint approach that simultaneously recovers the image sequence and the motion offers a significant advantage over subsequently and separately applying both methods (Burger et al. 2018).

Physical Motion Constraints

A common model for motion is given by the transport equation

$$\begin{cases} \frac{\partial f}{\partial t}(t, x) + \nabla \cdot (\mathbf{v}(t, x)f(t, x)) = 0, \\ f(0, x) = f_0(x) \end{cases} \quad \text{for } x \in \Omega \text{ and } t \in [0, T]. \quad (23)$$

Here, $f(t, \cdot): \Omega \rightarrow \mathbb{R}$ is the spatiotemporal image at time t contained in X , and the velocity field $\mathbf{v}(t, x): \Omega \rightarrow \mathbb{R}^d$ models the velocity with which points at x move at time t . The motion model is then given by the underlying equation in (23), which in turn yields the motion constraint

$$\Psi_{\mathbf{v}}(f(t, \cdot)) := \frac{\partial f}{\partial t}(t, \cdot) + \nabla \cdot (\mathbf{v}(t, \cdot)f(t, \cdot)) = 0 \quad \text{on } \Omega \subset \mathbb{R}^d. \quad (24)$$

This equation is generally referred to as *continuity equation* and it assumes mass preservation. Hence, with this model, mass can only be continually transformed, and no mass can be created, destroyed, or teleported.

A more restrictive model can be directly obtained from (24) under the assumption of incompressible flows or in our context brightness constancy. We give here an alternative derivation, assuming a constant image intensity $f(t, x)$ along a trajectory $t \mapsto x(t)$ with velocity $\dot{x}(t) = \mathbf{v}(t, x)$; thus, we obtain

$$0 = \frac{df}{dt} = \frac{\partial f}{\partial t} + \sum_{i=1}^d \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} = \partial_t f + \nabla f \cdot \mathbf{v}. \quad (25)$$

This equation is also called the *optical flow constraint*, and it is a popular approach to model motion between consecutive images (Horn and Schunck 1981). In the following, we will base the motion-constrained reconstruction as formulated in (13) on the continuity equation (24), assuming either mass conservation or the stronger assumption of brightness constancy in the form of the optical flow model. For both

models, the time-dependent parametrization of the motion model is by velocity fields, i.e., the motion model is given as $\Psi_{\theta_t}(f(t, \cdot))$ where $\theta_t := \mathbf{v}(t, \cdot)$ for some sufficiently regular velocity field $\mathbf{v}: [0, T] \times \Omega \rightarrow \mathbb{R}^d$ (motion field). Henceforth, we use the notation $\Psi_{\mathbf{v}} := \Psi_{\theta_t}$.

Joint Motion Estimation and Reconstruction

A joint model for motion estimation and tomographic reconstruction can be formulated, based on the motion-constrained model in (13) and following Burger et al. (2018) and Dirks (2015), for $p \in \{1, 2\}$ and $q, r > 1$ as

$$\begin{aligned} \arg \min_{\substack{t \mapsto f(t, \cdot) \in X \\ t \mapsto \mathbf{v}(t, \cdot) \in V}} \int_0^T \left[\frac{1}{p} \left\| \mathcal{A}(t, f(t, \cdot)) - g(t, \cdot) \right\|_p^p + \alpha |f(t, \cdot)|_{\text{BV}}^q + \beta |\mathbf{v}(t, \cdot)|_{\text{BV}}^r \right] dt, \\ \text{s.t. } \Psi_{\mathbf{v}}(f(t, \cdot)) = 0 \text{ on } \Omega \subset \mathbb{R}^d. \end{aligned} \tag{26}$$

Here we use for both image sequence and vector field the respective total variation as a regularizer, given by the semi-norm in the space of bounded variation. Consequently, given fixed domain $\Omega \subset \mathbb{R}^d$, the spaces under consideration here are $X = \text{BV}(\Omega, \mathbb{R})$ for the reconstructions and $V = \text{BV}(\Omega, \mathbb{R}^d)$ for the corresponding vector field. Other models can be considered such as L^2 -regularizer for the mass conservation or other convex regularizer (see Burger et al. 2018; Dirks 2015 for details). We furthermore assume the forward operator $\mathcal{A}(t, \cdot): X \rightarrow Y$ to be a bounded linear operator to some Hilbert space Y . In particular, it can be time-dependent (Burger et al. 2017; Frerking 2016).

The motion constraint in (24) is used to describe how image sequence and vector fields are connected. From the perspective of tomographic reconstructions, the motion constraint acts as an additional temporal regularizer along the motion field \mathbf{v} . Instead of imposing the motion constraint exactly as in (26), we can also relax it and add as a least-squares term to the functional itself, cf. Burger et al. (2018).

In order to establish existence of minimizers of (26), we need to ensure appropriate weak-star compactness of sublevel sets and lower semicontinuity. We will restrict the following results here now to dimension $d = 2$. For the minimization, we consider the space

$$\begin{aligned} D := \left\{ (f, \mathbf{v}) \in L^{\min\{p,q\}}([0, T]; X) \times L^r([0, T]; V) \mid \right. \\ \left. \|\mathbf{v}\|_{\infty} \leq c_v < \infty \text{ and } \|\nabla \cdot \mathbf{v}\|_E \leq c_d \right\}, \end{aligned} \tag{27}$$

where E above denotes a Banach space continuously embedded into $L^m([0, T]; L^k(\Omega, \mathbb{R}^d))$, $k > p$, and $m > q^*$ with q^* being the Hölder conjugate

of p . We can now state an existence result for the joint model (26) that is proven in Burger et al. (2018).

Theorem 1 (Existence of minimizers to (26)). *Given a linear forward operator $\mathcal{A}(t, \cdot) : X \rightarrow Y$, $p \in \{1, 2\}$ and dimension $d = 2$, let $1 < q, r$ and*

$$\mathcal{J}(f, \mathbf{v}) := \int_0^T \left[\frac{1}{p} \left\| \mathcal{A}(t, f(t, \cdot)) - g(t, \cdot) \right\|_p^p + \alpha |f(t, \cdot)|_{BV}^q + \beta |\mathbf{v}(t, \cdot)|_{BV}^r \right] dt.$$

Furthermore, let \mathcal{A} be such that it does not eliminate constants, i.e., $\mathcal{A}(t, \mathbf{1}) \neq 0$ for all $t \in [0, 1]$. Then, there exists a minimizer of $\mathcal{J}(f, \mathbf{v})$ in the constraint set

$$S := \{(f, \mathbf{v}) \in D \mid \Psi_{\mathbf{v}}(f) = 0\} \quad \text{where } D \text{ is given as in (27).}$$

The proof for $p = 2$ follows from Dirks (2015) and Burger et al. (2018), and the case for $p = 1$ follows similar arguments as outlined in Frerking (2016). Existence for the unconstrained case is proved by incorporating the constraint as a penalty term in the functional \mathcal{J} as shown in Burger et al. (2018). We note here that the choice $q, r > 1$ has to be made in the analysis in order to avoid dealing with measures in time. In the computational use cases considered below, it is however reasonable to set $q = r = 1$.

Implementation and Reconstruction

For computational reasons, as well as to allow slight deviations from the motion model, it is advantageous to consider a penalized version instead of the constrained formulation (26). Then the joint minimization problem for spatiotemporal reconstructions can be written as (Burger et al. 2017, 2018)

$$\begin{aligned} \arg \min_{\substack{t \mapsto f(t, \cdot) \in X \\ t \mapsto \mathbf{v}(t, \cdot) \in V}} \int_0^T \left[\frac{1}{p} \left\| \mathcal{A}(t, f(t, \cdot)) - g(t, \cdot) \right\|_p^p \right. \\ \left. + \alpha |f(t, \cdot)|_{BV} + \gamma \left\| \Psi_{\mathbf{v}}(f(t, \cdot)) \right\|_1 + \beta |\mathbf{v}(t, \cdot)|_{BV} \right] dt, \end{aligned} \quad (28)$$

where convergence to the constrained model is given for $\gamma \rightarrow \infty$. In practice, the BV-semi-norm is replaced by the discrete isotropic total variation.

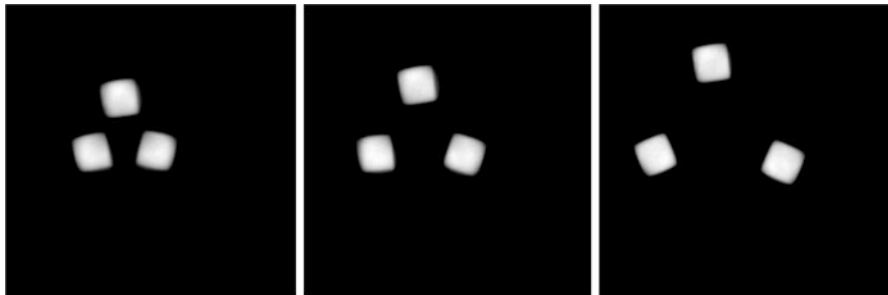
As the penalized formulation depends on the motion model $\Psi_{\mathbf{v}}(f)$, the energy to be minimized is nonlinear and therefore non-convex. Additionally, it is non-differentiable due to the involved L^1 -norms, and hence the computation of a solution to (28) is numerically challenging. Thus, in practice, it is advised to compute solutions using an intertwined scheme, which means that we split the joint model into two alternating optimization problems, one for f and the other for \mathbf{v} :

$$f^{k+1} = \arg \min_{t \mapsto f(t, \cdot) \in X} \int_0^T \left[\frac{1}{p} \|\mathcal{A}(t, f) - g\|_p^p + \alpha |f|_{\text{BV}} + \gamma \|\Psi_{\mathbf{v}^k}(f)\|_1 \right] dt \tag{29}$$

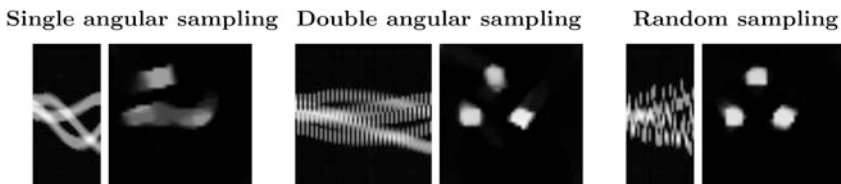
$$\mathbf{v}^{k+1} = \arg \min_{t \mapsto \mathbf{v}(t, \cdot) \in V} \int_0^T \left[\|\Psi_{\mathbf{v}}(f^{k+1})\|_1 + \frac{\beta}{\gamma} |\mathbf{v}|_{\text{BV}} \right] dt. \tag{30}$$

Most importantly, both subproblems are now linear and convex, but we note that the solution of the alternating scheme might correspond to local minima of the joint model. In practice, one would initialize $f^0 = 0$ and $\mathbf{v} = \mathbf{0}$, and then the first minimization problem for f^1 corresponds to a classic total variation regularized solution for each image time instance separately followed by a motion estimation. Reconstructions from Burger et al. (2017) using this alternating scheme for experimental μCT data are shown in Fig. 1 and an illustration of the influence of L^p -norms in the data fidelity in Fig. 2.

One can use any optimization algorithm that supports non-differentiable terms for computing solutions to each of the subproblems (29) and (30). In dimension $d = 2$, one could simply use a primal-dual hybrid gradient scheme (Chambolle and Pock 2011) as outlined in Burger et al. (2017) (see also Aviles-Rivero et al. 2018);



Ground truth spatiotemporal image at three time steps 7, 18, 25 out of 30.



Reconstructions and data from two consecutive angular sampling schemes with one and two source-detector pairs (left and middle) and a sampling scheme with only one measurement at each time instance from a randomly (uniformly) chosen direction (right). The data over time is shown to the left and reconstructions for time point 18 are shown to the right.

Fig. 1 Reconstructions from Burger et al. (2017) of experimental X-ray data using the approach in (28) with an optical flow constraint. Top row shows the ground-truth spatiotemporal image, and bottom row shows data and reconstruction for three sampling schemes

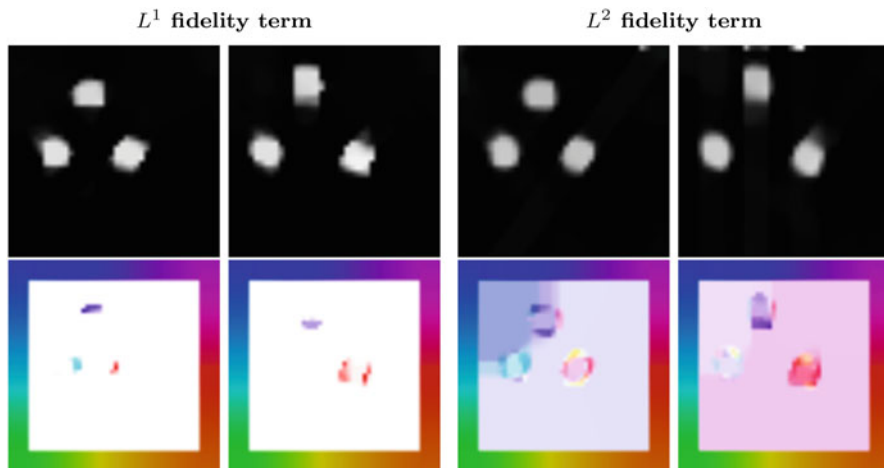


Fig. 2 Reconstruction results for the random sampling with both $p = 1, 2$ for the fidelity term in (28) for time points 17 and 25. The left images show that L^1 -norm clearly favors sparse reconstructions with a resulting sparse motion field. In contrast, the L^2 -norm shown in the right favors smoother reconstructions and motion fields

here, both applications use the optical flow constraint (25). In higher dimensions where the computational burden of the forward operator becomes more prevalent, it is advised to consider other schemes with fewer operator evaluations, and we refer to Lucka et al. (2018) for an application to dynamic 3D photoacoustic tomography as well as Djurabekova et al. (2019) for dynamic 3D computed tomography.

To conclude this section, we mention that in other applications, it might be more suitable to require mass conservation using the continuity equation instead (see, for instance, Lang et al. 2019a).

Deformable Templates Given by Diffeomorphisms

The reconstruction methods described here aim to solve (16) using deformable templates (section “[Reconstruction Using a Deformable Template](#)”).

Images are elements in the Hilbert space $X := L^2(\Omega, \mathbb{R})$ for some fixed bounded domain $\Omega \subset \mathbb{R}^d$. The deformation operator is given by acting with diffeomorphisms on images. Hence, let $\text{Diff}(\Omega)$ denote the group of diffeomorphisms (with composition as group law), and $(\phi, f_0) \mapsto \phi \cdot f_0$ denotes the (group) action of $\text{Diff}(\Omega)$ on X . In imaging, there are now two natural options:

Geometric group action: This group action simply moves image intensities without changing their gray scale values, which correspond to shape deformation:

$$\phi \cdot f_0 := f_0 \circ \phi^{-1} \quad \text{for } \phi \in \text{Diff}(\Omega) \text{ and } f_0 \in X. \quad (31)$$

Mass-preserving group action: Image intensities are allowed to change, but one preserves the total mass:

$$\phi \cdot f_0 := |D\phi^{-1}| (f_0 \circ \phi^{-1}) \quad \text{for } \phi \in \text{Diff}(\Omega) \text{ and } f_0 \in X. \tag{32}$$

The second key component is to describe how the deformation operator is parametrized, which here becomes a parametrization of the (sub)group of diffeomorphisms that are of interest. Much of the theory is motivated by image registration, and registration can in this setting be formulated as an optimization over Θ , so the chosen parametrization is preferably an element in a vector space Θ .

Flow of Diffeomorphisms and Intensities

The starting point in the LDDMM framework for image registration is to parametrize diffeomorphisms by a suitable Banach/Hilbert space of vector fields $\Theta = V \subset C_0^1(\Omega, \mathbb{R}^d)$. Diffeomorphisms in this parametrized family G_V are obtained by solving a flow equation (33) that is parametrized by a vector field in $\Theta = V$.

To more precisely define G_V , we consider solutions to the flow equation below for a given velocity field $\mathbf{v}: [0, T] \times \Omega \rightarrow \Omega$:

$$\begin{cases} \frac{d}{dt}\phi(t, x) = \mathbf{v}(t, \phi(t, x)) \\ \phi(0, x) = x \end{cases} \quad \text{for } x \in \Omega \text{ and } t \in [0, T]. \tag{33}$$

Next, let $L^1([0, T], V)$ denote the vector space of mappings $\mathbf{v}: [0, T] \times \Omega \rightarrow \mathbb{R}^d$ (velocity fields) where $\mathbf{v}(t, \cdot) \in V$. If V is *admissible*, then (33) has diffeomorphic solutions at any time $0 \leq t \leq 1$ whenever $\mathbf{v} \in L^1([0, T], V)$ (Younes 2019, Theorem 7.11 and Arguillere et al. 2015). Then, we can define $\phi_{s,t}^{\mathbf{v}}: \mathbb{R}^d \rightarrow \mathbb{R}^d$ as

$$\phi_{s,t}^{\mathbf{v}} := \phi(t, \cdot) \circ \phi(s, \cdot)^{-1} \quad \text{for } s, t \in [0, T] \text{ and } \phi(t, \cdot) \text{ solving (33)}. \tag{34}$$

This is a diffeomorphism for any $0 \leq s, t \leq 1$, so G_V defined below becomes a subgroup of diffeomorphisms parametrized by V :

$$G_V := \left\{ \phi: \mathbb{R}^d \rightarrow \mathbb{R}^d : \phi = \phi_{0,T}^{\mathbf{v}} \text{ for some } \mathbf{v} \in L^1([0, T], V) \right\}. \tag{35}$$

Remark 4. G_V is actually a subgroup of $\text{Diff}_0^{1,\infty}(\Omega)$ (Younes 2019, Theorem 7.16) where $\text{Diff}_0^{p,\infty}(\Omega)$ is the group of p -diffeomorphisms that tend to the identity at infinity:

$$\text{Diff}_0^{p,\infty}(\Omega) := \{\phi \in \text{Diff}^{p,\infty}(\Omega) : \phi - \text{Id} \in C_0^p(\Omega, \mathbb{R}^d)\}.$$

Next, if V is embedded in $C_0^p(\Omega, \mathbb{R}^d)$, then G_V is a subgroup of $\text{Diff}_0^{p,\infty}(\Omega)$.

Metamorphosis (Younes 2019, Chapter 13) is an extension of LDDMM in the sense that it considers a flow equation that jointly evolves shape and intensities:

$$\begin{cases} \frac{d}{dt} I_t^{v,\zeta}(x) = \zeta(t, \phi_{0,t}^v(x)) \\ I_0^{v,\zeta}(x) = f_0(x) \\ \phi_{0,t}^v \in G_V \text{ is given by (34)} \end{cases} \quad \text{for } x \in \Omega \text{ and } t \in [0, T]. \quad (36)$$

One can show that (36) has a unique solution $t \mapsto (\phi_{0,t}^v, I_t^{v,\zeta}) \in G_V \times X$ (Trouné and Younes 2005; Charon et al. 2018), so the above construction can be used for deforming images.

Deformable Templates by Metamorphosis

The aim here is to solve (16) with time discretized data. Following Gris et al. (2020), the idea is to adopt the *independent trajectory* approach outlined in section “Time Discretized Data”, so the inverse problem can be reformulated as a sequence of indirect registration problems (18). Hence, the task reduces to recovering and matching a template f_0 independently to data g_j in the sense of joint reconstruction and registration (indirect registration). One could here consider various approaches for indirect registration (see Yang et al. 2013; Chen and Öktem 2018 for surveys), and Gris et al. (2020) uses metamorphosis for this step.

The above considerations lead to the following variational formulation:

$$(\widehat{\theta}_1, \dots, \widehat{\theta}_n) \in \arg \min_{\theta_1, \dots, \theta_n \in V \times X} \left\{ \sum_{i=1}^n \mathcal{L} \left(\mathcal{A}_j \left(\mathcal{W}_{\theta_j}(f_0) \right), g_i \right) + \lambda \|v\|_2^2 + \tau \|\zeta\|_2^2 \right\}. \quad (37)$$

The template $f_0 \in X$ and data $g_1, \dots, g_n \in Y$ are related to each other as in (2), and the deformation operator $\mathcal{W}_{\theta_j} : X \rightarrow X$, which is parametrized by $\theta_j := (v(t_j, \cdot), \zeta(t_j, \cdot)) \in V \times X$, is given by the metamorphosis framework as

$$\mathcal{W}_{\theta_j}(f_0) := \phi_{0,t_j}^v \cdot I_{t_j}^{v,\zeta} \quad \text{where } (\phi_{0,t_j}^v, I_{t_j}^{v,\zeta}) \in G_V \times X \text{ solves (36)}. \quad (38)$$

The group action in (38) is usually the geometric one in (31).

The approach taken in Gris et al. (2020) is based on solving (37) by a scheme that intertwines updates of the image with updates of the deformation parameter. The latter involves solving an indirect registration problem, and a key part of Gris

et al. (2020) is to show that indirect registration by metamorphosis has a solution (Gris et al. 2020, Proposition 4) (existence) that is continuous w.r.t. data (Gris et al. 2020, Proposition 5) (stability) and convergent (Gris et al. 2020, Proposition 6). As such, the updates of the deformation parameter by metamorphosis-based indirect registration is a well-defined regularization method in the sense of Grasmair (2010). Likewise, the updates of the image are by a variational method that defines a well-defined regularization method, so both updates of the intertwined scheme for solving (37) are by regularization methods.

Figure 3 shows results of the above method applied to (gated) 2D tomographic data with a spatiotemporal target image. We see that (37) can be used for spatiotemporal reconstruction even when (gated) data is highly undersampled and incomplete. In particular, one can recover the evolution of the target regarding both shape deformation and photometric changes. The latter manifests itself in the appearance of the white disc.

Spatiotemporal Reconstruction with LDDMM

The aim here is to solve (16) with time continuous data by a variational formulation of the type (17). Following Chen et al. (2019), $W_{\theta_t} : X \rightarrow X$ in (17) (deformation operator) is given by the LDDMM framework, so it is parametrized by $\theta_t := \mathbf{v}(t, \cdot) \in V$ for some $\mathbf{v} \in L^2([0, T], V)$ as

$$W_{\theta_t}(f_0) := \phi_{0,t}^{\mathbf{v}} \cdot f_0 \quad \text{for } f_0 \in X \text{ and } \phi_{0,t}^{\mathbf{v}} \in G_V \text{ as in (34).} \tag{39}$$

The variant of (17) considered by Chen et al. (2019) is now

$$\arg \min_{\substack{f_0 \in X \\ t \mapsto \theta_t \in L^2([0, T], V)}} \left\{ \int_0^T \left[\mathcal{L} \left(\mathcal{A} \left(t, W_{\theta_t}(f_0) \right), g(t, \cdot) \right) + \tau \int_0^t \|\theta_s\|_V^2 ds \right] dt + \mathcal{S}_\gamma(f_0) \right\}. \tag{40}$$

Note that evaluating $W_{\theta_t}(f_0)$ requires solving the ODE in (34), so (40) is an ODE constrained optimization problem.

The temporal regularizer $\mathcal{F}_\tau(t, \cdot) : V \rightarrow \mathbb{R}$ in (17) is given by

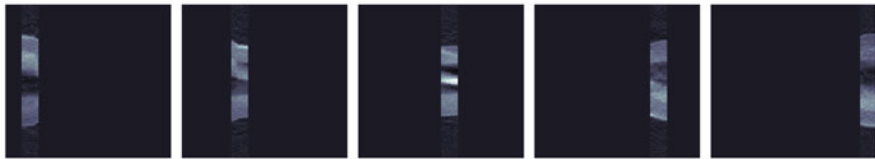
$$\mathcal{F}_\tau(t, \theta) := \tau \int_0^t \|\theta_s\|_V^2 ds \quad \text{for fixed } \tau > 0,$$

and $\mathcal{S}_\gamma : X \rightarrow \mathbb{R}$ is the spatial regularizer (typically is of Tikhonov type). In Fig. 4, we show results from Chen et al. (2019) on using (40) for spatiotemporal reconstruction in tomography.

We conclude by pointing out that the model in (40) can also be stated as PDE constrained optimal control problem as shown in Chen et al. (2019, Theorem 3.5) (see also Lang et al. 2019b). If $\theta_t = \mathbf{v}(t, \cdot) \in V$ for some velocity field $\mathbf{v} \in L^2([0, T], V)$, then (40) where the deformation operator in (39) is given by



Ground truth (unknown) 256×256 pixel grey scale spatiotemporal target image.



Gated noisy tomographic projection data of spatiotemporal target image. We sample the parallel beam ray transform at time t_i using 10 angles randomly distributed in $[(i - 1)\pi/10, i\pi/10]$. Data is corrupted with Poisson noise.

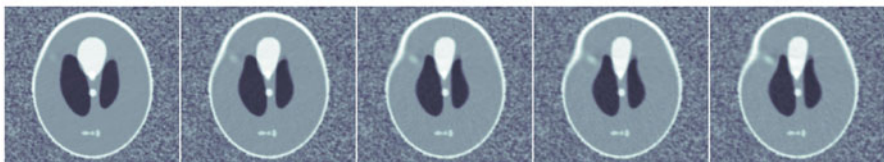


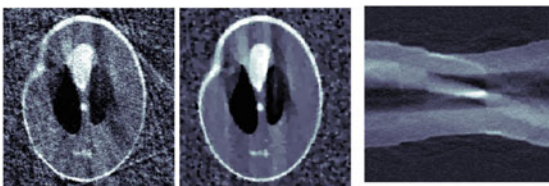
Image trajectory obtained by solving (37)



Shape trajectory obtained by solving (37)



Photometric trajectory obtained by solving (37)



Filtered back projection (left) and TV (middle) reconstructions from concatenating the 10 gated data sets (right), i.e., sampling the ray transform at 100 angles in $[0, \pi]$.

Fig. 3 (continued)

the geometric group action in (31) is equivalent to

$$\begin{aligned} \min_{\substack{f_0 \in X \\ t \mapsto \theta_t \in V}} & \left\{ \int_0^T \left[\mathcal{L} \left(\mathcal{A}_t (f(t, \cdot)), g(t, \cdot) \right) + \tau \int_0^t \|\theta_s\|_V^2 ds \right] dt + \mathcal{S}_\gamma(f_0) \right\} \\ \text{s.t. } & \partial_t f(t, \cdot) + \langle \nabla f(t, \cdot), \theta_t \rangle_{\mathbb{R}^n} = 0. \\ & f(0, \cdot) = f_0. \end{aligned}$$

In a similar manner, if the group action is the mass-preserving as in (32), then (40) becomes

$$\begin{aligned} \min_{\substack{f_0 \in X \\ t \mapsto \theta_t \in V}} & \left\{ \int_0^T \left[\mathcal{L} \left(\mathcal{A}_t (f(t, \cdot)), g(t, \cdot) \right) + \tau \int_0^t \|\theta_2\|_V^2 ds \right] dt + \mathcal{S}_\gamma(f_0) \right\} \\ \text{s.t. } & \partial_t f(t, \cdot) + \nabla \cdot (f(t, \cdot) \theta_t) = 0. \\ & f(0, \cdot) = f_0 \end{aligned}$$

This establishes the connection between ODE-based approaches discussed in this section and PDE-based approaches that are discussed in section “[Motion Models Based on Partial Differential Equations](#)”. As such, it illustrates how one can switch between a reconstruction method based on deformable templates and one based on a motion model (Remark 3).

Data-Driven Approaches

The variational approaches outlined in sections “[Reconstruction Without Explicit Temporal Models](#)”, “[Reconstruction Using a Motion Model](#)”, and “[Reconstruction Using a Deformable Template](#)” come with *two serious drawbacks* that limit their applicability. First, they typically result in complex non-convex optimization problems that are difficult to solve reasonably fast in time-critical applications. Second, they rely on a handcrafted family of parametrized temporal models that need to be computationally feasible yet are expressive enough to represent relevant temporal evolution.

Data-driven models, and especially those based on deep learning, offer means to address these drawbacks. Once trained, a deep learning model is typically very

←
Fig. 3 Spatiotemporal reconstruction using metamorphosis. Top row shows the target image we seek to recover at 5 (out of 20) selected time points in [0, 1]. Second row shows corresponding gated tomographic data. Third row shows the reconstruction of the target at these time points obtained from (37). Fourth and fifth rows show the corresponding shape and photometric trajectories. Bottom row shows reconstructions assuming a stationary target

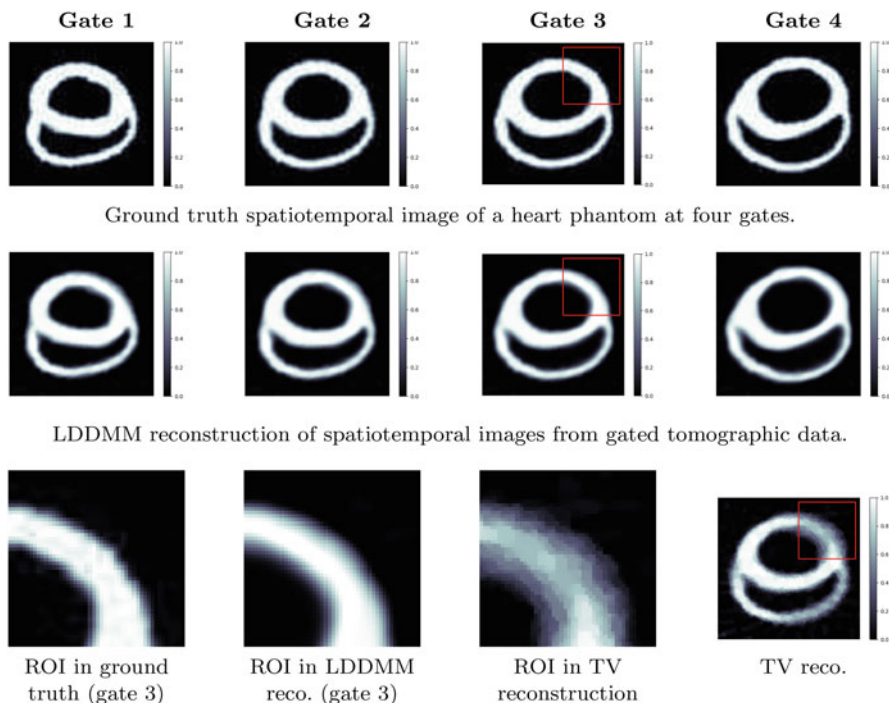


Fig. 4 Spatiotemporal reconstruction using LDDMM from gated tomographic data of a heart phantom obtained by solving (40). The heart phantom is a 120×120 pixel image with gray values in $[0, 1]$ that is taken from Grenander and Miller (2007). Data is gated 2D parallel beam tomography where the i :th gate has 20 evenly distributed directions in $[(i-1)\pi/5, \pi + (i-1)\pi/5]$. Data (not shown) also has additive Gaussian white noise corresponding to a noise level of about 14.9dB. Bottom row compares outcome at an enlarged region of interest (ROI). The ground truth (bottom leftmost image) is compared against LDDMM reconstruction (second image from left) and TV reconstruction (third image from left). The latter is computed assuming a stationary spatiotemporal target, and corresponding full image is also shown (bottom rightmost). It is clear that the cardiac wall is better resolved using a spatiotemporal reconstruction method. This is essential in CT imaging in coronary artery disease

fast to apply. Next, its large model capacity also allows for capturing complicated temporal evolution that is otherwise difficult to account for in handcrafted models. Embedding a deep learning model into a spatiotemporal reconstruction method is however far from straightforward.

Section “[Data-Driven Reconstruction Without Temporal Modelling](#)” outlines how to do this in the context of the reconstruction method in section “[Reconstruction Without Explicit Temporal Models](#)”. The situation is more complicated for reconstruction methods that use explicit temporal models. These methods rely on joint optimization of the image and the temporal model, so the latter needs to be parametrized. Embedding a deep learning-based temporal model is therefore only feasible if the said parametrization is preserved and most existing

deep learning approaches for temporal modelling of images do not fulfil this requirement. Section “[Learning Deformation Operators](#)” surveys selected deep learning models for deformations that can be embedded into reconstruction methods that use a deformable template (section “[Reconstruction Using a Deformable Template](#)”). Finally, section “[Learning Motion Models](#)” considers embedding deep learning-based models into reconstruction methods that use motion models (section “[Reconstruction Using a Motion Model](#)”).

Data-Driven Reconstruction Without Temporal Modelling

A data-driven approach for solving (1) starts by considering a family $\{\mathcal{R}_\vartheta\}_{\vartheta \in \mathcal{X}}$ of reconstruction operators $\mathcal{R}_\vartheta(t, \cdot): Y \rightarrow X$. In deep learning, \mathcal{R}_ϑ is represented by a deep neural network with network parameters ϑ . The learning amounts to finding the reconstruction operator $\mathcal{R}_{\widehat{\vartheta}}(t, \cdot): Y \rightarrow X$ where $\widehat{\vartheta} \in \mathcal{X}$ is learned from (supervised) training data as

$$\widehat{\vartheta} \in \arg \min_{\vartheta \in \mathcal{X}} L(\vartheta) \quad \text{where} \quad L(\vartheta) := \sum_{i=1}^N \int_0^T \ell_X \left(\mathcal{R}_\vartheta(t, g_i(t, \cdot)), f_i(t, \cdot) \right) dt. \quad (41)$$

Here, $\ell_X: X \times X \rightarrow \mathbb{R}$ quantifies goodness-of-fit of images, and $t \mapsto g_i(t, \cdot) \in Y$ and $t \mapsto f_i(t, \cdot) \in X$ for $i = 1, \dots, N$ represent noisy data and corresponding truth of spatiotemporal image, i.e.,

$$t \mapsto (f_i(t, \cdot), g_i(t, \cdot)) \in X \times Y \quad \text{satisfy (1) for } i = 1, \dots, N. \quad (42)$$

A key component is to specify the appropriate (deep) neural network architecture for $\mathcal{R}_\vartheta(t, \cdot): Y \rightarrow X$. One option is to set $\mathcal{R}_\vartheta := \mathcal{P}_\vartheta \circ \mathcal{A}^\dagger$ where $\mathcal{A}^\dagger(t, \cdot): Y \rightarrow X$ is a (non-learned) reconstruction operator for solving (1) and $\mathcal{P}_\vartheta(t, \cdot): X \rightarrow X$ is a data-driven post-processing operator (Hauptmann et al. 2019; Kofler et al. 2019). Hence, the input to the data-driven part is a spatiotemporal image, and the output is an “improved” spatiotemporal image. Such a model is trained against supervised data consisting of pairs of spatiotemporal images, one representing ground truth and the other the output from said reconstruction method. Alternatively, one can learn updates in an unrolled iterative scheme that is derived from some fixed point-scheme for solving (4) as in Schlemper et al. (2017). This includes a handcrafted forward operator, which in Schlemper et al. (2017) is time independent (Fourier transform), but its sampling in M depends on time. Such an approach needs supervised training data of the form (42) for its training.

Common for both approaches is that the neural network architecture does not make use of any explicit deformation/motion model. As such, they represent data-driven variants of methods outlined in section “[Reconstruction Without Explicit Temporal Models](#)”.

Learning Deformation Operators

The focus here is on using a deep learning model in a reconstruction method that uses a deformable template (section “[Reconstruction Using a Deformable Template](#)”). One possibility is to use deep learning to model the time evolution $t \mapsto \theta_t$ of the deformation parameter, which is the approach (deep diffeomorphic normalizing flow) taken in Salman et al. (2018). Another option is to use possibility in defining the parametrized deformation operator $\mathcal{W}_{\theta_t} : X \rightarrow X$ in (15). Our emphasis is on the latter, which essentially amounts to considering deep learning approaches for image registration.

There is a rich theory of variational approaches to image registration (see the books Grenander and Miller 2007, Younes 2019 and surveys in Pennec et al. 2020 and Kushnarev et al. 2020). The common trait with these approaches is that deformation models are parametrized. A variational problem is then formulated to select the “best” deformation by regularizing the deformation itself to avoid overfitting while ensuring adequate match between the template and target images. Recently, there are also many publications that consider deep learning for image registration (see Shen et al. 2017, Litjens et al. 2017, Fu et al. 2019, and Haskins et al. 2020 for surveys). Most of these learn a deformation operator directly from pairs of template and target images without accounting for any specific parametrization, i.e., the learned deformation operator is not parametrized by a deformation parameter.¹

A key aspect is that the trained deep neural network is parametrized explicitly with a (deformation) parameter, and it does not require retraining when the (deformation) parameter changes. Such a data-driven model can be used in reconstruction with deformable templates as shown in Liu et al. (2019) and Pouchol et al. (2019) for the case when data is time discretized. Both these approaches start out by stating a variational model of the type (17), which is then solved using an intertwined approach of the type (22). Here one considers diffeomorphic deformations as defined by the LDDMM framework, i.e., deformation operators are parametrized as in (47). A key part is the usage of deep learning-based deformation operators that are of the same form, i.e., the trained deep neural network retains the parametrization in (39). *In the following, our emphasis is on deep learning models for registration that adhere to a specific predefined parametrization.* Stated more precisely, one seeks to use a data-driven model for this deformation operator that belongs to a predefined parametrized family $\{\mathcal{W}_{\theta}\}_{\theta \in \Theta}$.

One way to achieve the above is by learning a mapping $\Lambda_{\vartheta} : X \times X \rightarrow \Theta$ that predicts the deformation parameter necessary for deforming a template to a target as

$$\theta := \Lambda_{\vartheta}(f_0, I) \implies \mathcal{W}_{\theta}(f_0) \approx I \quad \text{for } f_0, I \in X.$$

Note here that $\vartheta \in \mathfrak{X}$ is the deep neural network parameter that is set during training. It is *not* the same as the deformation parameter $\theta \in \Theta$, which parametrizes the

deformation operator $\mathcal{W}_\theta: X \rightarrow X$ and which is a control variable in the variational approaches for reconstruction. In some sense, Λ_ϑ can be seen as a generative model for the deformation parameter.

The mapping $\Lambda_\vartheta: X \times X \rightarrow \Theta$ can be trained in an unsupervised setting given access to sufficient amount of training data of the form

$$(I^i, f_0^i) \in X \times X \quad \text{for } i = 1, \dots, N \quad (43)$$

by computing $\widehat{\vartheta} \in \mathfrak{X}$ as

$$\widehat{\vartheta} \in \arg \min_{\vartheta \in \mathfrak{X}} L(\vartheta) \quad \text{where} \quad L(\vartheta) := \sum_{i=1}^N \ell_X \left(\mathcal{W}_{\Lambda_\vartheta(f_0^i, I^i)}(f_0^i), I^i \right). \quad (44)$$

Here, $\ell_X: X \times X \rightarrow \mathbb{R}$ is a distance notion between images, e.g., the squared L^2 -norm if $X = L^2(\Omega)$. One can also add an additional regularization term to (44) that measures registration accuracy in the image space X .

Remark 5. One can also train $\Lambda_\vartheta: X \times X \rightarrow \Theta$ in an supervised setting assuming access to training data of the form

$$(I^i, f_0^i, \theta^i) \in X \times X \times \Theta \quad \text{where } I^i \approx \mathcal{W}_{\theta^i}(f_0^i) \text{ for } i = 1, \dots, N. \quad (45)$$

The network parameter $\vartheta \in \mathfrak{X}$ is trained against the supervised data in (45) by computing $\widehat{\vartheta} \in \mathfrak{X}$ as

$$\widehat{\vartheta} \in \arg \min_{\vartheta \in \mathfrak{X}} L(\vartheta) \quad \text{where} \quad L(\vartheta) := \sum_{i=1}^N \ell_\Theta(\Lambda_\vartheta(f_0^i, I^i), \theta^i) \quad (46)$$

Here, $\ell_\Theta: \Theta \times \Theta \rightarrow \mathbb{R}$ is a distance notion between deformation parameters, so Θ must have a metric space structure. Hence, the registration accuracy is measured in the deformation parameter set Θ .

An example of this approach is Quicksilver (Yang et al. 2017), which considers deformation operators $\{\mathcal{W}_\theta\}_\theta$ given by the LDDMM framework. Then, $\theta := \mathbf{v}(1, \cdot)$ for some velocity field $\mathbf{v}: [0, 1] \times \Omega \rightarrow \mathbb{R}^d$ and

$$\mathcal{W}_\theta(f_0) := \phi_{0,1}^\mathbf{v} \cdot f_0 \quad \text{with } \phi_{0,1}^\mathbf{v} \in G_V \text{ as in (34),} \quad (47)$$

and the group action is typically geometric (31) or mass-preserving (32). It is known that the vector field $\theta \in \Theta$ that registers a template to a target can be computed by geodesic shooting (see Miller et al. 2006 and Younes 2019, Section 10.6.4). The registration problem, which is to find θ , thus reduces to finding the initial momenta. Quicksilver (Yang et al. 2017) trains a deep neural network in the unsupervised

setting (as in (44)) to learn these initial momenta. The network architecture for $\Lambda_{\vartheta} : X \times X \rightarrow \Theta$ is of convolutional neural network (CNN) type with an encoder and a decoder. The encoder acts as a feature extraction for both template and target images. The extracted features are then concatenated and fed into the decoder, which consists of three independent convolutional networks that predict the momenta for the three dimensions. To recover from prediction errors, correction networks with the same architecture are used for predicting the prediction error. Training such a deep neural network model with entire images is challenging, so Quicksilver only uses patches of images as input. In this way, relatively few images and ground-truth momenta result in a large amount of training data. A drawback is that the patches are extracted from the target, and template and deformation are on the same spatial grid locations, so the deformed patch in the target is assumed to lie (predominantly) in the same location as the one in the template image. This assumes the deformation is relatively small.

Another similar approach is VoxelMorph (Balakrishnan et al. 2019) where training is performed in an unsupervised manner (as in (44)) with only pairs of template and morphed image. The output is the displacement field $\theta \in \Theta$ necessary to register a template against a target, e.g., using an LDDMM-based deformation operator. VoxelMorph uses CNN architecture similar to U-net for $\Lambda_{\vartheta} : X \times X \rightarrow \Theta$ that consists of encoder and decoder sections with skip connections. The unsupervised loss (44) can be complemented by an auxiliary loss that leverages anatomical segmentations at training time. The trained network can also provide the registered image, i.e., it offers a deep learning-based registration operator. A further development of VoxelMorph is FAIM (Kuang and Schmah 2018) that has fewer trainable parameters (i.e., dimension of ϑ in FAIM is smaller than the one in VoxelMorph). Authors also claim that FAIM achieves higher registration accuracy than VoxelMorph, e.g., it produces deformations with many fewer “foldings,” i.e., regions of non-invertibility where the surface folds over itself.

One may also learn the spatially adaptive regularizer that is used for defining the deformation operator (Niethammer et al. 2019). See also Mussabayeva et al. (2019) for a closely related approach where one learns the regularizer in the LDDMM framework, which is the Riemannian metric for the group G_V in (35).

The above approaches all avoid learning the entire deformation; instead, they learn a deformation that belongs to a specific class of deformation models. This makes it possible to embed the learned deformation model in a variational model for image reconstruction.

Learning Motion Models

The methods mentioned here deals with using deep learning in reconstruction with a motion model (section “[Reconstruction Using a Motion Model](#)”). Many of the motion models are however sufficient for capturing the desired motion, so the main motivation with introducing deep learning is to speed up these methods.

In particular, the above means we still aim to solve the penalized variational formulation (28) with an explicit temporal model, such as the continuity equation (24). The network then essentially learns to produce the motion field $\mathbf{v}(t, \cdot)$ from the time series $f(t, \cdot)$. Such a network can then be utilized to estimate the motion field, instead of solving the corresponding subproblem (30) in the alternating minimization. For instance, one could use neural networks that are designed to compute the optical flow (Dosovitskiy et al. 2015; Ilg et al. 2017).

Another possibility is to account for the explicit structure of the PDE by using networks that aim to find a PDE representation for given data (Long et al. 2019). Alternatively, one may build network architectures based on the discretization of the underlying equations as motivated in Arridge and Hauptmann (2020). Finally, similar to the work of joint motion estimation and reconstruction, one can learn a motion map that is used in a learned reconstructions scheme (Qin et al. 2018).

Outlook and Conclusions

The variational approaches outlined in sections “[Reconstruction Using a Motion Model](#)” and “[Reconstruction Using a Deformable Template](#)”, and then in more detail in sections “[Deformable Templates Given by Diffeomorphisms](#)” and “[Motion Models Based on Partial Differential Equations](#)”, rely on explicit parametrized temporal models. These temporal models are given either by deformation operators with time-dependent parameters (section “[Reconstruction Using a Deformable Template](#)”) or through a motion model (section “[Reconstruction Using a Motion Model](#)”). Powerful techniques from analysis and differential geometry can be used to characterize regularizing properties of these reconstruction methods. They also provide state-of-the-art results when applied to challenging tomographic data that is highly noisy and/or incomplete. The methods are however difficult to use due to the computational burden and the sheer number of (regularization) parameters that needs to be chosen.

Data-driven temporal modelling offers a way to address the computational burden inherent in the variational approaches. Here, it is clear that deep learning needs to be embedded in such a way that the resulting learned temporal model is parametrized. VoxelMorph (Balakrishnan et al. 2019) and Quicksilver (Yang et al. 2017) are examples of how this can be done in the context of diffeomorphic deformation, and Liu et al. (2019) and Pouchol et al. (2019) show how such learned models can be used in reconstruction. In the near future, we expect more development along these lines. Finding appropriate training data however remains a key difficulty in data-driven approaches as in most dynamic imaging scenarios, there is no underlying ground-truth data available. Thus, most likely one will need to resort to simulations for training these models. Possibly, one could utilize reconstructions generated by variational approaches from experimental data as gold-standard reference reconstructions for a training procedure. In conclusion, there is a great need for dynamic digital phantoms that include both natural image and motion features that can serve as input for simulators.

A final challenge that applies to all reconstruction methods in dynamic inverse problems is to formulate relevant validation and comparison protocols.

Note

¹The temporal model is defined by considering a time-dependent deformation parameter. The deep neural network representing the deformation operator also has parameters, but these are not the same as the deformation parameter. In particular, the network parameters are set during training. In contrast, the deformation parameter varies with time.

References

- Arguillere, S., Trélat, E., Trouvé, A., Younes, L.: Shape deformation analysis from the optimal control viewpoint. *Journal de Mathématiques Pures et Appliquées* **104**(1), 139–178 (2015)
- Arridge, S., Hauptmann, A.: Networks for nonlinear diffusion problems in imaging. *J. Math. Imag. Vis.* **62**(3), 471–487 (2020). <https://doi.org/10.1007/s10851-019-00901-3>
- Aviles-Rivero, A.I., Williams, G., Graves, M.J., Schönlieb, C.B.: Compressed sensing plus motion (CS+M): a new perspective for improving undersampled mr image reconstruction. *ArXiv preprint 1810.10828* (2018)
- Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: VoxelMorph: a learning framework for deformable medical image registration. *IEEE Trans. Med. Imag.* **38**(8), 1788–1800 (2019)
- Beg, F.M., Miller, M.I., Trouvé, A., Younes, L.: Computing large deformation metric mappings via geodesic flow of diffeomorphisms. *Int. J. Comput. Vis.* **61**(2), 139–157 (2005)
- Benning, M., Burger, M.: Modern regularization methods for inverse problems. *Acta Numer.* **27**, 1–111 (2018)
- Bertero, M., Lantéri, H., Zanni, L.: Iterative image reconstruction: a point of view. In: Censor, Y., Jiang, M., Louis, A.K. (eds.) *Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation (IMRT)*, Pisa, pp. 37–63 (2008)
- Bubba, T.A., März, M., Purisha, Z., Lassas, M., Siltanen, S.: Shearlet-based regularization in sparse dynamic tomography. In: *Wavelets and Sparsity XVII*, vol. 10394, p. 103940Y. International Society for Optics and Photonics, Bellinghams (2017)
- Burger, M., Dirks, H., Frerking, L., Hauptmann, A., Helin, T., Siltanen, S.: A variational reconstruction method for undersampled dynamic x-ray tomography based on physical motion models. *Inverse Probl.* **33**(12), 124008 (2017)
- Burger, M., Dirks, H., Schönlieb, C.B.: A variational model for joint motion estimation and image reconstruction. *SIAM J. Imag. Sci.* **11**(1), 94–128 (2018)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.* **40**(1), 120–145 (2011)
- Charon, N., Charlier, B., Trouvé, A.: Metamorphoses of functional shapes in Sobolev spaces. *Found. Comput. Math.* **18**(6), 1535–1596 (2018). <https://doi.org/10.1007/s10208-018-9374-3>
- Chen, C., Öktem, O.: Indirect image registration with large diffeomorphic deformations. *SIAM J. Imag. Sci.* **11**(1), 575–617 (2018)
- Chen, B., Abascal, J., Soleimani, M.: Extended joint sparsity reconstruction for spatial and temporal ERT imaging. *Sensors* **18**(11), 4014 (2018)
- Chen, C., Gris, B., Öktem, O.: A new variational model for joint image reconstruction and motion estimation in spatiotemporal imaging. *SIAM J. Imag. Sci.* **12**(4), 1686–1719 (2019)
- De Schryver, T., Dierick, M., Heyndrickx, M., Van Stappen, J., Boone, M.A., Van Hoorebeke, L., Boone, M.N.: Motion compensated micro-CT reconstruction for in-situ analysis of dynamic processes. *Sci. Rep.* **8**, 7655 (10pp) (2018)

- Dirks, H.: Variational methods for joint motion estimation and image reconstruction. Phd thesis, Institute for Computational and Applied Mathematics, University of Münster (2015)
- Djurabekova, N., Goldberg, A., Hauptmann, A., Hawkes, D., Long, G., Lucka, F., Betcke, M.: Application of proximal alternating linearized minimization (PALM) and inertial PALM to dynamic 3D CT. In: 15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine, vol. 11072, p. 1107208. International Society for Optics and Photonics, Bellingham (2019)
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: FlowNet: learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision, pp. 2758–2766 (2015)
- Feng, L., Grimm, R., Block, K.T., Chandarana, H., Kim, S., Xu, J., Axel, L., Sodickson, D.K., Otazo, R.: Golden-angle radial sparse parallel MRI: combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric MRI. *Magn. Reson. Med.* **72**(3), 707–717 (2014)
- Frerking, L.: Variational methods for direct and indirect tracking in dynamic imaging. Phd thesis, Institute for Computational and Applied Mathematics, University of Münster (2016)
- Fu, Y., Lei, Y., Wang, T., Curran, W.J., Liu, T., Yang, X.: Deep learning in medical image registration: a review. ArXiv preprint 1912.12318 (2019)
- Glover, G.H.: Overview of functional magnetic resonance imaging. *Neurosurg. Clin.* **22**(2), 133–139 (2011)
- Grasmair, M.: Generalized Bregman distances and convergence rates for non-convex regularization methods. *Inverse Probl.* **26**(11), 115014 (2010)
- Grenander, U., Miller, M.: Pattern Theory. From Representation to Inference. Oxford University Press, Oxford (2007)
- Gris, B., Chen, C., Öktem, O.: Image reconstruction through metamorphosis. *Inverse Probl.* **36**(2), 025001 (27pp) (2020)
- Hakkarainen, J., Purisha, Z., Solonen, A., Siltanen, S.: Undersampled dynamic x-ray tomography with dimension reduction kalman filter. *IEEE Trans. Comput. Imag.* **5**(3), 492–501 (2019). <https://doi.org/10.1109/TCI.2019.2896527>
- Haskins G., Kruger, U., Yan, P.: Deep learning in medical image registration: a survey. *Mach. Vis. Appl.* **31**(8) (2020)
- Hauptmann, A., Arridge, S., Lucka, F., Muthurangu, V., Steeden, S.A.: Real-time cardiovascular mr with spatio-temporal artifact suppression using deep learning—proof of concept in congenital heart disease. *Magn. Reson. Med.* **81**(2), 1143–1156 (2019)
- Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artif. Intell.* **17**(1–3), 185–203 (1981)
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2462–2470 (2017)
- Kofler, A., Dewey, M., Schaeffter, T., Wald, C., Kolbitsch, C.: Spatio-temporal deep learning-based undersampling artefact reduction for 2D radial cine MRI with limited training data. *IEEE Trans. Med. Imag.* **39**(3), 703–717 (2019). <https://doi.org/10.1109/TMI.2019.2930318>
- Kuang, D., Schmah, T.: FAIM – a ConvNet method for unsupervised 3D medical image registration. ArXiv preprint 1811.09243 (2018)
- Kushnarev, S., Qiu, A., Younes, L. (eds.): Mathematics of Shapes and Applications. World Scientific, Singapore (2020)
- Kwong, Y., Mel, A.O., Wheeler, G., Troupis, J.M.: Four-dimensional computed tomography (4DCT): a review of the current status and applications. *J. Med. Imag. Radiat. Oncol.* **59**(5), 545–554 (2015)
- Lang, L.F., Dutta, N., Scarpa, E., Sanson, B., Schönlieb, C.B., Étienne, J.: Joint motion estimation and source identification using convective regularisation with an application to the analysis of laser nanoablations. *bioRxiv* 686261 (2019a)
- Lang, L.F., Neumayer, S., Öktem, O., Schönlieb, C.B.: Template-based image reconstruction from sparse tomographic data. *Appl. Math. Optim.* (2019b). <https://doi.org/10.1007/s00245-019-09573-2>

- Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. *Med. Image Anal.* **42**, 60–88 (2017)
- Liu, J., Aviles-Rivero, A.I., Ji, H., Schönlieb, C.B.: Rethinking medical image reconstruction via shape prior, going deeper and faster: deep joint indirect registration and reconstruction. To appear in *Medical Image Analysis*, preprint on arxiv 1912.07648 (2019)
- Long, Z., Lu, Y., Dong, B.: Pde-net 2.0: learning pdes from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.* **399**, 108925 (2019)
- Lucka, F., Huynh, N., Betcke, M., Zhang, E., Beard, P., Cox, B., Arridge, S.: Enhancing compressed sensing 4D photoacoustic tomography by simultaneous motion estimation. *SIAM J. Imag. Sci.* **11**(4), 2224–2253 (2018)
- Lustig, M., Santos, J.M., Donoho, D.L., Pauly, J.M.: kt SPARSE: high frame rate dynamic MRI exploiting spatio-temporal sparsity. In: 13th Annual Meeting of ISMRM, Seattle, vol. 2420 (2006)
- Miller, M.I., Trounev, A., Younes, L.: Geodesic shooting for computational anatomy. *J. Math. Imag. Vis.* **24**(2), 209–228 (2006)
- Mokso, R., Schwyn, D.A., Walker, S.M., Doube, M., Wicklein, M., Müller, T., Stampanoni, M., Taylor, G.K., Krapp, H.G.: Four-dimensional in vivo x-ray microscopy with projection-guided gating. *Sci. Rep.* **5**, 8727 (6pp) (2014)
- Mussabayeva, A., PISOV, M., Kurmukov, A., Kroshnin, A., Denisova, Y., Shen, L., Cong, S., Wang, L., Gutman, B.: Diffeomorphic metric learning and template optimization for registration-based predictive models. In: Zhu, D., Yan, J., Huang, H., Shen, L., Thompson, P.M., Westin, C.F., Pennec, X., Joshi, S., Nielsen, M., Fletcher, T., Durrleman, S., Sommer, S. (eds.) *Multimodal Brain Image Analysis and Mathematical Foundations of Computational Anatomy (MBIA 2019/MFCA 2019)*. Lecture Notes in Computer Science, vol. 11846, pp. 151–161. Springer Nature Switzerland, Cham (2019)
- Niemi, E., Lassas, M., Kallonen, A., Harhanen, L., Hämäläinen, K., Siltanen, S.: Dynamic multi-source x-ray tomography using a spacetime level set method. *J. Comput. Phys.* **291**, 218–237 (2015)
- Niethammer, M., Kwitt, R., Vialard, F.X.: Metric learning for image registration. In: *Computer Vision and Pattern Recognition (CVPR 2019)* (2019)
- Pennec, X., Sommer, S., Fletcher, T. (eds.): *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press, Cambridge (2020)
- Pouchol, C., Verdier, O., Öktem, O.: Spatiotemporal PET reconstruction using ML-EM with learned diffeomorphic deformation. In: Knoll, F., Maier, A., Rueckert, D., Ye, J.C. (eds.) *Machine Learning for Medical Image Reconstruction. Second International Workshop, MLMIR 2019, Held in Conjunction with MICCAI 2019*. Lecture Notes in Computer Science, vol. 11905, pp. 151–162. Springer (2019). Selected for oral presentation
- Qin, C., Bai, W., Schlemper, J., Petersen, S.E., Piechnik, S.K., Neubauer, S., Rueckert, D.: Joint learning of motion estimation and segmentation for cardiac mr image sequences. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 472–480. Springer (2018)
- Rahmim, A., Lodge, M.A., Karakatsanis, N.A., Panin, V.Y., Zhou, Y., McMillan, A., Cho, S., Zaidi, H., Casey, M.E., Wahl, R.L.: Dynamic whole-body PET imaging: principles, potentials and applications. *Eur. J. Nucl. Med. Mol. Imag.* **46**, 501–518 (2019)
- Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1–4), 259–268 (1992)
- Ruhlandt, A., Töpperwien, M., Krenkel, M., Mokso, R., Salditt, T.: Four dimensional material movies: high speed phase-contrast tomography by backprojection along dynamically curved paths. *Sci. Rep.* **7**, 6487 (9pp) (2017)
- Salman, H., Yadollahpour, P., Fletcher, T., Batmanghelich, K.: Deep diffeomorphic normalizing flows. *ArXiv preprint 1810.03256* (2018)
- Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: *Variational Methods in Imaging*. Applied Mathematical Sciences, vol. 167. Springer, New York (2009)

- Schlemper, J., Caballero, J., Hajnal, J.V., Price, A.N., Rueckert, D.: A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE Trans. Med. Imag.* **37**(2), 491–503 (2017)
- Schmitt, U., Louis, A.K.: Efficient algorithms for the regularization of dynamic inverse problems: I. Theory. *Inverse Probl.* **18**(3), 645 (2002)
- Schmitt, U., Louis, A.K., Wolters, C., Vauhkonen, M.: Efficient algorithms for the regularization of dynamic inverse problems: II. Applications. *Inverse Probl.* **18**(3), 659 (2002)
- Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. *Ann. Rev. Biomed. Eng.* **19**, 221–248 (2017)
- Steeden, J.A., Kowalik, G.T., Tann, O., Hughes, M., Mortensen, K.H., Muthurangu, V.: Real-time assessment of right and left ventricular volumes and function in children using high spatiotemporal resolution spiral bssfp with compressed sensing. *J. Cardiovasc. Magn. Reson.* **20**(1), 79 (2018)
- Trouvé, A., Younes, L.: Metamorphoses through Lie group action. *Found. Comput. Math.* **5**(2), 173–198 (2005)
- Trouvé, A., Younes, L.: Shape spaces. In: Otmar, S. (ed.) *Handbook of Mathematical Methods in Imaging*, pp. 1759–1817. Springer, New York (2015)
- Yang, G., Hipwell, J.H., Hawkes, D.J., Arridge, S.R.: Numerical methods for coupled reconstruction and registration in digital breast tomosynthesis. *Ann. Br. Mach. Vis. Assoc.* **2013**(9), 1–38 (2013)
- Yang, X., Kwitt, R., Styner, M., Niethammer, M.: Quicksilver: fast predictive image registration—a deep learning approach. *NeuroImage* **158**, 378–396 (2017)
- Younes, L.: *Shapes and Diffeomorphisms*. Applied Mathematical Sciences, vol. 171, 2nd edn. Springer, Heidelberg (2019)



Computational Conformal Geometric Methods for Vision

49

Na Lei, Feng Luo, Shing-Tung Yau, and Xianfeng Gu

Contents

Fundamental Concepts	1742
Riemann Surfaces	1742
Conformal Maps	1743
Uniformization	1746
Quasi-conformal Maps	1746
Holomorphic Quadratic Differential	1749
Teichmüller Map	1750
Teichmüller Space	1751
Ricci Flow	1752
Computational Methods	1754
Concepts in Discrete Setting	1755
A Discrete Conformal Geometry of Polyhedral Surfaces Derived from Vertex Scaling	1758
A Discrete Conformal Geometry of Polyhedral Surfaces Derived from Circle Patterns	1764
Harmonic Maps	1767
Hodge Decomposition	1770
Direct Applications	1772

N. Lei
Dalian University of Technology, Dalian, China
e-mail: nalei@dlut.edu.cn

F. Luo
Rutgers University, Piscataway, NJ, USA
e-mail: flu@math.rutgers.edu

S.-T. Yau
Harvard University, Cambridge, MA, USA
e-mail: yau@math.harvard.edu

X. Gu (✉)
Stony Brook University, Stony Brook, NY, USA
e-mail: gu@cs.stonybrook.edu

Shape Space	1773
Surface Registration	1780
Medical Imaging	1785
Conclusion	1788
References	1788

Abstract

Conformal geometry studies the geometric properties of objects invariant under conformal transformation group. It is a powerful theoretic tool to study shape classification, surface deformation, and registration. Computational conformal geometry is an emerging field combining modern geometry and computer science and develops both theories in the discrete setting and computational algorithms.

This work first briefly introduces the fundamental concepts, theorems in conformal geometry, such as the Riemann mapping theorem, the uniformization theorem, the Beltrami equation, Teichmüller space theory, and so on; then explains three categories of computational algorithms: discrete surface curvature flow, harmonic maps, and holomorphic differentials based on Hodge theory; and finally demonstrates practical applications in engineering and medical imaging fields. In computer vision, the work explains Teichmüller shape space for surface classification, landmark constrained surface registration based on Teichmüller map, and optimal transport map. In medical imaging, the work introduces brain mapping, brain morphology study, virtual colonoscopy, and so on.

Keywords

Conformal geometry · Ricci flow · Harmonic map · Hodge theory · Uniformization · Shape classification · Surface deformation and registration · Parameterization · Brain mapping · Colonoscopy · Graph embedding

Conformal geometry has deep roots in pure mathematics fields, such as Riemann surfaces, complex analysis, differential geometry, algebraic topology, partial differential equations, and others. Historically, conformal geometry has been broadly used in many engineering applications (Bobenko et al. 2015), such as electromagnetics, vibrating membranes, acoustics, elasticity, heat transfer, and fluid flow. Most of these applications depend on conformal mappings between planar domains.

Recently, with the rapid development of 3D scanning and medical imaging technologies, 3D geometric data has become ubiquitous. Figure 1 shows a human facial surface acquired using a scanning system based on structured light. The system can capture dynamic geometric data with very high spacial resolution and scanning speed. It is challenging to process the huge amount of this geometric data with high accuracy and efficiency. The challenge can be tackled using various geometric theories. Compared to topology or Riemannian geometry, conformal

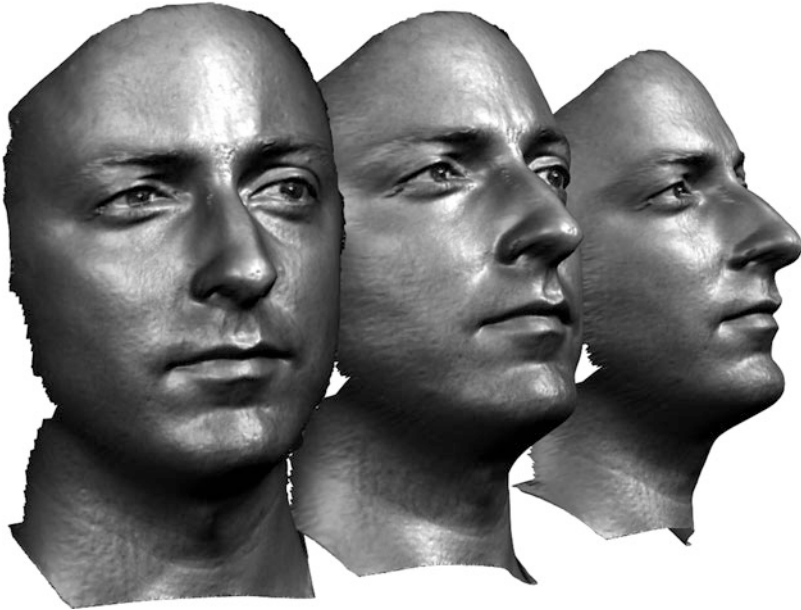


Fig. 1 3D human facial surface data scanned using structured light technology

geometry better fits this purpose because conformal structure has much richer information than topological structure and conformal mappings are much more flexible than isometries.

With the increase of computational power and further advances in mathematical theories, computational conformal geometry emerges as an interdisciplinary field, bridging mathematics and computer science. Computational conformal geometric theories and algorithms have been generalized from planar domains to surfaces with arbitrary topologies and have been applied to many engineering and medical fields. This paper is not intended to be an overview of the field and will mainly focus on our contributions to the field. Many important works have not been touched upon and many references are missing. More details can be found, for instance, in Gu and Yau (2007, 2020).

Essentially, conformal geometry focuses more on surface conformal structures and conformal mappings, which are limited. In practice, most mappings are not conformal. Fortunately, quasi-conformal geometry studies much more broad range of mappings (quasi-conformal mappings), which model most homeomorphisms in reality. From computational point of view, quasi-conformal mappings are converted to conformal ones under special metric transformations and therefore can be achieved using the same techniques in conformal geometry. In the following, we introduce the concepts, theorems, and computational methods in both conformal geometry and quasi-conformal geometry.

Conformal geometry and quasi-conformal geometry are based on Riemann surfaces; therefore they focus mainly on two dimensional manifolds, namely, surfaces. But the fundamental theorems and computational methods can be generalized to higher dimensional manifolds. For example, the surface uniformization theorem can be generalized to Thurston’s geometrization theorem for three manifolds; the discrete surface Ricci flow algorithm can be generalized to higher dimensional discrete manifolds directly.

Fundamental Concepts

In the following, we introduce the basic concepts and theorems in conformal geometry and quasi-conformal geometry.

Riemann Surfaces

The underlying spaces for two-dimensional conformal geometry are Riemann surfaces. Roughly speaking, a Riemann surface is a topological surface on which the notation of angle can be defined. More precisely, given a surface S , a *complex structure* on S is a special collection of coordinate charts $\{(U_i, \varphi_i) | i \in I\}$ such that $S = \bigcup_i U_i$ and the transition functions $\varphi_i \circ \varphi_j^{-1}$ are biholomorphic maps for all choices of indices i, j , as shown in Fig. 2. (Similarly, if all transition functions $\varphi_i \circ \varphi_j^{-1}$ are smooth, then the collection of coordinate charts is called a smooth structure.) A *Riemann surface* is a topological surface together with a complex structure. Since biholomorphic maps are orientation preserving and angle preserving, each Riemann surface is oriented, and one can naturally measure the angle between two intersecting curves on a Riemann surface. Furthermore, since the composition of a harmonic function and a holomorphic function is again harmonic,

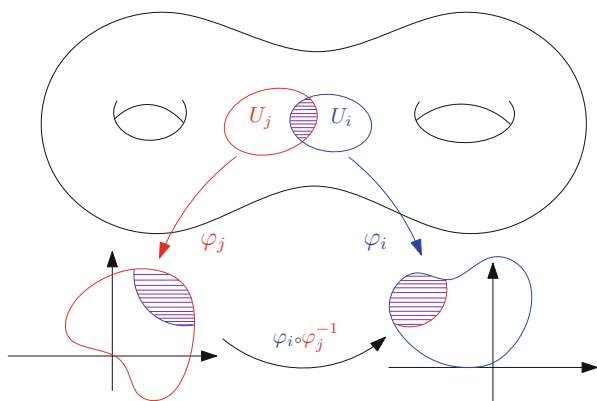


Fig. 2 Coordinate charts

the notions of harmonic functions, and more generally, harmonic and holomorphic differentials, are well defined on a Riemann surface.

Almost every surface we encounter is a Riemann surface. For instance, every open set in the plane is a Riemann surface. In fact, complex analysis that we learn in undergraduate and graduate courses is the Riemann surface theory on open sets in the plane \mathbb{C} . Furthermore, every oriented smooth surface S with a Riemannian metric \mathbf{g} is naturally a Riemann surface – the complex structure on S is induced by the Riemannian metric \mathbf{g} , and the notion of angle defined by the complex structure coincides. This was first observed by C. F. Gauss for the case of real analytic Riemannian metrics. He showed that at each point $p \in S$, one can find a coordinate chart (U, φ) such that $\varphi : (U, \mathbf{g}) \rightarrow (\mathbb{R}^2, dx^2 + dy^2)$ is an angle-preserving smooth embedding. These coordinate charts (U, φ) are called the *isothermal coordinates*. In particular, all smooth oriented surfaces in 3-space are naturally Riemann surfaces. Another class of Riemann surfaces comes from algebraic geometry. Namely, an algebraic curve in \mathbb{C}^2 , i.e., a surface defined by a polynomial equation $p(z, w) = 0$, is naturally a Riemann surface where coordinate charts are derived from the implicit function theorem.

Conformal Maps

The natural correspondences between Riemann surfaces are those bijections that preserve angles. We call them conformal maps. From complex analysis, we know that holomorphic maps are angle preserving (away from singularities). Thus, conformal maps can be considered as generalizations of injective holomorphic maps. A prominent example of a conformal map is the stereographic map from the unit sphere to the plane.

Conformal maps can be characterized as those smooth maps which preserve infinitesimal circles. In Fig. 3, two diffeomorphisms map a female facial surface to the planar unit disk. The top row shows a conformal mapping, which maps the infinitesimal circles on the face to the infinitesimal circles on the disk. In contrast, the bottom illustrates a general diffeomorphism which maps infinitesimally ellipses to circles and vice versa. If the eccentricities of the ellipses (the ratio between the major axis and the minor axis) are uniformly bounded, then the mapping is called a *quasi-conformal map*.

Equivalently, a conformal map preserves local shapes; namely, locally it is a scaling transformation followed by a rotation, where the scaling factor varies from point to point. This is illustrated in Fig. 4. The head surface of the Michelangelo's David sculpture is conformally flattened onto a planar rectangle. The complicated curved surface becomes a planar sheet under this conformal map. From the shading, one can see that the complicated local geometric shapes, such as the eyes, ears, and curly hair, are well recognizable on the plane. We can identify the major geometric features from their planar images.

In engineering applications, the distortions of mappings are classified into two categories, angle distortion and area distortion. It is always desirable to find

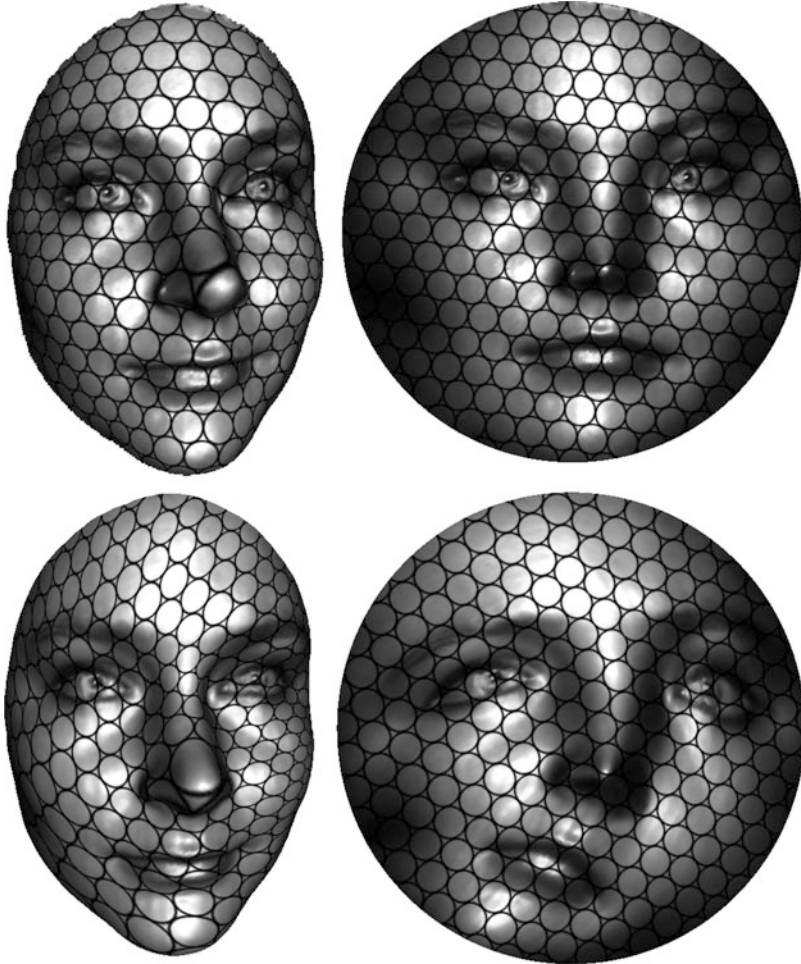


Fig. 3 Top row: conformal mapping transforms infinitesimal circles to infinitesimal circles; Bottom row: general diffeomorphism maps infinitesimal ellipses to infinitesimal circles

the optimal mappings that minimize distortions. A conformal mapping preserves angles, but distorts areas. The area distortion is called the conformal factor induced by the mapping. Depending on the topology and the geometry of the surface, the distortion of area by a conformal map could be drastic. Figure 5 compares a conformal mapping (left) and an area-preserving mapping (right) from a Buddha surface to the planar unit disk. It can be seen that the conformal mapping induces large area deformations in the head region, whereas the area-preserving mapping induces large angle deformations along the boundary of the Buddha surface. If a mapping preserves both angle and area, then it is isometric and preserves the Gaussian curvature. Hence, there doesn't exist a mapping from the Buddha surface



Fig. 4 Conformal mapping preserves local shapes

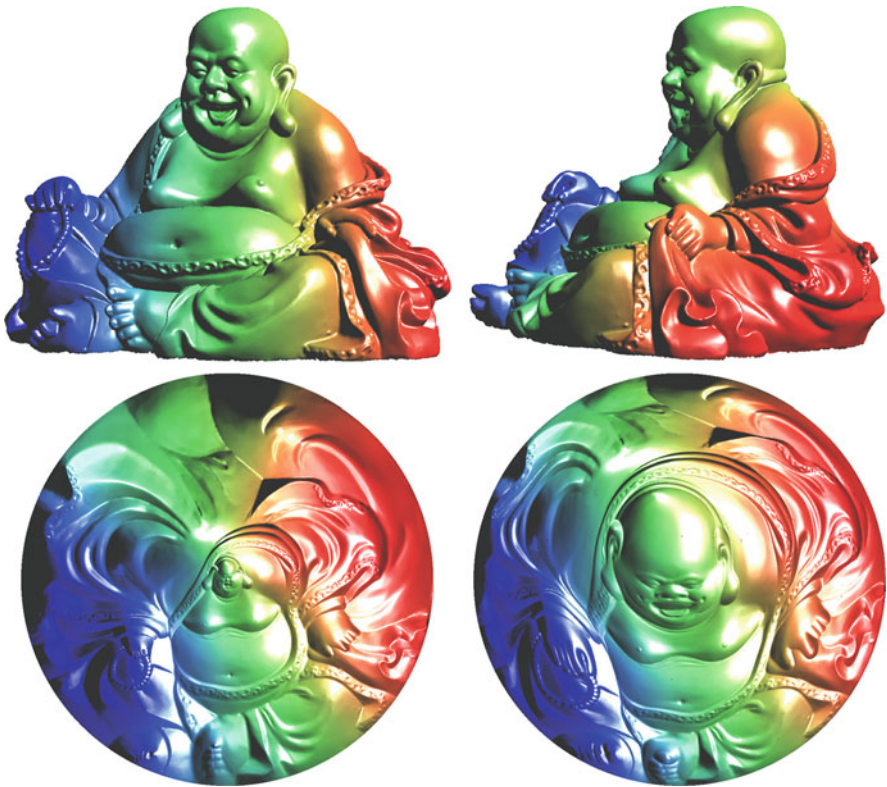


Fig. 5 Comparison between an angle-preserving mapping (left) and an area-preserving mapping (right) from a Buddha surface onto a planar disk

to the planar disk that preserves both angle and area. But it is possible to pursue either a mapping without angle distortion or a mapping without area distortion or a mapping with a good balance between angle and area distortions.

Uniformization

The famous *Riemann mapping theorem* classifies simply connected planar domains up to conformal diffeomorphism. Can one classify all connected Riemann surfaces up to conformal diffeomorphisms? This classification is achieved by the remarkable *uniformization theorem* of Poincaré and Koebe proved in 1907. It states that every simply connected Riemann surface is conformally diffeomorphic to the 2-sphere \mathbb{S}^2 , the plane \mathbb{E}^2 , or the open unit disc \mathbb{H}^2 , as shown in Fig. 6. Using covering space theory, the uniformization theorem implies that every connected oriented surface with a Riemannian metric (S, \mathbf{g}) is conformally diffeomorphic to one of three canonical models of surfaces: (i) the unit sphere \mathbb{S}^2 ; (ii) a flat torus \mathbb{E}^2/Γ , or \mathbb{E}^2 , or $\mathbb{E}^2 - \{0\}$; or (iii) a hyperbolic surface \mathbb{H}^2/Γ where Γ is a discrete torsion-free subgroup of isometries of the hyperbolic plane \mathbb{H}^2 . Equivalently, the uniformization theorem states that for any connected Riemannian surface (S, \mathbf{g}) there exists a real-valued function, $\lambda : S \rightarrow \mathbb{R}$, such that the conformal Riemannian metric $e^\lambda \mathbf{g}$ is a complete Riemannian metric of constant Gaussian curvature 1, 0, or -1 . The three curvatures correspond to the three cases (i), (ii), and (iii) above. The uniformization theorem also holds for compact surfaces with boundaries. As shown in Fig. 7, Riemannian metric surfaces with boundaries can be conformally mapped to the canonical surfaces with constant curvatures with a finite number of geodesic disks removed. We remark that there is still a famous open problem on conformal classification of planar domains. In 1910, P. Koebe conjectured that every connected open set in the plane is conformally diffeomorphic to a new domain whose boundary components are either round circles or points.

The uniformization theorem plays a fundamental role for applications in engineering and medical imaging. It sorts all kinds of shapes in the real physical world to only three canonical types. If one can develop an algorithm that can handle the canonical type surfaces, then the algorithm can process all shapes via uniformization. This greatly simplifies the algorithmic design task for engineers.

Quasi-conformal Maps

Suppose Ω is an open set in the plane and $\varphi : \Omega \rightarrow \mathbb{C}$ is a C^1 diffeomorphism of the unit disk on the complex plane, the *Beltrami coefficient* μ of φ is given by

$$\frac{\partial \varphi(z)}{\partial \bar{z}} = \mu(z) \frac{\partial \varphi(z)}{\partial z}, \quad (1)$$

where $\partial_z = 1/2(\partial_x - i\partial_y)$ and $\partial_{\bar{z}} = 1/2(\partial_x + i\partial_y)$. The *dilatation* of φ is defined as

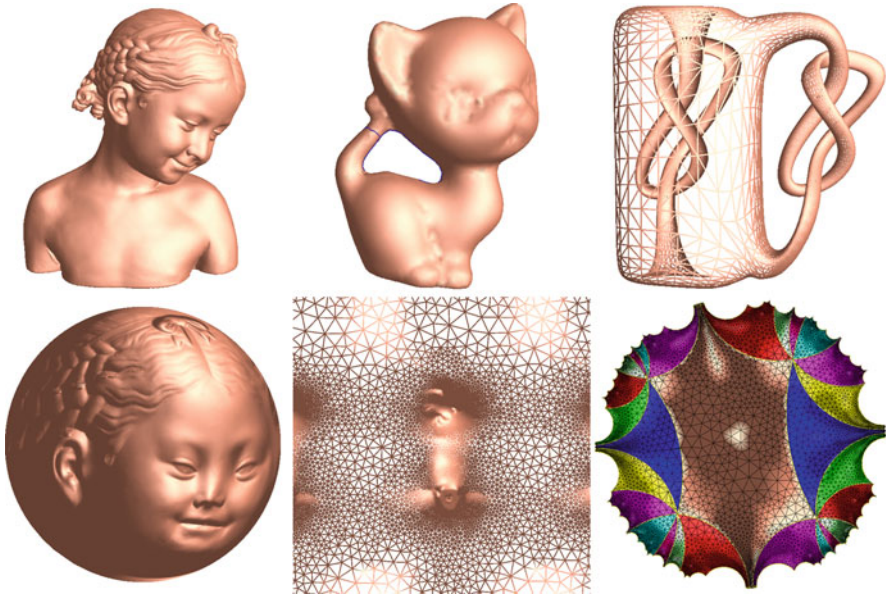


Fig. 6 Uniformization for closed surfaces

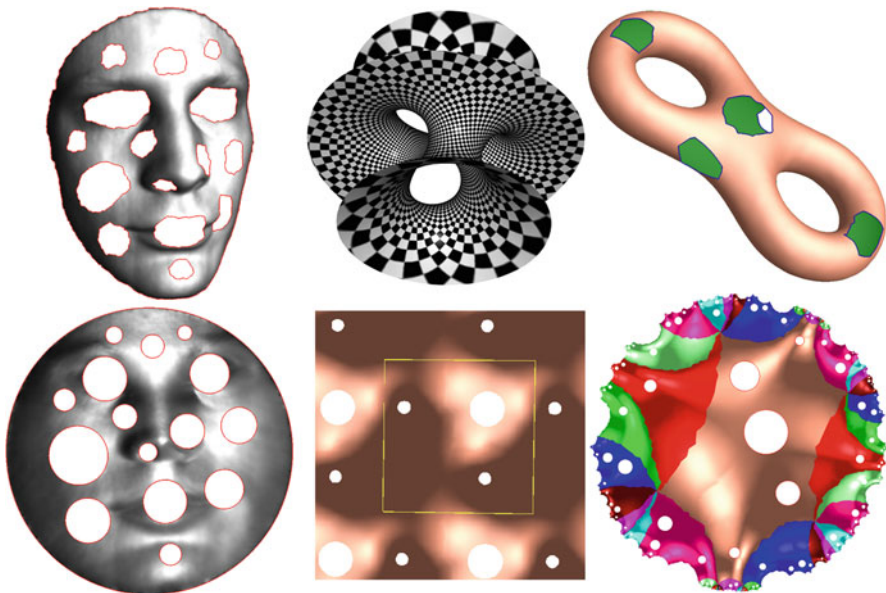


Fig. 7 Uniformization for surfaces with boundaries

$$K_\varphi = \frac{1 + |\mu_\varphi|}{1 - |\mu_\varphi|}. \tag{2}$$

The map φ is said to be *quasi-conformal* if K_φ is bounded and called K -quasi-conformal if $K_\varphi \leq K$ in the domain Ω . Note that the map φ is conformal if μ_φ is zero everywhere. Geometrically, a quasi-conformal map φ transforms infinitesimal circles to infinitesimal ellipses. The eccentricity of the ellipse at $\varphi(z)$ is given by K_φ , and the angle between the major axis of the ellipse and the real axis is given by $1/2 \arg \mu(z)$. We define the *maximal dilatation* of φ as

$$K(\varphi) := \frac{1 + \|\mu_\varphi\|_\infty}{1 - \|\mu_\varphi\|_\infty}. \tag{3}$$

Equation 1 is called the *Beltrami equation*. An important theorem says that given the Beltrami coefficient μ , one can solve the Beltrami equation Eq. 1 in φ . More precisely, the *measurable Riemann mapping theorem* says that given a measurable complex function $\mu : \mathbb{D} \rightarrow \mathbb{C}$, such that $\|\mu\|_\infty < 1$, then there exists a quasi-conformal homeomorphism $\varphi : \mathbb{D} \rightarrow \mathbb{D}$ satisfying the Beltrami equation Eq. 1. Furthermore, two such solutions differ by a Möbius transformation,

$$z \mapsto e^{i\theta} \frac{z - z_0}{1 - \bar{z}_0 z}, \quad \theta \in [0, 2\pi), |z_0| < 1. \tag{4}$$

All the Möbius transformations of the disk \mathbb{D} form a *three-dimensional* group as shown in Fig. 8. The measurable Riemann mapping theorem establishes the relationship among quasi-conformal homeomorphism group of the disk, the space of Beltrami coefficients, and the Möbius transformation group:

$$\{\text{Quasi-conformal Homeomorphisms}\} \cong \frac{\{\text{Beltrami Coefficients } \|\mu\|_\infty < 1\}}{\{\text{Möbius Transforms}\}}$$



Fig. 8 Möbius transformation

Given a diffeomorphism between two Riemann surfaces $\varphi : (S_1, \{z_i\}) \rightarrow (S_2, \{w_j\})$, the *Beltrami differential* $\mu(z_i)d\bar{z}_i/dz_i$ is a tensor on S_1 defined by

$$\frac{\partial w_j}{\partial \bar{z}_i} d\bar{z}_i = \mu(z_i) \frac{\partial w_j}{\partial z_i} dz_i.$$

Note that the definition shows $\mu(z_i)d\bar{z}_i/dz_i$ is invariant under the coordinate transitions and thus is globally defined. The K -quasi-conformal map and its associated Beltrami differential can be generalized to the Riemann surface cases directly. For instance, any C^1 -smooth diffeomorphism between two compact Riemann surfaces is a quasi-conformal map.

Holomorphic Quadratic Differential

Given a Riemann surface S , a *holomorphic k -differential* is a tensor which assigns a holomorphic function $\varphi_i(z_i)$ to each local chart z_i , such that if z_j is another local coordinate, then we have

$$\varphi_i(z_i) = \varphi_j(z_j) \left(\frac{dz_j}{dz_i} \right)^k.$$

The complex linear space of all the holomorphic one-forms on a closed Riemann surface of genus g is g dimensional. The space is isomorphic to the cohomology group of the surface $H^1(S, \mathbb{R})$. For a quadratic differential $\omega = \phi(z)dz^2$ on a Riemann surface S , its L^1 norm or its *area* is

$$\|\omega\|_{L^1} = \int_S |\omega| = \int_S |\phi(z)| |dz|^2.$$

All integrable holomorphic quadratic differentials on the Riemann surface S are denoted as $\Omega(S)$. By the Riemann-Roch theorem, for a closed Riemann surface of genus g , $\Omega(S)$ is a $3g - 3$ -dimensional complex linear space.

For example, suppose S is a sphere with n punctures,

$$S = \mathbb{C} \cup \{\infty\} - \{a_1, a_2, \dots, a_n\},$$

then every integrable holomorphic quadratic differential has the form $\varphi(z)dz^2$, where

$$\varphi(z) = \sum_{i=1}^n \frac{\rho_k}{z - a_k},$$

such that



Fig. 9 Horizontal trajectories of a holomorphic quadratic differential on a cat surface

$$\sum_{k=1}^n \rho_k = 0, \quad \sum_{i=1}^n \rho_k a_k = 0, \quad \sum_{i=1}^n \rho_k a_k^2 = 0.$$

Suppose φ is a holomorphic quadratic differential, a point $p \in S$ is called a *zero point* of φ , if $\varphi_i(p)$ equals to zero. Given a holomorphic quadratic differential φ on a closed genus g surface, there are $4g - 4$ zero points. The local behavior of a quadratic differential can be well understood. Take a nonzero point $p \in S$ of φ and a small coordinate chart (U, z) at p . There is a holomorphic one-form, denoted by $\sqrt{\varphi}$ such that $(\sqrt{\varphi})^2 = \varphi$ where $\sqrt{\varphi} = \sqrt{a(z)}dz$ and $\varphi = a(z)dz^2$. The *natural holomorphic coordinate* of φ is defined as

$$\xi(z) := \int_p^z \sqrt{\varphi}.$$

Note that in this coordinate, the quadratic form φ is $d\xi^2$. A curve γ on S is called a *horizontal trajectory* of φ if it is a horizontal line under the natural coordinates of φ . This is the same as $\phi(\gamma'(t), \gamma'(t)) \geq 0$. The *vertical trajectory* is defined in the similar way.

Figure 9 shows the horizontal trajectories of a holomorphic quadratic differential φ on a cat surface. The bifurcation points are the zero points of φ .

Teichmüller Map

Generally speaking, given two homeomorphic surfaces with Riemannian metrics, there may not be a conformal map between them. Instead, there is a map that is closest to being conformal, namely, the *Teichmüller map*. A Teichmüller map minimizes the angle distortion and has many special properties.

Given two homeomorphic Riemann surfaces S_1 and S_2 , let $f : S_1 \rightarrow S_2$ be a quasi-conformal map between them. We say f is *extremal mapping* or *Teichmüller mapping* if for any quasi-conformal map $h : S_1 \rightarrow S_2$, h is isotopic to f relative to the boundary,

$$K(f) \leq K(h),$$

i.e., extremal quasi-conformal map minimizes the angle distortion.

For closed Riemann surfaces, Teichmüller proved that the Beltrami differential of an extremal map f is of the form

$$\mu_f = k \frac{\bar{\varphi}}{|\varphi|}$$

for some $0 \leq k < 1$ and quadratic differential $\|\varphi\|_{L^1} < \infty$. The maximal dilatation of μ_f is $\|\mu_f\|_\infty = k$ which is equal to the dilatation $|\mu_f(z)|$ at each point z . This means the infinitesimal ellipses have the same eccentricity everywhere except at zeros of φ . Furthermore, it is known that if the Beltrami differential of a quasi-conformal map $f : S_1 \rightarrow S_2$ is of the form $k \frac{\bar{\varphi}}{|\varphi|}$ for some nonzero quadratic differential φ , then f is an extremal quasi-conformal map.

On the target surface S_2 , there is a corresponding holomorphic quadratic differential η , such that the Teichmüller map f maps the horizontal trajectories of φ to the horizontal trajectories of η , the vertical trajectories of φ to the vertical trajectories of η , and the zeros of φ to the zeros of η . Furthermore, suppose $x + iy$ is the natural coordinates of φ , $u + iv$ the natural coordinates of η , then the Teichmüller map f has the local representation: $x + iy \mapsto u + iv$,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1+k & 0 \\ 0 & 1-k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

which is a linear map.

Teichmüller Space

Metric surfaces with same topology can be further classified by conformal equivalence. If there is a conformal map between the two surfaces, then the surfaces are conformal equivalent. All the conformal equivalence classes form a finite dimensional manifold, the so-called Teichmüller space, which also admits a natural Riemannian metric, the Weil-Petersson metric. Therefore, we can use the Teichmüller space as the model of shape space and measure the distances among shapes.

Let S be an orientable smooth surface; the *Teichmüller space* $T(S)$ of S is the space of Riemann surface structures on S up to isotopy. More precisely, two conformal structures X and Y on S are said to be *Teichmüller equivalent*, if there is a diffeomorphism f , such that f is isotopic to the identity of S and $f : (S, X) \rightarrow (S, Y)$ is conformal. The $T(S)$ is the space of equivalence classes of conformal structures on S modulo this relation. Suppose S is a punctured Riemann surface of genus $g > 1$ surface with n punctures, then $T(S)$ is of $6g - 6 + 3n$ dimension.

Given two conformal structures $[X], [Y] \in T(S)$, there is a unique Teichmüller map between them, $f : (S, X) \rightarrow (S, Y)$. Then *Teichmüller distance* between them is given by the dilatation of f ,

$$d_{T(S)}([X], [Y]) := \frac{1}{2} \log K(f).$$

Suppose $\mu_f = k\bar{\varphi}/|\varphi|$, $0 \leq k < 1$, the *Teichmüller geodesic* connecting $[X]$ and $[Y]$ in $T(S)$ is given by $[X_t]$, $t \in [0, 1]$, and f_t is the Teichmüller map between (S, X) and (S, X_t) , then the Beltrami differential of f_t is associated with the holomorphic quadratic differential φ , $\mu_{f_t} = kt\bar{\varphi}/|\varphi|$.

Ricci Flow

Riemannian metric Suppose S is a topological surface, a *Riemannian metric* \mathbf{g} assigns an inner product to each tangent space T_pS . Locally, suppose $\mathbf{v}_1, \mathbf{v}_2 \in T_pS$, with local coordinates

$$\mathbf{v}_k = \xi_1^k \frac{\partial}{\partial x_1} + \xi_2^k \frac{\partial}{\partial x_2}, \quad k = 1, 2,$$

then

$$\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_{\mathbf{g}} = [\xi_1^1 \ \xi_2^1] \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} \xi_1^2 \\ \xi_2^2 \end{bmatrix}.$$

Here $g_{ij} = \langle \frac{\partial}{\partial x_i}, \frac{\partial}{\partial x_j} \rangle$.

Conformal Map Suppose \mathbf{g}_1 and \mathbf{g}_2 are two Riemannian metrics on S , we say they are *conformal equivalent*, if there is a function $u : S \rightarrow \mathbb{R}$, such that

$$\mathbf{g}_1(p) = e^{2u(p)} \mathbf{g}_2(p), \quad \forall p \in S.$$

Given a smooth mapping $f : (S, \mathbf{g}) \rightarrow (T, \mathbf{h})$ with local representation $(x, y) \mapsto (u, v)$, the Jacobian of the map is given by

$$DT = \begin{bmatrix} \partial u / \partial x & \partial u / \partial y \\ \partial v / \partial x & \partial v / \partial y \end{bmatrix},$$

and the *pullback* metric induced by f has local representation,

$$f^*\mathbf{h} = \begin{bmatrix} \partial u/\partial x & \partial v/\partial x \\ \partial u/\partial y & \partial v/\partial y \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \partial u/\partial x & \partial u/\partial y \\ \partial v/\partial x & \partial v/\partial y \end{bmatrix}.$$

If the pullback metric $f^*\mathbf{h}$ is conformal equivalent to the original metric \mathbf{g} , then the mapping f is conformal. This definition is consistent with the one based on conformal structure.

Isothermal Coordinates For any point $p \in S$, there is a neighborhood $U(p)$ with local coordinate (x, y) such that the metric \mathbf{g} is

$$\mathbf{g} = e^{2u(x,y)}(dx^2 + dy^2).$$

We call (x, y) an *isothermal coordinate* of (S, \mathbf{g}) at p . Given an orientable metric surface, the isothermal coordinate charts form the conformal structure of the surface. This shows all orientable metric surfaces are Riemann surfaces.

Gaussian Curvature Under the isothermal coordinate, the Gaussian curvature of the surface is given by

$$K(x, y) = -\Delta_{\mathbf{g}}u(x, y) = -\frac{1}{e^{2u(x,y)}}\Delta u(x, y) = -\frac{1}{e^{2u(x,y)}}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u(x, y).$$

Yamabe Equation Suppose $\bar{\mathbf{g}} = e^{2u}\mathbf{g}$ is a conformal metric on the surface S , then the Gaussian curvature \bar{K} of $\bar{\mathbf{g}}$ is

$$\bar{K} = e^{-2u}(-\Delta_{\mathbf{g}}u + K), \tag{5}$$

and the geodesic curvature on the surface boundary becomes

$$\bar{k}_g = e^{-u}(-\partial_{\mathbf{n}}u + k_g), \tag{6}$$

Equations 5 and 6 are called *Yamabe equations*. In engineering applications, it is highly desirable to find Riemannian metrics with prescribed curvatures, which is equivalent to solve the Yamabe equations.

Surface Ricci Flow Most geometric problems in engineering and medical applications can be reduced to find an appropriate Riemannian metric with required curvature. Surface Ricci flow is a powerful tool for this purpose. Intuitively, surface Ricci flow deforms the Riemannian metric proportional to the current curvature, such that the curvature evolves according to a diffusion-reaction process. If the diffusion component dominates, the curvature will converge to a constant. This gives us the uniformization metric. Hamilton’s surface Ricci flow is defined as follows:

$$\frac{\partial \mathbf{g}(x, t)}{\partial t} = -2K(x, t)\mathbf{g}(x, t), \quad (7)$$

and the curvature evolution equation is

$$\frac{\partial K(x, t)}{\partial t} = \Delta_{\mathbf{g}(t)}K(x, t) + K^2(x, t). \quad (8)$$

The normalized surface Ricci flow is given by

$$\frac{\partial \mathbf{g}(x, t)}{\partial t} = \left(\frac{4\pi \chi(S)}{A(0)} - 2K(x, t) \right) \mathbf{g}(x, t), \quad (9)$$

where $A(0)$ is the total surface area at time 0 and $\chi(S)$ is the Euler characteristic number of S . Surface Ricci flow deforms the Riemannian metric conformally; hence the conformal factor equation can be written down as

$$\frac{\partial u(x, t)}{\partial t} = \frac{2\pi \chi(S)}{A(0)} - K(x, t). \quad (10)$$

Normalized Ricci flow on closed surface converges to the uniformization metric.

Computational Methods

With the advances of modern technologies (digital cameras, 3D scanners, CT scanners, etc.), surfaces are produced digitally at an alarming rate these days. There is an urgent need to process and categorize them. A useful form of these digital surfaces is polyhedral surfaces. In Fig. 10 Michelangelo's David sculpture surface is

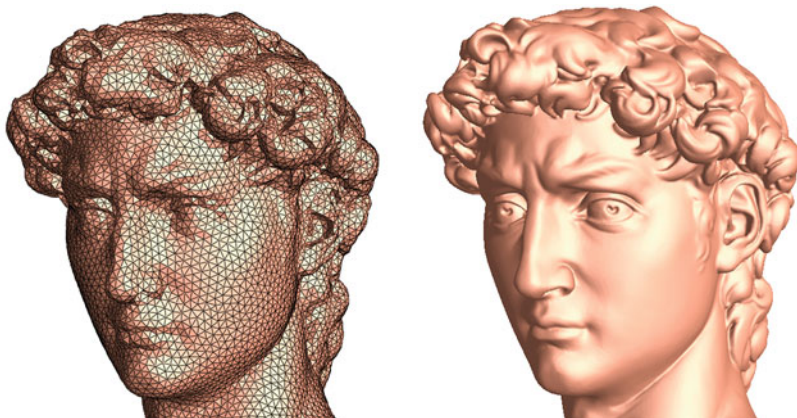


Fig. 10 Discrete representation of Michelangelo's David sculpture surface

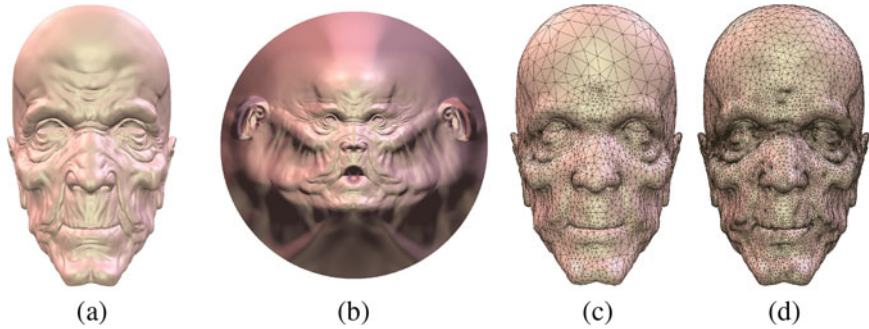


Fig. 11 Geometric approximation using Riemann mapping and normal cycle. (a) Original surface. (b) Conformal mapping. (c) 2k samples. (d) 8k samples

approximated by a polyhedral surface. Classical differential geometric theories are inadequate to deal with polyhedral and digital surfaces. It is a major challenge to develop computable theories for conformal, harmonic, quasi-conformal, isometric, area-preserving, and other mappings for polyhedral surfaces. We will present our approaches to the challenge below. There is no doubt that more discrete theories will be developed. There are several guiding principles one tries to follow in discretizing classical concepts. Firstly, the discrete counterparts should have their own intrinsic geometric structures. Secondly, there should be a finite dimensional variational principle whose critical points would correspond to the discrete entities (e.g., discrete Riemann surfaces, discrete conformal maps). As usual, a finite dimensional variational characterization will then lead to practical computational algorithms with efficiency, accuracy, and robustness. Finally, the discrete entities should converge to their smooth counterparts when the triangular meshes are suitably chosen (Fig. 11).

There are three categories of computational algorithms, as illustrated in the computation of surface uniformization (Fig. 6): harmonic maps method in the left frame, Hodge decomposition and meromorphic differential method in the middle frame, and the discrete surface Ricci/Yamabe flow method in the right frame. Different methods have different advantages and disadvantages and are able to solve different problems. None of them can be replaced by others. For example, in order to find the conformal hyperbolic metric, discrete surface curvature flow should be used; in order to compute holomorphic differentials, Hodge decomposition method should be applied and so on.

Concepts in Discrete Setting

Discrete Surface Let us begin by recalling what triangulations and polyhedral surfaces are. Take a collection of Euclidean triangles and identify pairs of edges by homeomorphisms. The quotient space is a topological surface Σ together with a

triangulation \mathcal{T} . Triangles in \mathcal{T} come from the quotients of the Euclidean triangles. If we identify pairs of edges by isometries (i.e., length-preserving homeomorphisms), we obtain a polyhedral metric, or *piecewise linear (PL) metric*, on the triangulated surface (Σ, \mathcal{T}) . A *polyhedral surface* is a surface with a PL metric. For instance, the boundary of a three-dimensional polytope in the 3-space carries a natural PL metric. Clearly a PL metric d on (Σ, \mathcal{T}) can be determined by the *edge length function* $\ell : E(\mathcal{T}) \rightarrow \mathbb{R}_{>0}$ which records the length of an edge e in the set $E(\mathcal{T})$ of all edges in \mathcal{T} . The function ℓ must only satisfy the triangle inequality, that is, if e_i, e_j, e_k form the edges of a triangle, then

$$\ell(e_i) + \ell(e_j) > \ell(e_k).$$

Therefore, a PL metric can be coded by a computer easily. Following A. D. Alexandrov, we consider a PL metric $d : \Sigma \times \Sigma \rightarrow \mathbb{R}_{\geq 0}$ as a metric in the sense of point set topology. The distance $d(x, y)$ between two points $x, y \in \Sigma$ is the infimum of the lengths of all paths on Σ joining x and y . Here the length of a path inside each triangle is computed using the Euclidean metric induced by the edge lengths of the triangle. It follows that the PL metric d is flat away from the vertices of \mathcal{T} .

Note that a triangulation \mathcal{T} is a tool and the edge length function ℓ is a “coordinate” to describe the metric d . There may be many different triangulations \mathcal{T}' and the associated length functions $\ell' : E(\mathcal{T}') \rightarrow \mathbb{R}_{>0}$ describing the same PL metric d . One of the goals is to develop a computable discrete counterpart of conformal geometry which is independent of the choice of triangulations. For instance, the discrete curvature $K_d(v)$ is independent of the choice of triangulations \mathcal{T} .

Cosine Laws In general, we can isometrically glue Euclidean, spherical, or hyperbolic triangles to construct a discrete surface (Σ, \mathcal{T}) and call the surface is with Euclidean, spherical or hyperbolic *background geometry*. As shown in Fig. 12, given a triangular face formed by vertices $v_i, v_j,$ and $v_k,$ the corner angle at v_i is denoted as $\theta_i,$ and the length of the edge against v_i is $\ell_i.$ Then the corner angles are determined by the edge lengths via cosine laws. The Euclidean, hyperbolic, and spherical cosine laws are given by:

$$1 = \frac{\cos \theta_i + \cos \theta_j \cos \theta_k}{\sin \theta_j \sin \theta_k} \tag{11}$$

$$\ell_i^2 = \ell_j^2 + \ell_k^2 - 2\ell_j \ell_k \cos \theta_i \tag{12}$$

$$\cosh \ell_i = \frac{\cosh \theta_i + \cosh \theta_j \cosh \theta_k}{\sinh \theta_j \sinh \theta_k} \tag{13}$$

$$\cos \ell_i = \frac{\cos \theta_i + \cos \theta_j \cos \theta_k}{\sin \theta_j \sin \theta_k} \tag{14}$$

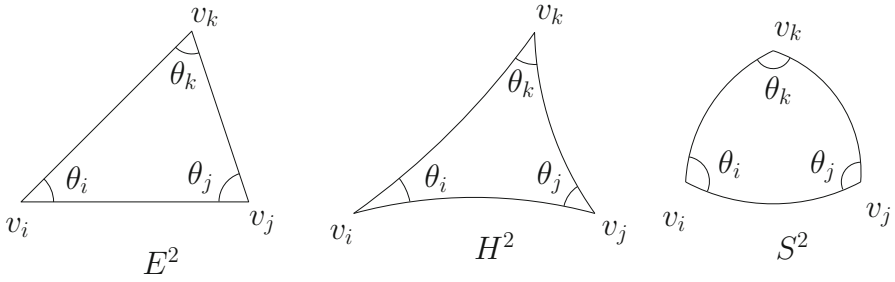


Fig. 12 Cosine laws with different background geometries

Discrete Curvature Let $V(\mathcal{T})$ be the set of all vertices in the triangulation \mathcal{T} . At each vertex $v \in V(\mathcal{T})$, the *discrete curvature* $K_d(v)$ of d is

$$K(v_i) = \begin{cases} 2\pi - \sum_{jk} \theta_i^{jk} & v_i \notin \partial \Sigma \\ \pi - \sum_{jk} \theta_i^{jk} & v_i \in \partial \Sigma \end{cases} \tag{15}$$

where θ_i^{jk} is the corner angle at v_i in the triangle face $[v_i, v_j, v_k]$, as shown in Fig. 13. The discrete curvature satisfies the Gauss-Bonnet theorem,

$$\sum_{v \in \mathcal{T}} K(v) + kA(S) = 2\pi \chi(S), \tag{16}$$

where $k = 0, +1, -1$ for Euclidean, spherical, and hyperbolic background geometries, $A(S)$ is the total area of the surface, and $\chi(S)$ is the Euler characteristic number of S .

The table below summarizes common smooth notions and their discrete counterparts.

Smooth category	Discrete category
Smooth surfaces S	Triangulated surfaces (Σ, \mathcal{T})
Functions on S	Functions on $V(\mathcal{T})$
Riemannian metric \mathbf{g}	PL metric \mathbf{d} on (Σ, \mathcal{T})
Gaussian curvature of \mathbf{g}	Discrete curvature K_d on $V(\mathcal{T})$
Conformal class $\{e^u \mathbf{g}\}$	Discrete conformal class of \mathbf{d}

Next we define the discrete conformal equivalence of PL metrics on a surface. There are now several ways to formulate it. In this paper, we will focus on two such definitions. A more general form of discrete conformal equivalences, which

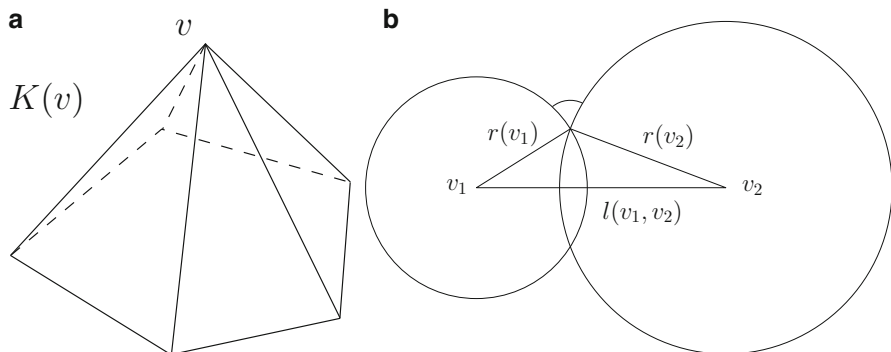


Fig. 13 Discrete curvature and circle packing

includes these two as special cases, was proposed by Glickenstein (2011). Both of these definitions were motivated by the seminal work of R. Hamilton on Ricci flows for smooth Riemannian manifolds.

A Discrete Conformal Geometry of Polyhedral Surfaces Derived from Vertex Scaling

Vertex Scaling Given two PL metrics on a triangulated surface (Σ, \mathcal{T}) whose edge length functions are ℓ and $\hat{\ell}$, we say ℓ and $\hat{\ell}$ are related by a *vertex scaling* (Luo 2004; Roček and Williams 1981), written as $\hat{\ell} = u * \ell$, if there exists a function $u : V(\mathcal{T}) \rightarrow \mathbb{R}$ such that for each edge e with end points v_1, v_2 ,

$$\hat{\ell}(e) = e^{u(v_1)+u(v_2)} \ell(e). \tag{17}$$

Equation (17) represents a discretization of the conformal Riemannian metric $e^u \mathbf{g}$. It is proved in Gu et al. (2019) that this formal analogy has a more deep connection. Indeed, if \mathbf{g} is a Riemannian metric on a compact manifold M and $u : M \rightarrow \mathbb{R}$ is a smooth function, then there exists a constant $C > 0$ such that for any pairs of points $x, y \in M$,

$$|d_{e^{4u}\mathbf{g}}(x, y) - e^{u(x)+u(y)} d_{\mathbf{g}}(x, y)| \leq C d_{\mathbf{g}}(x, y)^3.$$

Here $d_{\mathbf{g}}$ is the Riemannian distance associated with the Riemannian metric \mathbf{g} , i.e., $d_{\mathbf{g}}(x, y)$ is the infimum of the lengths of all paths joining x to y . The above estimate holds the key for showing that discrete conformal maps defined using (17) converge to the smooth case.

Variational Principle The definition of vertex scaling in Eq. (17) carries a natural variational principle relating a PL metric to its discrete curvature (Luo 2004).

Fix a Euclidean triangle $[v_i, v_j, v_k]$, with edge lengths l_i, l_j, l_k and corner angles $\theta_i, \theta_j, \theta_k$. Let Δ be the new triangle whose edge lengths are $e^{u_j+u_k}l_i$, and then the Jacobian matrix is symmetric and negative semi-definite:

$$\begin{bmatrix} \frac{\partial \theta_i}{\partial u_i} & \frac{\partial \theta_i}{\partial u_j} & \frac{\partial \theta_i}{\partial u_k} \\ \frac{\partial \theta_j}{\partial u_i} & \frac{\partial \theta_j}{\partial u_j} & \frac{\partial \theta_j}{\partial u_k} \\ \frac{\partial \theta_k}{\partial u_i} & \frac{\partial \theta_k}{\partial u_j} & \frac{\partial \theta_k}{\partial u_k} \end{bmatrix} = \begin{bmatrix} -\cot \theta_k - \cot \theta_j & \cot \theta_k & \cot \theta_j \\ \cot \theta_k & -\cot \theta_k - \cot \theta_i & \cot \theta_i \\ \cot \theta_j & \cot \theta_i & -\cot \theta_j - \cot \theta_i \end{bmatrix}. \tag{18}$$

In particular, the locally concave function

$$F(u_i, u_j, u_k) = \int_0^u \theta_i du_i + \theta_j du_j + \theta_k du_k$$

is well defined and satisfies

$$\nabla F(u_i, u_j, u_k) = (\theta_i, \theta_j, \theta_k)^T.$$

Note that discrete curvature is built from the inner angles θ_i 's. The above formula relates a PL metric $u * \ell$ and its discrete curvature. The explicit form of the function F was found in the work of Bobenko-Pinkall-Springborn (Bobenko et al. 2015). They showed that F can be extended to a concave function on \mathbb{R}^3 and is related to the three-dimensional hyperbolic volume of ideal tetrahedra and is expressed in terms of the Lobachevsky function (i.e., dilogarithm).

Discrete Yamabe Flow A basic goal in geometry is to find the relationship between the metric and its curvature. In the discrete setting, it translates into the following questions.

Question 1. Metric Design by Curvature Given a polyhedral metric with edge length function ℓ on a closed triangulated surface (Σ, \mathcal{T}) and a function $\hat{K} : V(\mathcal{T}) \rightarrow (-\infty, 2\pi)$, can one find $u : V(\mathcal{T}) \rightarrow \mathbb{R}$ such that $u * \ell$ is still an edge length function on \mathcal{T} and its curvature $K_{u*\ell}$ is the given function \hat{K} ? Is the function u unique up to the addition of a constant? Suppose one can solve the prescribing curvature equation $K_{u*\ell} = \hat{K}$, how can one find u effectively?

Obviously the function \hat{K} must satisfy the Gauss-Bonnet condition in Eq. 16. If such a function u exists, then any other function that differs from u by a constant is also a solution of the problem.

These questions, together with Hamilton's Ricci flow, led to the introduction of the discrete Yamabe flow (Luo 2004):

$$\frac{du(t)}{dt}(v) = \hat{K}(v) - K_{u*\ell}(v). \tag{19}$$

The variational principle associated with (17) shows that the flow is the gradient flow of the locally concave discrete energy

$$\mathcal{E}(u) = \int_0^u \sum_{v \in V(\mathcal{T})} (\hat{K}(v) - K_{u*\ell}(v)) du(v), \tag{20}$$

We call $\mathcal{E}(u)$ as *Yamabe energy*. By direct computation, the gradient of Yamabe energy is

$$\nabla \mathcal{E}(u) = \bar{K} - K(u). \tag{21}$$

The Hessian matrix of Yamabe energy can be obtained by Eq. (18),

$$\frac{\partial^2 \mathcal{E}(u)}{\partial u_i \partial u_j} = -w_{ij}, \quad \frac{\partial^2 \mathcal{E}(u)}{\partial u_i^2} = \sum_{k \neq j} w_{ij}, \tag{22}$$

where w_{ij} is the *cotangent edge weight*: suppose two corner angles against edge $[v_i, v_j]$ are θ_k and θ_l ,

$$w_{ij} := \frac{1}{2}(\cot \theta_k + \cot \theta_l). \tag{23}$$

It is proved in Bobenko et al. (2015) that the solution to the equation $K_{u*\ell} = \hat{K}$ is unique in u up to the addition of a constant function. However, the existence of u , even if one assumes the Gauss-Bonnet condition on \hat{K} , is in general false, and the discrete Yamabe flow develops singularities in finite time.

Dynamic Yamabe Flow The drawback of (17) is that it depends on the choices of the triangulation \mathcal{T} . Recall that a *marked surface* is a pair (Σ, V) where V is a finite set in S . A PL metric on (Σ, V) is a PL metric on S such that its conical singularities are contained in V . By a triangulation \mathcal{T} of (Σ, V) , we mean a triangulation of Σ such that $V(\mathcal{T}) = V$.

Suppose d_1 and d_2 are two PL metrics on a marked surface (Σ, V) and \mathcal{T} and \mathcal{T}' are two triangulations of (Σ, V) . Let ℓ_k and ℓ'_k be the associated edge length functions of d_k for \mathcal{T} and \mathcal{T}' , $k = 1, 2$. As shown in the following diagram, where $\varrho_k : (\mathcal{T}, d_k) \rightarrow (\mathcal{T}', d_k)$ are isometries.

$$\begin{array}{ccc} \{S, V, d_1\}(\mathcal{T}, \ell_1) & \xrightarrow{u^*} & \{S, V, d_2\}(\mathcal{T}, \ell_2) \\ \varrho_1 \downarrow & & \downarrow \varrho_2 \\ \{S, V, d_1\}(\mathcal{T}', \ell'_1) & \xrightarrow{w^*} & \{S, V, d_2\}(\mathcal{T}', \ell'_2) \end{array}$$

Question 2. Triangulation Independence If $\ell_2 = u * \ell_1$, does it follow that $\ell'_2 = w * \ell'_1$ for some $w \in \mathbb{R}^V$?

An affirmative answer would imply that the vertex scaling operator in Eq. (17) is independent of the choice of triangulations. Unfortunately, the answer is negative in general. However, the condition $\ell'_2 = w * \ell'_1$ does hold for some w if we assume all triangulations \mathcal{T} and \mathcal{T}' are Delaunay in d_k for $k = 1, 2$. This is proved in Gu et al. (2018b). Recall that a *Delaunay triangulation* of a polyhedral surface is a geometric triangulation such that the sum of two angles facing each edge is at most π . Given a PL metric d on a marked surface (S, V) , there is always a Delaunay triangulation of (Σ, V, d) whose vertex set is V . Generically, Delaunay triangulation on (Σ, V, d) is unique. However, non-uniqueness occurs when the sum of the two angles facing an edge e is π . In this case, consider the quadrilateral Q formed by the two triangles adjacent to e and replace the diagonal e in Q by the other diagonal. The resulting triangulation is still Delaunay and the operation is called an edge flip. Note that edge flip does not change the underlying PL metric, but only the combinatorics.

A triangulation-independent definition of discrete conformal equivalence of PL metrics on a marked surface (Σ, V) was introduced in Gu et al. (2018b) by modifying vertex scaling in Eq. (17) and adding the Delaunay condition on triangulations.

Definition 1 (Related Discrete Metrics). Two PL metrics d_1 and d_2 on (Σ, V) are said to be related by a *move* if one can find Delaunay triangulations \mathcal{T}_k of d_k such that one of the following conditions holds:

1. $\mathcal{T}_1 = \mathcal{T}_2$ and their associated edge length functions ℓ_{d_1} and ℓ_{d_2} differ by a vertex scaling Eq. (17) on \mathcal{T}_1 .
2. $d_1 = d_2$ and \mathcal{T}_1 differ from \mathcal{T}_2 by an edge flip.

Definition 2 (Discrete Conformal Metrics). Two PL metrics on (Σ, V) are *discrete conformal* if they are related by a finite sequence of moves.

It turns out this is the correct notion to solve both the existence and uniqueness questions.

Theorem 1 (Existence and Uniqueness (Gu et al. 2018b)). Suppose d is a PL metric on a compact connected surface (Σ, V) and $\hat{K} : V \rightarrow (-\infty, 2\pi)$ is any function such that $\sum_{v \in V} \hat{K}(v) = 2\pi \chi(\Sigma)$. Then there exists a PL metric d^* , unique up to scaling, on (Σ, V) such that d^* is discrete conformal to d and its discrete curvature satisfies $K_{d^*} = \hat{K}$. Furthermore, d^* can be found using a finite dimensional convex variational principle.

Take the function \hat{K} to be the constant $\frac{2\pi \chi(\Sigma)}{|V|}$, we obtain the discrete version of uniformization theorem directly.

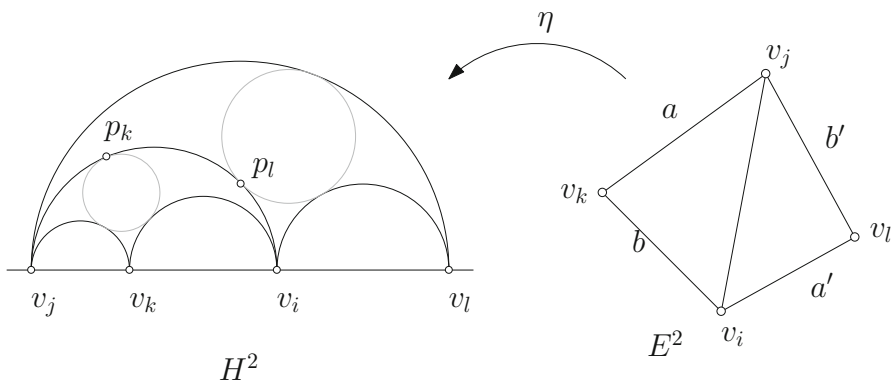


Fig. 14 Conversion from Euclidean geometry to hyperbolic geometry

Corollary 1 (Discrete Uniformization). *Each PL metric d on (Σ, d) is discrete conformal to a unique (up to scaling) PL metric d^* of constant discrete curvature.*

We remark that the above theorem for the case of the torus was first proved by F. Fillastre in a different context.

Hyperbolic Interpretation There is a hyperbolic geometric interpretation of the above discrete conformal equivalence which makes the geometric picture clear. As shown in Fig. 14, for each PL metric d on (Σ, V) , we choose a Delaunay triangulation \mathcal{T} of (S, V, d) . For each edge $[v_i, v_j]$ adjacent to two faces, the cross ratio of the edge $[v_i, v_j]$ is defined as

$$\text{Cr}([v_i, v_j]) = \frac{aa'}{bb'}$$

We convert each Euclidean triangle to an ideal hyperbolic triangle. Each ideal triangle has a unique inner circle which is tangent to each edge at a point. When two ideal triangles are isometrically glued along a common edge $[v_i, v_j]$, the *shear coordinates* on the edge $[v_i, v_j]$ are the signed hyperbolic distance from the tangent point p_l to p_k . A simple calculation shows that the shear coordinates equal to

$$d_{H^2}(p_l, p_k) = -\ln \text{Cr}([v_i, v_j]),$$

as shown in Fig. 14 left frame, where we use the upper half plane model for the hyperbolic plane. By isometrically gluing the ideal triangles with the shear coordinates, we obtain a complete finite area hyperbolic metric d_h on the punctured surface $\Sigma - V$. We denote this conversion as $d_h = \eta(d)$.

$$\begin{array}{ccc}
 \{S, V, d_1\} & \xrightarrow{\sim} & \{S, V, d_2\} \\
 \eta_1 \downarrow & & \downarrow \eta_2 \\
 \{S, V, d_{h,1}\} & \xrightarrow{=} & \{S, V, d_{h,2}\}
 \end{array}$$

It can be shown that the metric d_h is independent of the choice of the Delaunay triangulations and two PL metrics d_1 and d_2 are discrete conformal if and only if their associated hyperbolic metrics are isometric. Namely, the above diagram commutes. For more details, see Gu et al. (2018a,b).

The convergence of discrete conformal metrics to the Poincaré metrics on the torus was established in Gu et al. (2019). The algorithmic details can be found in Algorithm 1. More details can be found in Jin et al. (2008), Zhang et al. (2014), and Chen et al. (2016).

Algorithm 1 Discrete Surface Yamabe Flow

- 1: **Input:** A polyhedral surface (Σ, \mathcal{T}, d) ; Target curvature \bar{K} satisfying Gauss-Bonnet condition; Step length δ ; Error threshold ε ;
- 2: **Output:** A discrete metric \bar{d} , conformal to d and inducing the curvature \bar{K} .
- 3: Initialize the conformal factor $u_i = 0$, for all $v_i \in V(\mathcal{T})$;
- 4: **while true do**
- 5: Compute the edge length using vertex scaling using Eq. (17);
- 6: Update to Delaunay triangulation under the updated metric by edge swaps;
- 7: Compute the corner angles using cosine law Eq. (12);
- 8: Compute the cotangent edge weights using Eq. (23);
- 9: Compute the Hessian matrix of the Yamabe energy $D^2\mathcal{E}(u)$ using Eq. (22);
- 10: Compute the discrete Gaussian curvature using Eq. (15);
- 11: Compute the gradient of the Yamabe energy $\nabla\mathcal{E}(u)$ using Eq. 21;
- 12: **if** the norm of $\nabla\mathcal{E}(u)$ is less than ε **then**
- 13: return current triangulation, conformal factor u and the edge length.
- 14: **end if**
- 15: Use Newton’s method to update the conformal factor:

$$u \leftarrow u + \delta(D^2\mathcal{E}(u))^{-1}\nabla\mathcal{E}(u) \tag{24}$$

16: **end while**

Hyperbolic Yamabe Flow For discrete surfaces with hyperbolic background geometry, the vertex scaling $y = u * \ell$ is defined by

$$\sinh \frac{y_k}{2} = e^{u_i} \sinh \frac{l_k}{2} e^{u_j} \tag{25}$$

This was introduced in Bobenko et al. (2015). The Yamabe energy is defined similarly

$$\mathcal{E}(u) = \int^u \sum_{v \in \Sigma} (\bar{K} - K(v)) du(v). \tag{26}$$

The gradient of the Yamabe energy is

$$\nabla \mathcal{E}(u) = \bar{K} - K(u). \tag{27}$$

The Hessian matrix of the Yamabe energy can be derived from one face case,

$$\begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \end{bmatrix} = \frac{-1}{A} \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & S_3 \end{bmatrix} \begin{bmatrix} -1 & \cos \theta_3 & \cos \theta_2 \\ \cos \theta_3 & -1 & \cos \theta_1 \\ \cos \theta_2 & \cos \theta_1 & -1 \end{bmatrix} \begin{bmatrix} 0 & \frac{S_1}{C_{1+1}} & \frac{S_1}{C_{1+1}} \\ \frac{S_2}{C_{2+1}} & 0 & \frac{S_2}{C_{2+1}} \\ \frac{S_3}{C_{3+1}} & \frac{S_3}{C_{3+1}} & 0 \end{bmatrix} \begin{bmatrix} du_1 \\ du_2 \\ du_3 \end{bmatrix} \tag{28}$$

where $S_k = \sinh y_k$ and $C_k = \cosh y_k$.

A Discrete Conformal Geometry of Polyhedral Surfaces Derived from Circle Patterns

The edge flip operation used in the above discrete conformal equivalence relation has created computational complications. There is a more robust, triangulation-dependent discrete curvature flow which one can use to find PL metrics with the targeted curvatures. The basic idea comes from W. Thurston’s work on circle packing and Hamilton’s work on Ricci flow. Unlike the previous conformal equivalence which is derived from discretizing the conformal Riemannian metric $e^u \mathbf{g}$, this new discretization focuses on the infinitesimal circle-preserving property of the conformal maps. The associated finite dimensional variational principle was first established by Colin de Verdière (1991) in the tangential case and in the general case in Chow and Luo (2003). Based on the variational principle, the work Chow and Luo (2003) introduced discrete Ricci flow (30) on surfaces and established its basic properties. Algorithmic details can be found in Luo et al. (2007) and Zeng and Gu (2013).

Here are some mathematical details. Given a triangulated surface (S, \mathcal{T}) and an assignment of edge weight $\Theta : E(\mathcal{T}) \rightarrow [0, \pi)$ (measuring the intersection angles of circles), a circle packing metric is a function, called radius assignment, $r : V(\mathcal{T}) \rightarrow \mathbb{R}_{>0}$ such that the associated length function

$$l(v_1 v_2) = \sqrt{r(v_1)^2 + r(v_2)^2 + 2r(v_1)r(v_2) \cos(\Theta(v_1 v_2))} \tag{29}$$

produces a PL metric on (S, \mathcal{T}) , i.e., satisfies the triangular inequality $l(e_i) + l(e_j) > l(e_k)$ for every triple of edges $\{e_i, e_j, e_k\}$ belonging to a triangle in \mathcal{T} . Thurston proved that if $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$ (see Fig. 13b), then the triangle inequality always holds for all choices of $r \in \mathbb{R}^{V(\mathcal{T})}$ (see Thurston 1997). The discrete

curvature K_r of r is defined to be the discrete curvature of the PL metric l . In this setting, a discrete conformal class is defined as the set of all PL metrics induced by different choices of $r : V(\mathcal{T}) \rightarrow \mathbb{R}_{>0}$ for a fixed edge weight Θ . Since different choices of r amount to different sizes of circles at vertices, this discretization captures the circle-preserving properties of the conformal maps.

The basic questions are for a fixed prescribed Θ to find a radius assignment $r \in \mathbb{R}^{V(\mathcal{T})}$ such that its curvature K_r is a prescribed function \hat{K} and to determine if r is unique up to scaling. For $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$, both of them were solved by W. Thurston in his famous notes (Thurston 1997). He proved that r is unique up to scaling and found the necessary and sufficient conditions on \hat{K} to be solvable by r . Most remarkably the conditions discovered by Thurston on \hat{K} are the Gauss-Bonnet (linear) equation and a finite set of linear inequalities.

The variational principle associated with the circle packing takes the following form. Fix $\Phi_1, \Phi_2, \Phi_3 \in [0, \pi)$. Suppose a triangle Δ has edge lengths l_1, l_2, l_3 of the form given by $l_i^2 = r_j^2 + r_k^2 + 2r_j r_k \cos(\Phi_i)$ and $r_i = e^{u_i}$. If one denotes the angles of Δ by a_i , then the Jacobian matrix $[\frac{\partial a_i}{\partial u_j}]_{3 \times 3}$ is symmetric. If furthermore $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$, then the matrix is negative semi-definite of rank 2. In particular, there is a concave function $W(u)$ defined on $\mathbb{R}^{V(\mathcal{T})}$ such that $\frac{\partial W}{\partial u_i} = a_i$. This implies that the discrete Ricci flow defined as

$$\frac{dr(t)}{dt}(v) = -2(K_r(v) - \hat{K}(v))r(t)(v) \tag{30}$$

is the gradient flow of a concave function (namely, W). From this fact, many of the basic properties, including long time existence, of discrete Ricci flow follow. The flow is robust and algorithmically effective if $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$.

Discrete Ricci flow does not work well if one $\Theta(e)$ lies outside of the interval $[0, \pi/2]$. This is one of the drawbacks of the flow for real-world applications. Many polyhedral surfaces produced by digital media cannot be expressed as circle packing metrics such that $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$. Modifications of the triangular meshes are needed to achieve the condition $\Theta(E(\mathcal{T})) \subset [0, \pi/2]$.

The convergence of circle packing metrics on bounded simply connected domains to the Riemann mapping was first conjectured by Thurston in 1985 and proved in a celebrated paper by Rodin-Sullivan in (1987). However, convergence questions for nonplanar surfaces remain open.

Below are some examples of discrete Ricci flows. Figure 19 shows one example of computing the extremal length of a topological quadrilateral using Ricci flow. Basically, we set the target curvature to be zero for all interior and boundary vertices, except the four corners, and set the target curvatures for the corners to be $\pi/2$, then we run Ricci flow to get the target metric, and isometrically embed the surface using the target metric to obtain the planar rectangle.

Figure 15 shows a generalization of circle packing by replacing circles by squares to compute the extremal length of a combinatorial quadrilateral. The left frame shows a three-connected graph, with four corner nodes. The right frame shows

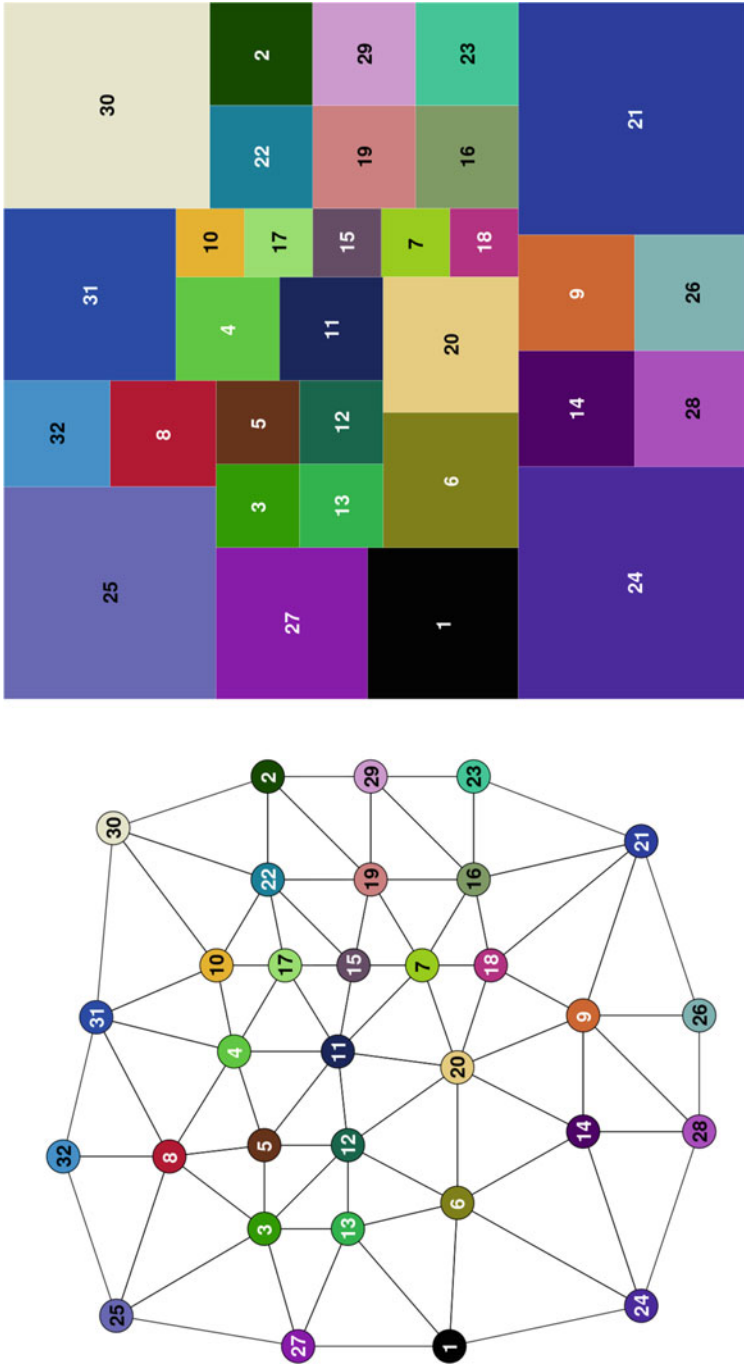


Fig. 15 Square tiling of a three-connected graph; each node is replaced by a square with the same label and color. Two nodes are connected in the graph, if and only if their corresponding squares are tangent

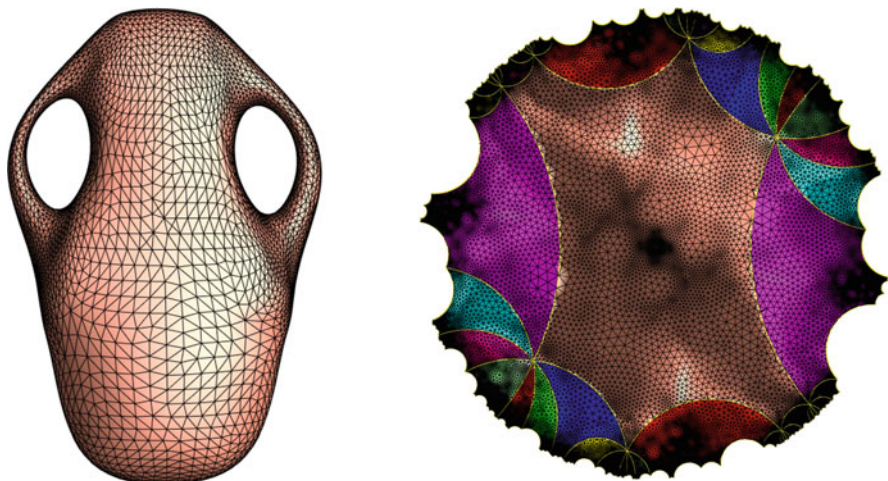


Fig. 16 Uniformization for high genus surfaces

the extremal length, where each node is replaced by a square with the same label and color. Two nodes are connected in the graph if and only if their corresponding squares are tangent. In theory, squares can be replaced by more general convex shapes.

Figure 16 shows an example for computing the hyperbolic metric on a high genus surface. As shown in the left frame, the input surface is triangulated, and each face is a hyperbolic triangle instead of a Euclidean triangle. The theoretic formulation and the algorithmic details are very similar. After obtaining the uniformization metric, we isometrically embed a finite portion of the universal covering space of the surface onto the Poincaré model of \mathbb{H}^2 . Each color represents a fundamental polygon, and the boundaries of the fundamental polygons are hyperbolic geodesics.

Harmonic Maps

Another useful algorithm is based on surface harmonic maps for a genus zero closed surfaces (Gu et al. 2004). Figure 17 shows the computational method for genus zero closed surface: harmonic mapping.

Intuitively, the harmonic energy measures the elastic deformation energy induced by a mapping between surfaces. It depends on the Riemannian metric of the target surface and the conformal structure of the source surface. Given a C^1 mapping between two surfaces $f : (S, \mathbf{g}) \rightarrow (T, \mathbf{h})$, with isothermal parameters,

$$\mathbf{g} = e^{2\mu(x,y)}(dx^2 + dy^2), \quad \mathbf{h} = e^{2\lambda(u,v)}(du^2 + dv^2),$$

the *harmonic energy density* of the mapping is given by

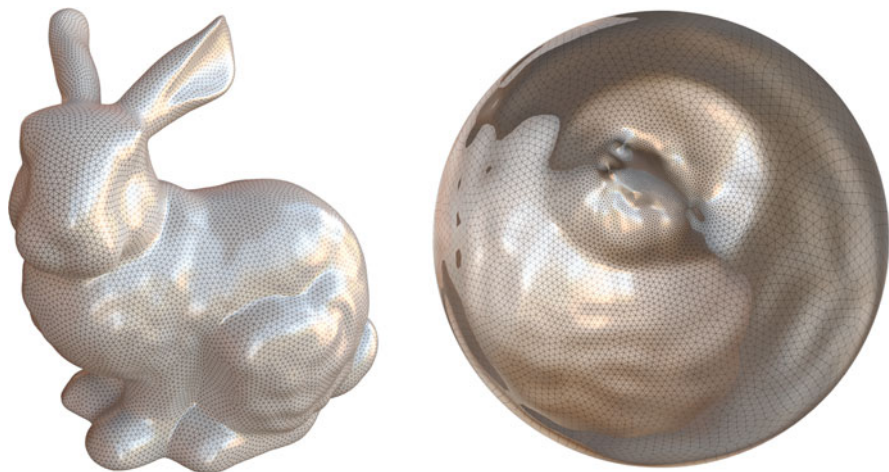


Fig. 17 A spherical harmonic mapping from the Stanford bunny surface onto the unit sphere

$$e(f, \mathbf{g}, \mathbf{h}) := \frac{e^{2\lambda(u,v)}}{e^{2\mu(x,y)}} (|\nabla u|^2 + |\nabla v|^2)$$

The *harmonic energy* of the map is given by

$$E(f, \mathbf{g}, \mathbf{h}) = \int_S e^{2\lambda(u,v)} (|\nabla u|^2 + |\nabla v|^2) dx dy.$$

Harmonic maps are the critical points of the harmonic energy.

If the target surface has negative Gaussian curvature, and the mapping degree is one, then the harmonic mapping is diffeomorphic. A mapping between two surfaces induces the so-called Hopf differential on the source surface. If the mapping is harmonic, then its Hopf differential is a holomorphic quadratic differential on the source surface. Furthermore, if the Hopf differential is zero, then the mapping is conformal. Since holomorphic quadratic differentials on a genus zero surface must be zero, harmonic maps between genus zero closed surfaces must be conformal. Suppose $f_1, f_2 : (S, \mathbf{g}) \rightarrow \mathbb{S}^2$ are two degree one harmonic maps from a genus zero surface to the unit sphere, then they differ by a Möbius transform of the sphere. By stereographic projection, we can map \mathbb{S}^2 to the extended complex plane $\mathbb{C} \cup \{\infty\}$. The Möbius transform has the formula

$$z \mapsto \frac{az + b}{cz + d}, \quad ad - bc = 1, a, b, c, d \in \mathbb{C}.$$

Suppose (Σ, \mathcal{T}) is a polyhedral surface, a vector valued function defined on the vertex set $f : V(\mathcal{T}) \rightarrow \mathbb{R}^3$ can be linearly extended to a global piecewise linear function, and by abusing the notation, we denote it as $f : \Sigma \rightarrow \mathbb{R}^3$. By direct computation, the *harmonic energy* of the mapping is given by

$$E(f) = \sum_{[v_i, v_j] \in \Sigma} w_{ij} |f(v_i) - f(v_j)|^2, \quad (31)$$

where w_{ij} is the *cotangent edge weight*: suppose two corner angles against edge $[v_i, v_j]$ are θ_k and θ_l , then

$$w_{ij} := \frac{1}{2}(\cot \theta_k + \cot \theta_l). \quad (32)$$

The variation of the harmonic energy is the Laplacian operator,

$$\Delta f(v_i) = 2 \sum_{v_i \sim v_j} w_{ij} (f(v_i) - f(v_j)). \quad (33)$$

In practice, we first construct the Gauss map from the Stanford bunny surface to the unit sphere and then use the nonlinear heat diffusion method to reduce the harmonic energy. At the k -th step, we compute the Laplacian of f_k , Δf_k ; then we project Δf_k to the tangent space of the sphere. The normal component of Δf_k is

$$(\Delta f_k)^\perp := \langle \Delta f_k(v_i), f_k(v_i) \rangle f_k(v_i), \quad (34)$$

and the tangential component is

$$(\Delta f_k)^T(v_i) = \Delta f_k(v_i) - \Delta f_k^\perp(v_i). \quad (35)$$

Then we update the mapping by removing tangential component of Laplacian

$$f_{k+1}(v_i) \leftarrow f_k(v_i) - \tau (\Delta f_k)^T(v_i) \quad (36)$$

where τ is the step length. In order to remove the Möbius ambiguity, we add one constraint that the mass center of the image of f_{k+1} is at the origin,

$$c_{k+1} := \frac{1}{|V|} \sum_{v_i \in V} f_{k+1}(v_i),$$

If c_{k+1} is not zero, then we enforce it by adding a normalization step

$$f_{k+1}(v_i) \leftarrow f_{k+1}(v_i) - c_{k+1}. \quad (37)$$

We repeat this procedure, until the norm of the tangential component of the map is less than a user prescribed threshold. The details can be found in Algorithm 2. More algorithmic details can be found in Gu et al. (2004) and Gotsman et al. (2003).

Algorithm 2 Harmonic Map

```

1: Input: Genus zero closed polyhedral surface  $\Sigma$ ; step length  $\tau$ ; error threshold  $\varepsilon$ 
2: Output: Harmonic map  $f : \Sigma \rightarrow \mathbb{S}^2$ .
3: Compute the discrete Gauss map  $f_0 : \Sigma \rightarrow \mathbb{S}^2$ ;
4: while true do
5:   Compute the Laplacian of the map  $\Delta f_k$  using Eq. (33);
6:   Compute the normal component of the Laplacian  $(\Delta f_k)^\perp$  using Eq. (34);
7:   Compute the tangential component of the Laplacian  $(\Delta f_k)^T$  using Eq. (35);
8:   if the norm of  $(\Delta f_k)^T$  is less than  $\varepsilon$  then
9:     return  $\hat{f}_k$ .
10:  end if
11:  Update the mapping by Eq. (36);
12:  Normalize the mapping  $f_{k+1}$  using Eq. (37);
13: end while

```

The harmonic map between hyperbolic surfaces was introduced in Shi et al. (2016), for high genus surface registration. The harmonic map between a surface and a graph with distance was introduced in Lei et al. (2017a,b), and this is applied for computing holomorphic quadratic differentials for the purpose of computational mechanics.

Hodge Decomposition

Another algorithm is based on Hodge decomposition theorem (Gu and Yau 2003). Hodge decomposition says that any differential form ω on a closed Riemannian manifold can be uniquely written as the sum of three parts: $\omega = d\alpha + \delta\beta + \gamma$, where γ is harmonic $\Delta\gamma = 0$ and $\Delta = d\delta + \delta d$. Intuitively, this can be interpreted as any vector field on a surface can be decomposed into three components: a curl-free part, divergence-free part, and harmonic part. A vector field is harmonic if and only if it has zero curl and zero divergence, as shown in Fig. 18.

Homology Group We compute the basis of the homology group of the polyhedral surface (Σ, \mathcal{T}) , denoted as $H_1(\Sigma, \mathbb{Z})$. We compute the Poincaré dual $\bar{\Sigma}$ of the surface Σ and compute a spanning tree \bar{T} of the vertices of $\bar{\Sigma}$, and then the *cut graph* of Σ is given by

$$\Gamma := \{e \in (\Sigma, \mathcal{T}) : \bar{e} \notin \bar{T}\}.$$

Then we compute a spanning tree T of Γ , suppose the edges

$$\Gamma \setminus T = \{e_1, e_2, \dots, e_{2g}\}.$$

The union of e_k and T has a unique loop γ_k , and then $\{\gamma_1, \gamma_2, \dots, \gamma_{2g}\}$ is a set of basis of $H_1(\Sigma, \mathbb{Z})$.



Fig. 18 Two conjugate harmonic one-forms, in the left and middle frames, consist a holomorphic one-form in the right frame. The loops in the left (middle) frame show the vertical (horizontal) trajectories of the holomorphic form

Cohomology Group Second we compute a set of basis of the cohomology group $H^1(\Sigma, \mathbb{R})$. For each homology group base loop $\gamma_k, k = 1, 2, \dots, 2g$, we slice the surface along γ_k to obtain $\Sigma_k = \Sigma \setminus \gamma_k$. Then Σ_k has two boundary components,

$$\partial \Sigma_k = \gamma_k^+ - \gamma_k^-.$$

We construct a function $f_k : \Sigma \rightarrow \mathbb{R}$, such that the restriction of f_k on γ_k^+ is $+1$, on γ_k^- is 0 , and random on interior vertices. Then we define the discrete one-form ω_k ,

$$\omega_k([v_i, v_j]) := df_k([v_i, v_j]) = f_k(v_j) - f_k(v_i).$$

Then ω_k equals to zero on the boundary edges; hence ω_k is defined on the original closed surface Σ . By this construction, the closed one-forms $\{\omega_1, \omega_2, \dots, \omega_{2g}\}$ form a set of basis of the first cohomology group of the polyhedral surface $H^1(\Sigma, \mathbb{R})$.

Harmonic One-Form Group According to Hodge theory, each cohomological class has a unique harmonic form. All the harmonic one-forms consist of a group $H_\Delta(\Sigma, \mathbb{R})$, which is isomorphic to the cohomology group $H^1(\Sigma, \mathbb{R})$.

For each base one-form $\omega_k \in H^1(\Sigma, \mathbb{R})$, there is a unique function $f_k : \Sigma \rightarrow \mathbb{R}$, such that $\omega_k + df_k$ is harmonic, which satisfies: $\delta(\omega_k + df_k) = 0$,

$$\sum_{v_i \sim v_j} w_{ij}(\omega_k([v_i, v_j]) + f_k(v_j) - f_k(v_i)) = 0, \quad \forall v_i \in V(\mathcal{T}).$$

These equations determine f_k unique up to a constant. Let $\eta_k = \omega_k + df_k$, and then $\{\eta_1, \eta_2, \dots, \eta_{2g}\}$ forms a set of basis of $H_\Delta(\Sigma, \mathbb{R})$.

Holomorphic One-Form Group Each harmonic one-form on the surface is equivalent to a tangent vector field \mathbf{v} , which is curl-free and divergence-free. If we rotate the tangent vector $\mathbf{v}(p)$ about the normal $\mathbf{n}(p)$ by $\pi/2$ angle, we obtain another curl-free and divergence-free tangent vector field $*\mathbf{v}$. $*\mathbf{v}$ is equivalent to a harmonic

one-form ${}^*\omega_k$, and this operator is called *Hodge star*. Because $\{\omega_1, \omega_2, \dots, \omega_{2g}\}$ is a set of basis of $H_\Delta(\Sigma, \mathbb{R})$, ${}^*\omega_k$ can be represented as a linear combination of them. We can construct linear equations to find the linear combination coefficients,

$$\int_{\Sigma} {}^*\omega_k \wedge \omega_i = \sum_{j=1}^{2g} \lambda_{kj} \omega_j \wedge \omega_i.$$

And the left-hand side can be evaluated using vector field representation. For example, we isometrically embed one face Δ on the (x, y) -plane, $\omega_k = \alpha_k dx + \beta_k dy$, and then ${}^*\omega_k = \alpha_k dy - \beta_k dx$. $\omega_i = \alpha_i dx + \beta_i dy$,

$$\int_{\Delta} {}^*\omega_k \wedge \omega_i = -(\alpha_i \alpha_k + \beta_i \beta_k) A(\Delta),$$

where $A(\Delta)$ is the area of the triangle. We form the holomorphic one-form $\varphi_k = \omega_k + i{}^*\omega_k$, and then $\{\varphi_1, \varphi_2, \dots, \varphi_{2g}\}$ is a set of basis of the holomorphic one-form group of (Σ, \mathcal{T}) . The algorithmic pipeline is summarized in Algorithm 3

Algorithm 3 Holomorphic One-Forms

- 1: **Input:** Genus $g > 0$ closed polyhedral surface with a triangulation (Σ, \mathcal{T}) ;
 - 2: **Output:** Holomorphic one-form group basis.
 - 3: Compute the basis of homology group $H_1(\Sigma, \mathbb{Z})$, $\{\gamma_1, \gamma_2, \dots, \gamma_{2g}\}$;
 - 4: Compute the basis of cohomology group $H^1(\Sigma, \mathbb{R})$;
 - 5: Compute the basis of harmonic one-form group $H_\Delta(\Sigma, \mathbb{R})$;
 - 6: Compute the basis of holomorphic one-form group $\{\varphi_1, \varphi_2, \dots, \varphi_{2g}\}$.
-

As shown in Fig. 18, the left frame shows a harmonic one-form ω , the middle frame shows the conjugate harmonic one-form ${}^*\omega$, and the right frame shows a holomorphic one-form $\omega + \sqrt{-1}{}^*\omega$. Using this method, we can construct the basis of the group of holomorphic one-forms of the Riemann surface. By linear combination, we can construct any holomorphic one-form. The algorithmic details can be found in Gu and Yau (2003) and Jin et al. (2004).

Direct Applications

Conformal geometry can be applied for computer vision and medical imaging directly. In the following, we introduce some of the most direct applications. More applications can be found in Gu and Yau (2007, 2020), Gu et al. (2012), and Zeng and Gu (2013).

Shape Space

All the surfaces in the real world form a *shape space*. The shape space can be classified using different transformation groups. The equivalence classes form the quotient shape spaces. The transformation groups form a hierarchical chain of subgroups, the corresponding quotient spaces for a sequence of subspaces. The *homeomorphism group* classifies the shape space by topology; each topological equivalent class can be further classified by *conformal transformation group*, all the conformal equivalence classes form the Teichmüller space; each conformal equivalence class can be further classified by the *isometric transformation group*; each isometric class can be further classified by the *rigid motion group* (translation and rotation). Two surfaces differ by a rigid motion if and only if they have the same Riemannian metric, mean curvature, and boundary position.

This work focuses on conformal classification, namely, discriminating shapes in the Teichmüller space. In the following, we introduce efficient algorithms to compute the Teichmüller coordinates for metric surfaces with different topologies. In practice, homeomorphic surfaces can be differentiated by their Teichmüller coordinates.

Topological Quadrilateral A topological disk with four boundary markers is called a *topological quadrilateral*. As shown in Fig. 19, we choose four markers $\{p_1, p_2, p_3, p_4\}$ on the boundary of a human facial surface. A topological quadrilateral with a Riemannian metric can be conformally mapped onto a planar rectangle. Two topological quadrilaterals are conformally equivalent, if and only if their corresponding rectangles are similar. Therefore, we use the ratio between the height and the width of the rectangle as its Teichmüller coordinate, which is also called *extremal length* of the topological quadrilateral. The extremal length is determined by both the geometry of the surface and the choices of the four markers.

The computation of the extremal length is straightforward by using the curvature flow algorithm. We set the target Gaussian curvature for interior vertices to be zero, those for the four boundary corner vertices to be $\pi/2$, and the target geodesic curvatures for all other boundary vertices to be zero as well. The discrete surface Yamabe flow will compute a flat metric with the target curvature, and then we isometrically flatten the polyhedral surface to obtain the planar rectangle.

Topological Annulus A topological annulus is a genus zero surface with two connected boundary components. Suppose Σ is a topological annulus with a Riemannian metric \mathbf{g} and the boundary of S are two loops $\partial S = \gamma_1 - \gamma_2$. We compute a holomorphic one-form ω , such that the imaginary component of the integration of ω along γ_1 is 2π . Fix a point q , the conformal mapping $\varphi(p) = \exp(\int_q^p \omega)$ maps the surface onto a canonical annulus, as shown in Fig. 20.

The Teichmüller coordinates of a topological annulus are given by the ratio between the inner radius and the outer radius. Two topologically annuli are

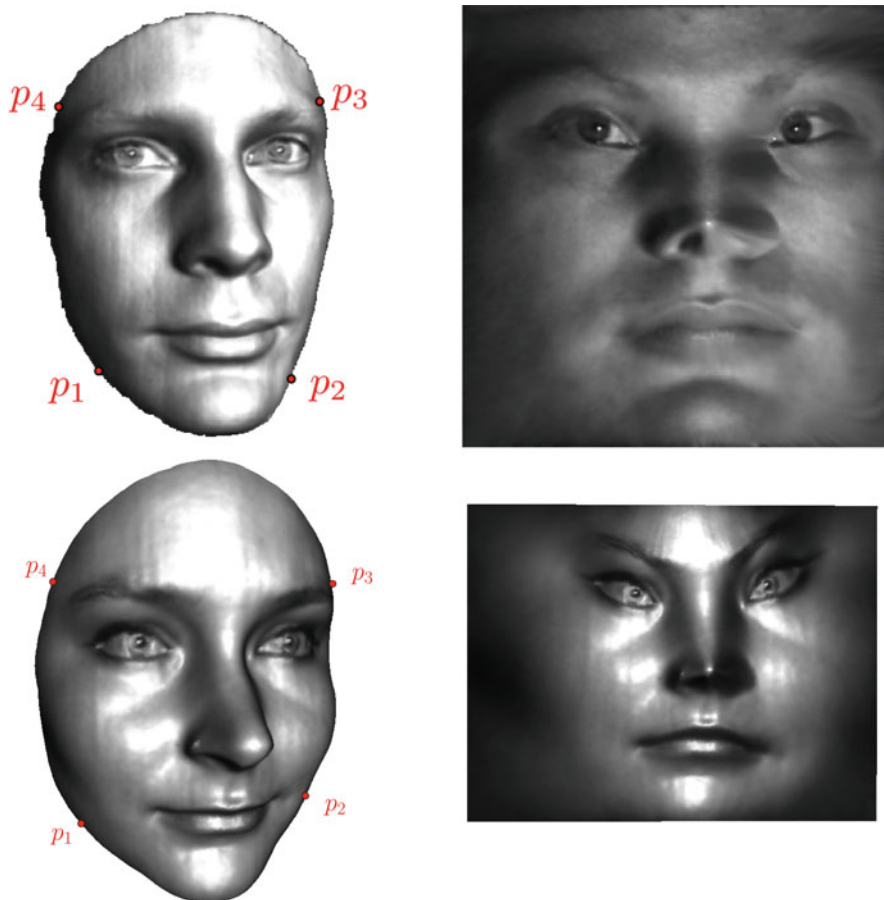


Fig. 19 The extremal lengths of topological quadrilateral surfaces

conformally equivalent if and only if their conformal planar annuli are similar, namely, they share the same Teichmüller coordinate.

Topological Poly-annulus Suppose Σ is a genus zero surface with multiple connected boundary components, $\partial S = \gamma_0 - \gamma_1 - \gamma_2 \cdots - \gamma_n$, then S is called a *topological poly-annulus*. A topological poly-annulus can be conformally mapped onto a planar annulus with concentric circular slits as shown in Fig. 21. The outer boundary component γ_0 is mapped onto the unit circle, one of the inner boundary components γ_1 is mapped to a circle centered at the origin, and all other boundary components are mapped to the concentric circular slits. The circle radii of all inner boundary components and the starting and ending angles of all circular slits form

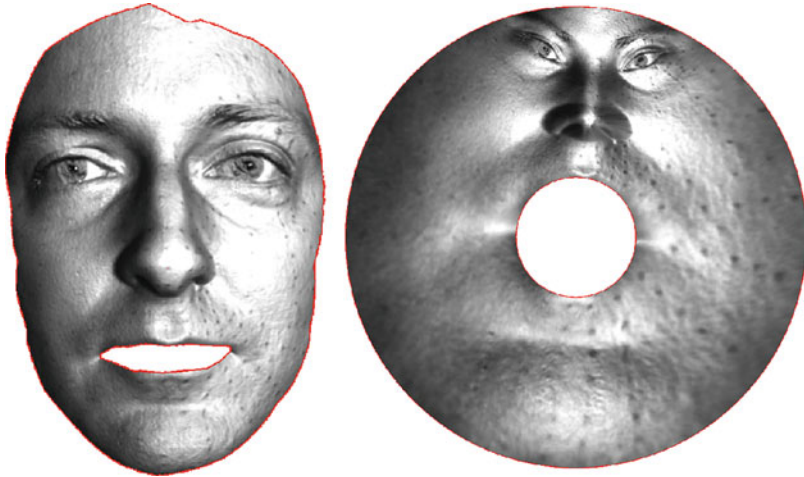


Fig. 20 Conformal mapping for a topological annulus

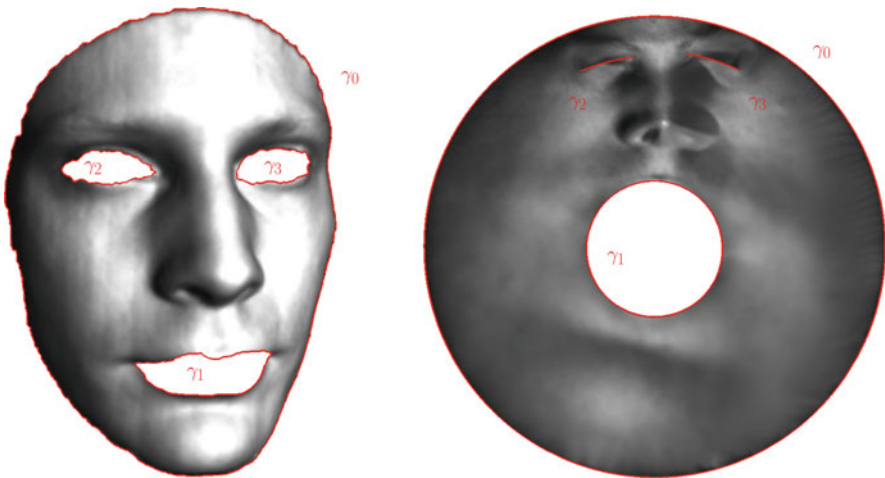


Fig. 21 Circular slit mapping for a topological poly-annulus

the Teichmüller coordinate of the surface. For topological poly-annulus with n inner holes, the dimension of the Teichmüller space is $3n - 3$.

We can construct a unique holomorphic one-form φ , such that the imaginary part of the integration of φ along γ_0 is 2π , -2π along γ_1 , and 0 along all other boundary components. Then we fix a base point q , and the conformal mapping $f(p) = \exp(\int_q^p \varphi)$ maps the surface onto a canonical annulus with concentric circular slits, as shown in Fig. 21. The algorithm was introduced in Yin et al. (2008).

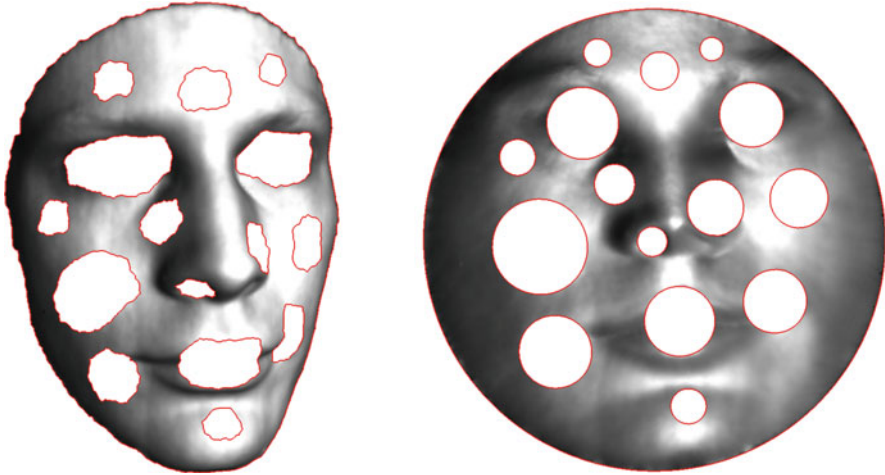


Fig. 22 Conformal mapping from a topological poly-annulus to a planar circle domain

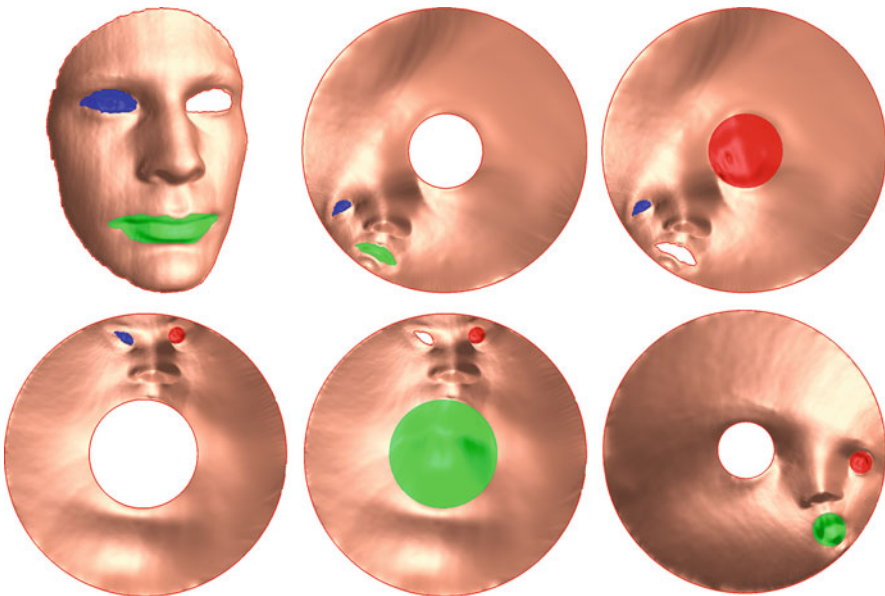


Fig. 23 Koebe's iteration algorithm

Another way to compute the conformal invariants of topological poly-annulus is to conformally map the surface onto a *circle domain*, namely, the unit disk with circular inner holes as shown in Fig. 22, and all such kind of mappings differ by a Möbius transformation of the disk. The Teichmüller coordinate of the surface is given by the centers and radii of the inner circles. Therefore if the surface has n inner

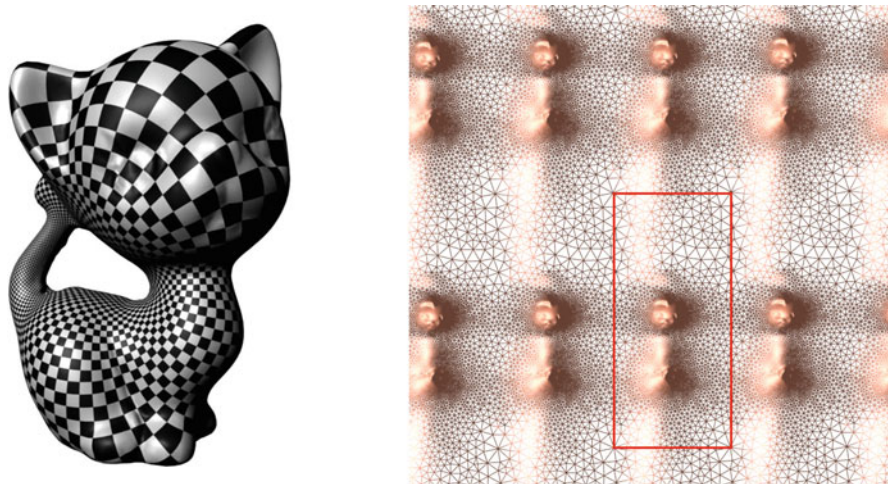


Fig. 24 Conformal periodic mapping from a genus one closed surface to the plane

boundary components, we need $3n - 3$ parameters to describe the circle domain, and the Teichmüller space is $3n - 3$ dimensional.

Figure 23 demonstrates Koebe’s iteration algorithm (Zeng et al. 2009) that conformally maps a poly-annulus onto a circle domain, namely, the complement of the union of a finite number of disks. First, we fill the holes of the mouth and the right eye and then conformally map the topological annulus onto a canonical annulus; second, we fill the center circular hole of the left eye, open the hole of the mouth, and map the topological annulus onto a canonical annulus; third, we fill the center circular hole of the mouth, open the hole of the right eye, and map the topological annulus onto a canonical annulus. We repeat this procedure, sequentially opening one hole and filling all the other holes, and then map the topological annulus to the canonical annulus. The boundary components become rounder and rounder, and the mapping images converge to a circle domain exponentially fast.

Genus One Closed Surfaces

As shown in Fig. 24, a genus one closed surface Σ can be conformally periodically mapped onto the plane. Each period is a parallelogram, which is called a fundamental domain. All the fundamental domains tessellate the whole complex plane. The vertices of the parallelograms form a lattice,

$$\Gamma = \{a + bz : a, b \in \mathbb{Z}\},$$

where $z \in \mathbb{C}$ is a constant.

The flat torus is defined as the quotient space \mathbb{C}/Γ , and the conformal mapping is between the input surface and the flat torus $f : \Sigma \rightarrow \mathbb{C}/\Gamma$. The Teichmüller coordinate of the genus one closed surface is given by the z parameter for the

flat torus. Therefore, the Teichmüller space of genus one closed surface is two-dimensional.

There are two ways to compute the conformal mapping for a torus. One way is based on discrete surface Yamabe flow. We set the target curvature to be zero everywhere and compute the flat metric using the flow. We slice the surface Σ open along a set of homology group basis $\{\gamma_1, \gamma_2\}$ to obtain a topological disk $\bar{\Sigma}$ and isometrically flatten $\bar{\Sigma}$ on the plane to obtain a fundamental domain $f(\bar{\Sigma})$. By gluing the translated copies of the fundamental domain, we can tessellate the whole plane and construct the flat torus \mathbb{C}/Γ .

The second method is based on holomorphic one-form algorithm. First, we compute a holomorphic one-form φ , then we choose a base point $q \in \bar{\Sigma}$, and define the mapping by integration,

$$f(p) = \int_q^p \varphi, \quad \forall p \in \bar{\Sigma}.$$

This gives a fundamental domain $f(\bar{\Sigma})$ on the plane.

High Genus Closed Surface The conformal invariants of a high genus closed surface can be computed using hyperbolic uniformization metric. As shown in Fig. 25, given a genus $g > 0$ closed surface Σ , we can choose a set of canonical basis of the fundamental group $\pi_1(\Sigma, q)$, $\{a_1, b_1, a_2, b_2, \dots, a_g, b_g\}$, such that all of them go through the base point $q \in \Sigma$, and satisfy the intersection conditions:

$$a_i \cdot b_i = 1, \quad a_i \cdot b_j = 0 \quad a_i \cdot a_j = 0, \quad b_i \cdot b_j = 0,$$

where $a_i \cdot b_j$ represents the algebraic intersection number between two loops a_i and b_j . We slice the surface along the canonical basis to form a fundamental domain $\bar{\Sigma}$, whose boundary is given by

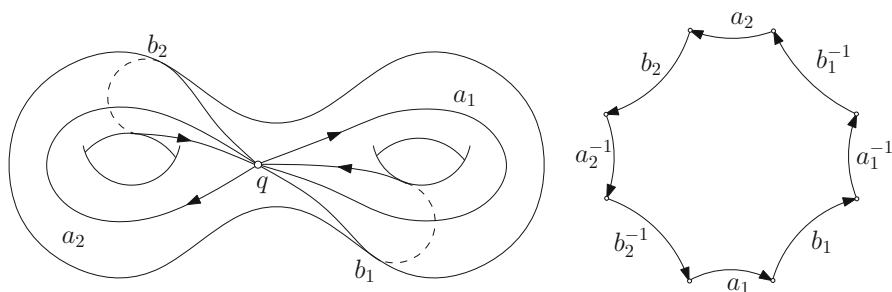


Fig. 25 Canonical fundamental group basis of a genus two closed surface

$$\partial \bar{\Sigma} = a_1 b_1 a_1^{-1} b_1^{-1} a_2 b_2 a_2^{-1} b_2^{-1} \cdots a_g b_g a_g^{-1} b_g^{-1}.$$

Similar to the torus case, by using the hyperbolic uniformization metric, we can isometrically embed $\bar{\Sigma}$ onto the hyperbolic plane to get a hyperbolic fundamental polygon. By transforming the fundamental polygon by hyperbolic rigid motions, we can generate a tessellation of the whole hyperbolic plane. All such kind of hyperbolic rigid motions form the so-called Fuchsian group of the surface. The generators of the Fuchsian group gives the Teichmüller coordinate of the surface Σ .

As shown in Fig. 26, we use the Poincaré’s disk model to represent the hyperbolic plane \mathbb{H}^2 ,

$$\mathbb{H}^2 := \left\{ |z| < 1, ds^2 = \frac{dzd\bar{z}}{(1 - z\bar{z})^2} \right\}.$$

The hyperbolic rigid motions are Möbius transformations

$$z \mapsto e^{i\theta} \frac{z - z_0}{1 - \bar{z}_0 z}, \quad |z_0| < 1, \theta \in [0, 2\pi).$$

The Fuchsian group of Σ is generated by Möbius transformations,

$$\text{Fuchs}(\Sigma) = \langle \alpha_1, \beta_1, \dots, \alpha_g, \beta_g | \prod_{i=1}^g [\alpha_i, \beta_i] = e \rangle,$$

where $[\alpha_i, \beta_i] = \alpha_i \beta_i \alpha_i^{-1} \beta_i^{-1}$. Each Möbius transformation requires *three* parameters, and there are $6g$ parameters in total. But the constraint $\prod_{i=1}^g [\alpha_i, \beta_i] = e$ removes three freedoms. Let φ be a Möbius transformation, and then $\{\varphi \alpha_i \varphi^{-1}, \varphi \beta_i \varphi^{-1}\}_{i=1}^g$ is also a set of generators of the Fuchsian group of Σ . This removes another 3 degrees of freedoms. Hence there are $6g - 6$ independent parameters of $\text{Fuchs}(\Sigma)$, and the Teichmüller space is $6g - 6$ dimensional.

The computation is straightforward. We use hyperbolic surface Yamabe flow to compute the hyperbolic uniformization metric and then isometrically embed $\bar{\Sigma}$ onto the hyperbolic plane \mathbb{H}^2 , $f : \bar{\Sigma} \rightarrow \mathbb{H}^2$. Then there are unique Möbius transformations α_k, β_k such that

$$f(b_k) = \alpha_k(f(b_k^{-1})), f(a_k) = \beta_k^{-1}(f(a_k^{-1})), \quad k = 1, 2, \dots, g.$$

$\{\alpha_k, \beta_k\}_{k=1}^g$ is a set of generators of the Fuchsian group of Σ . We use transformations in $\text{Fuchs}(\Sigma)$ to transform $f(\bar{\Sigma})$ and glue all the copies to tessellate the hyperbolic plane. Figure 26 shows two examples of the Fuchsian groups of surfaces; one is a genus two surface, while the other is a genus three surface.

More applications and algorithmic details for shape space can be found in Jin et al. (2009a,b) and Zeng et al. (2010).

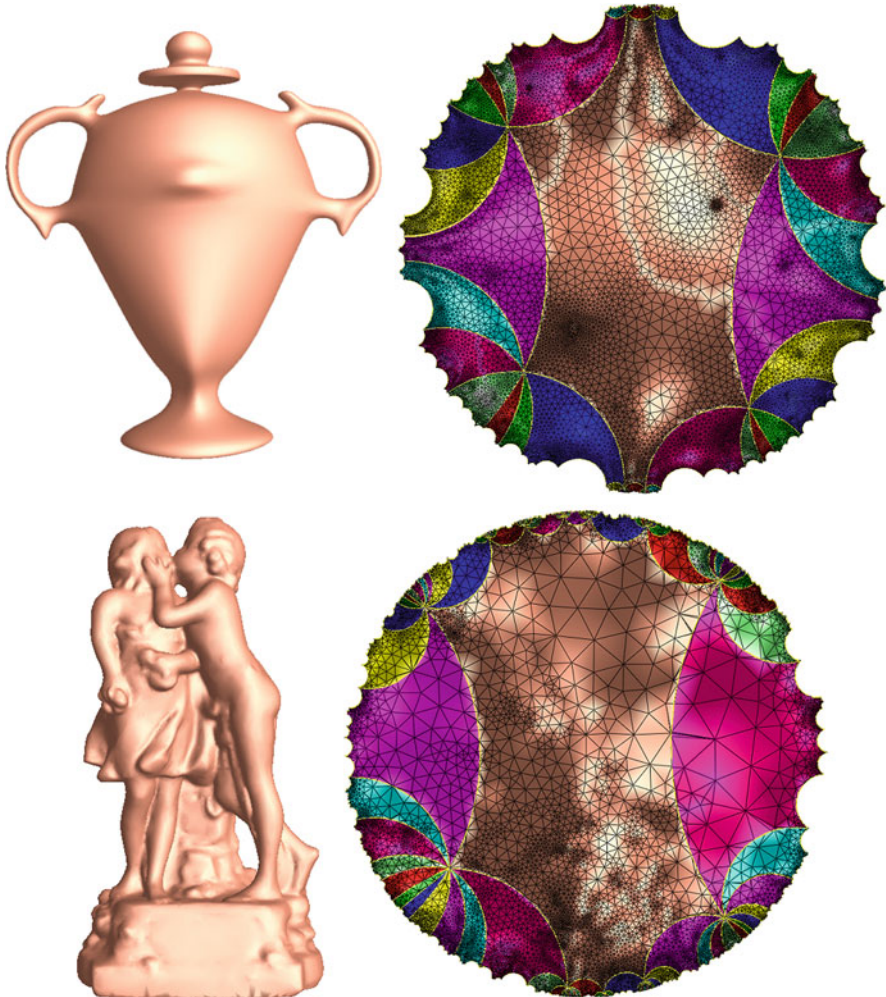


Fig. 26 Fuchsian groups of high genus surfaces

Surface Registration

3D deformable surface registration plays a fundamental role in computer vision. Given two surfaces (Σ_1, \mathbf{g}_1) and (Σ_2, \mathbf{g}_2) with the same topology and a set of landmarks $\{p_1, p_2, \dots, p_n\} \subset \Sigma_1$ and $\{q_1, q_2, \dots, q_n\} \subset \Sigma_2$, surface registration aims at finding a homeomorphism $f : \Sigma_1 \rightarrow \Sigma_2$, such that $f(p_k) = q_k$ for all landmarks $k = 1, 2, \dots, n$ and f minimizes the geometric and textural distortions.

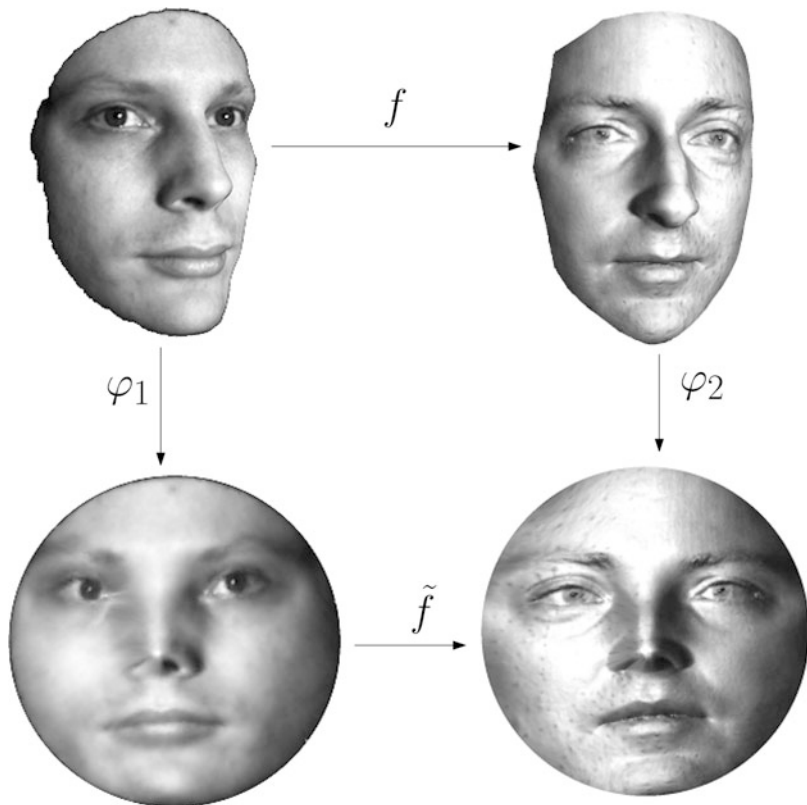


Fig. 27 Framework for 3D deformable surface registration

Registration Framework Figure 27 shows the framework for surface registration based on conformal geometry. By surface uniformization theorem, we can conformally map the surfaces onto canonical domains, the sphere, the Euclidean plane, or the hyperbolic plane (unit disk), $\varphi_k : \Sigma_k \rightarrow \mathbb{D}$, and then we construct a mapping $\tilde{f} : \mathbb{D} \rightarrow \mathbb{D}$ and then lift the planar map to the map between surfaces $f = \varphi_2^1 \circ \tilde{f} \circ \varphi_1$, which gives the registration result. In this way, we convert the surface registration problem to the 2D image registration problem. The conformal mappings are solely determined by the Riemannian metrics; therefore this method eliminates the ambiguity of rigid motions in \mathbb{R}^3 . Furthermore, it is much easier to compute planar mappings than surface mappings. If the desired mapping f is close to an isometry, then the planar mapping \tilde{f} is close to the identity map, and this greatly reduces the searching space for the algorithm.

Quasi-Conformal Map The planar mapping $\tilde{f} : \mathbb{D} \rightarrow \mathbb{D}$ satisfies the landmark constraints, $\tilde{f}(\varphi_1(p_k)) = \varphi_2(q_k), k = 1, 2, \dots, n$. We can use Teichmüller theory to find the desired quasi-conformal mapping.

If we know the Beltrami coefficient μ of the mapping, then by solving the Beltrami equation

$$\frac{\partial \tilde{f}}{\partial \bar{z}} = \mu \frac{\partial \tilde{f}}{\partial z},$$

we can obtain the map \tilde{f} . One way to solve the Beltrami equation is to construct an *auxiliary metric* \mathbf{g} , such that the mapping $\tilde{f} : (\mathbb{D}, dzd\bar{z}) \rightarrow (\mathbb{D}, dwd\bar{w})$ is quasi-conformal, where z and w are the complex parameters of the domain and the range, the exact same map under the auxiliary metric becomes a conformal map, $\tilde{f} : (\mathbb{D}, dzd\bar{z}) \rightarrow (\mathbb{D}, \mathbf{g})$. Since we know how to compute conformal maps, then we can find \tilde{f} . So the key step is to construct the auxiliary metric. Fortunately, the auxiliary metric can be easily constructed as

$$\mathbf{g}_\mu = |dz + \mu d\bar{z}|^2.$$

By using auxiliary metric method, we can solve the Beltrami equation and obtain the quasi-conformal map.

Teichmüller Map By controlling the Beltrami coefficient μ , we can fully control the homeomorphism; therefore, we can perform optimizations in the space of homeomorphisms. Suppose we want to minimize the angle distortion induced by the map, we can compute the Teichmüller map by an iterative procedure.

As shown in Fig. 28, we map the male and female facial surfaces onto the planar disk by Riemann mappings $\varphi_k : \Sigma_k \rightarrow \mathbb{D}$. Then we use Möbius transformations to map the nose tip to the origin, the line connecting the eye corners to be horizontal.

The algorithm is as follows: we compute a harmonic map $f_0 : \mathbb{D} \rightarrow \mathbb{D}$, with landmark constraints $f_0(\varphi_1(p_k)) = \varphi_2(q_k)$. Then we compute the Beltrami coefficient of f_0 , $\mu_0 = \partial_{\bar{z}} f_0 / \partial_z f_0$. Harmonic maps with constraints may not be diffeomorphic, and the neighborhoods of landmarks may have foldings; therefore $\|\mu_0\|_\infty$ may be greater than 1. Then we set $v_0 \leftarrow c_0 \mu_0 / |\mu_0|$, where c_0 is the mean of the norm of μ_0 .

At the k -th step, we construct an auxiliary metric $\mathbf{g}_k = |dz + v_{k-1} d\bar{z}|^2$; compute the harmonic map $f_k : (\mathbb{D}, \mathbf{g}_k) \rightarrow (\mathbb{D}, |dw|^2)$, with landmark constraints $f_k(\varphi_1(p_i)) = \varphi_2(q_i)$; compute the Beltrami coefficient $\mu_k = \partial_{\bar{z}} f_k / \partial_z f_k$; construct Beltrami coefficient $v_k \leftarrow c_k \mu_k / |\mu_k|$; and repeat this procedure, until it converges. Figure 28 shows the Teichmüller map between the faces. We use a circle-packing texture on the female face, which is pulled back to the male face by the map. We can see all the ellipses on the male face are with the same eccentricities; this shows the result map is close to a Teichmüller map. The convergence of $\{f_k\}$ to a Teichmüller map can be found in Lui et al. (2015).

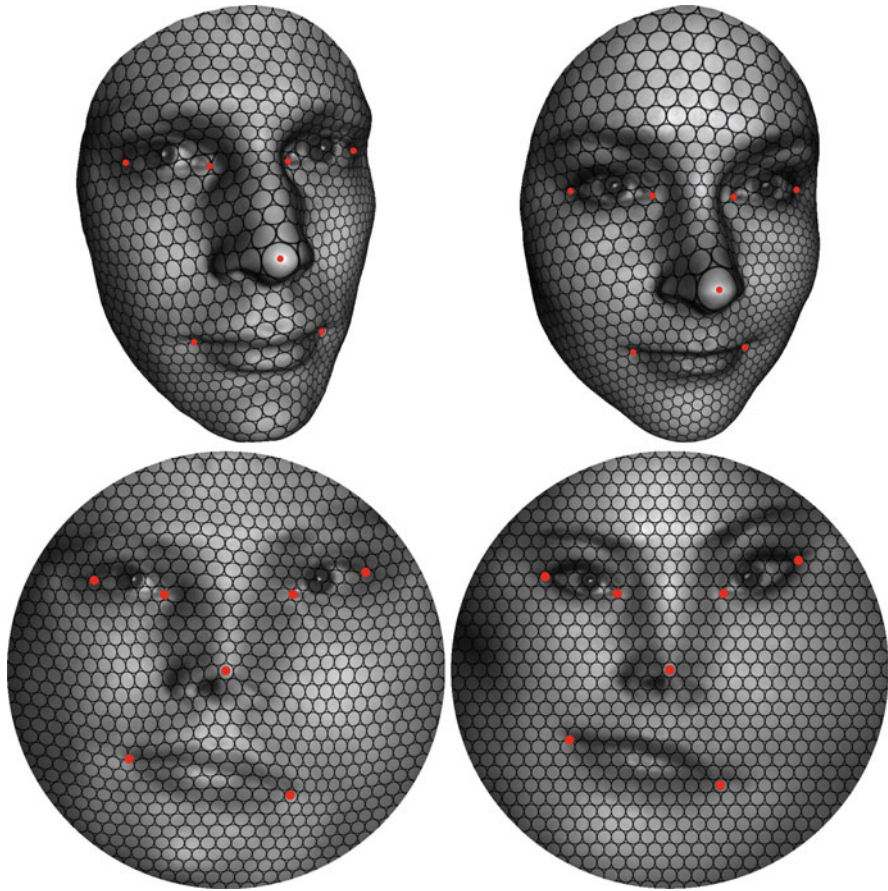


Fig. 28 Facial surface matching by a Teichmüller map; all ellipses have the same eccentricity

Registration Using Optimal Transport Map

Although conformal mappings preserve angles, it may distort the area drastically. As shown in Fig. 29, the armadillo surface is conformally mapped onto the planar disk, and the map induces large area distortions for the tubular shapes, such as the arms, head, tips of ears, and fingers as shown in the bottom row. This may lead to the inaccuracy and the instability of the registration algorithm.

We can conquer this difficulty by composing the conformal map by an optimal transport map (Su et al. 2015), as shown in the middle row, such that the mapping preserves the area element from the surface to the planar domain. In the images of the area-preserving mappings, the finger tips and the head region are enlarged significantly. The armadillo changes the postures, but the two surfaces are close to be isometric; therefore the registration mapping is close to the identity of the unit

disk. As shown in the top row of Fig. 29, the algorithm automatically registers each finger tip to the corresponding one without any mismatching. This demonstrates the accuracy of the registration.

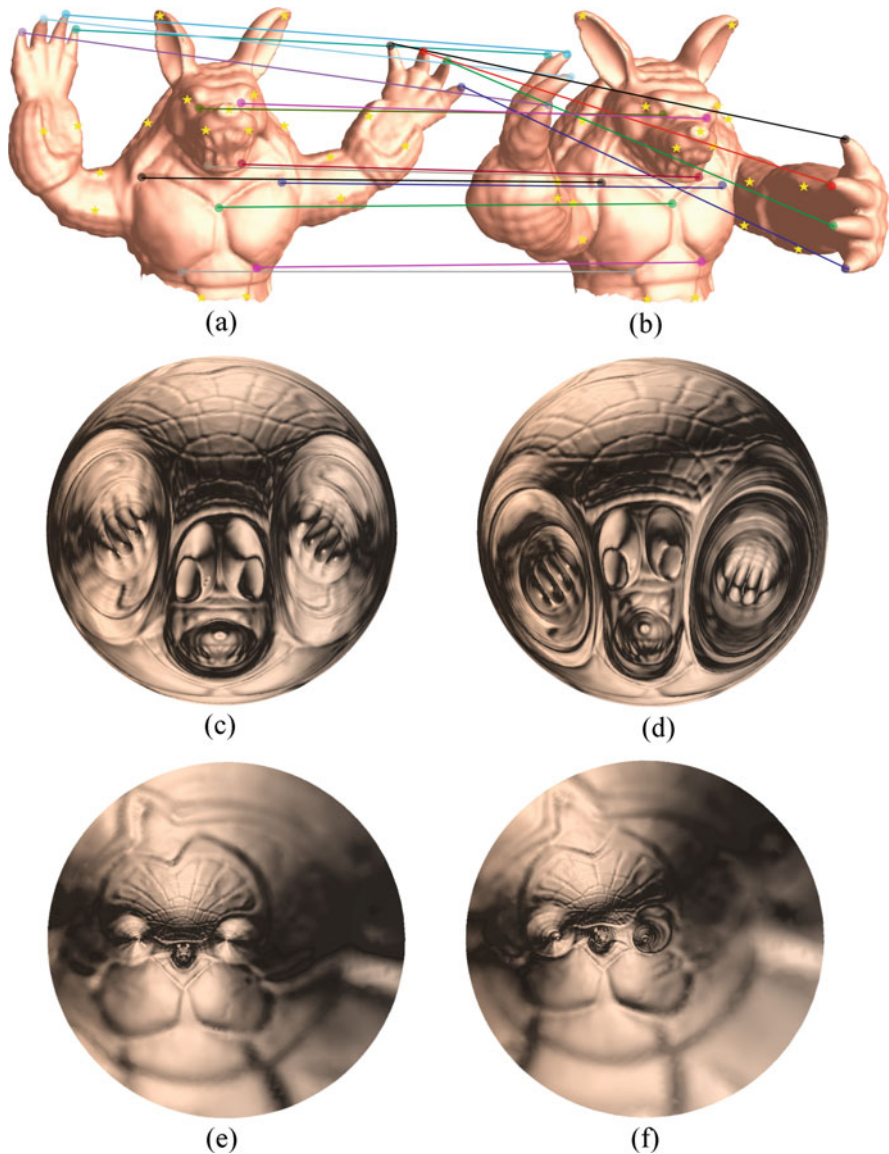


Fig. 29 Surface registration based on conformal and optimal transport maps. (a) Armadillo #1. (b) Armadillo #2. (c) Area-preserving map #1. (d) Area-preserving map #2. (e) Conformal map #1. (f) Conformal map #2

Dynamic Surface Tracking Surface registration methods can be generalized to surface tracking. Given a sequence of facial surfaces with dynamic expression changes, the Teichmüller map can be used to find natural diffeomorphisms among them. The trajectories of the feature points represent the facial expression, which can be transferred to other models for animation purposes.

Given a sequence facial surface $\{(\Sigma_k, \mathbf{g}_k)\}_{k=1}^n$, we locate the feature points for each frame $\{(p_1^k, p_2^k, \dots, p_m^k)\}_{k=1}^n$, and then we compute Riemann mappings $\varphi_k : \Sigma_k \rightarrow \mathbb{D}$. Our goal is to find a sequence homeomorphisms $f_k : \varphi_k(\Sigma_k) \rightarrow \varphi_{k+1}(\Sigma_{k+1})$, with the landmark constraints $f_k(\varphi_k(p_i^k)) = \varphi_{k+1}(p_i^{k+1})$, $k = 1, \dots, n-1, i = 1, \dots, m$. Each map f_k is a quasi-conformal map with Beltrami coefficient $\mu_k, \|\mu_k\|_\infty < 1$. The Beltrami coefficients can be obtained by optimizing the following energy:

$$\int_{\mathbb{D}} |\nabla \mu_k|^2 dA + \int_{\mathbb{D}} |\mu_k - \mu_{k-1}|^2 dA + \int_{\mathbb{D}} |H(p) - H \circ f^{\mu_k}(p)|^2 + |c(p) - c \circ f^{\mu_k}(p)|^2 dp$$

where $H(\cdot)$ and $c(\cdot)$ represent the mean curvature and the texture color of the surface. Figure 30 demonstrates an expression tracking result; the blue quadrilateral mesh is attached to the first facial surface and moves along with it. The trajectories of the vertices of the blue mesh represent the expression (Yu et al. 2017). The facial expression tracking technique plays an important role in the movie industry. More algorithmic details and applications of quasi-conformal mappings can be found in Lui et al. (2010, 2012), Ng et al. (2014), and Wong and Zhao (2014).

Medical Imaging

Conformal geometry has been applied to many fields in medical imaging. For example, in the field of brain imaging, it is crucial to register different brain cortex surfaces reconstructed from MRI or CT images. Because brain surfaces are

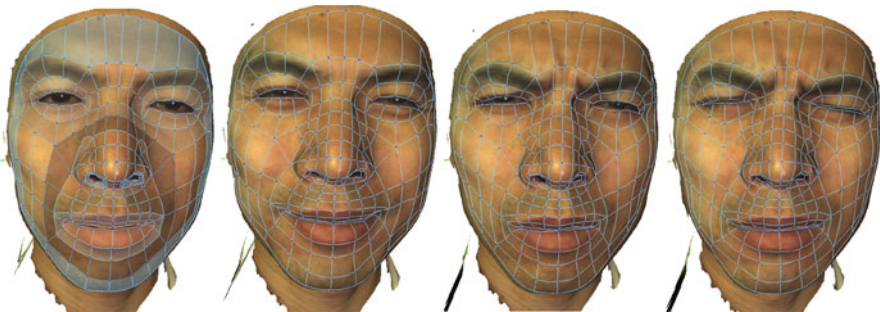


Fig. 30 Facial expression tracking using quasi-conformal mapping

highly convoluted, and different people have different anatomic structures, it is quite challenging to find proper matching between cortex surfaces. Cortex surfaces are topological spheres and can be uniformized onto the unit sphere conformally. Figure 31 illustrates this solution (Gu and Yau 2007; Gu et al. 2004) by mapping brains to the unit sphere in a canonical way. Then, by finding an automorphism of the sphere, the registration between cortical surfaces can be easily established.

Many conformal geometric methods have been applied for studying brain morphology. For example, the area distortion factor (conformal factor) induced by the conformal brain mapping defines a measure on the unit sphere. Different cortical surfaces give different spherical measures. By computing the Wasserstein distance among the spherical measures, we can define a global shape distance among cortical surfaces. Another method is shown in Fig. 32. The major landmark curves (sulci and gyri) on the cortical surfaces are located, and then the surface is sliced open along these landmarks and conformally mapped onto a circle domain. The centers and radii of the inner circles give the conformal module of the surface. The conformal module can be treated as the fingerprint of the cortical surface and used for classification and comparison. These methods have been applied for neurological disease diagnosis, such as Alzheimer's disease, autism, Williams syndrome, and so on. More applications and algorithmic details can be found in Wang et al. (2005, 2007) and Peng et al. (2015).

Colon cancer is the third most common cause of cancer-related death in the United States. The most effective way to prevent colon cancer is through colonoscopy. Conventional colonoscopy is invasive and may cause complications. Virtual colonoscopy is less invasive and with fewer complications. In virtual colonoscopy (Zeng and Gu 2013), the colon surface is reconstructed from CT images and analyzed using the geometric method. As shown in Fig. 33 left frame, a colon surface has many haustral folds in anatomy, and when the polyps are hidden

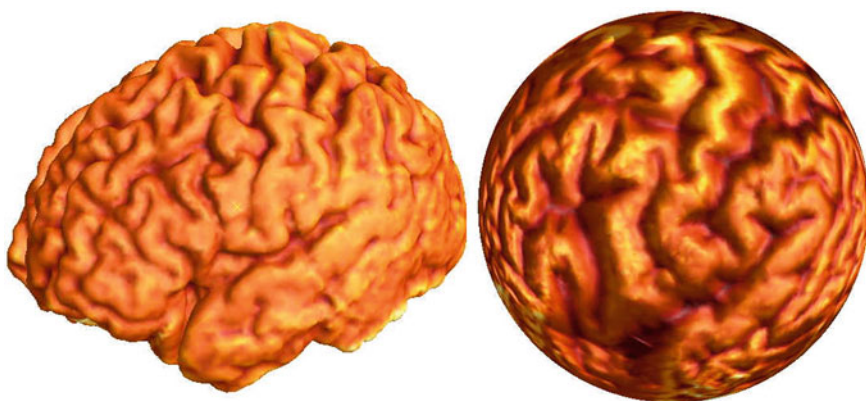


Fig. 31 Brain spherical conformal mapping

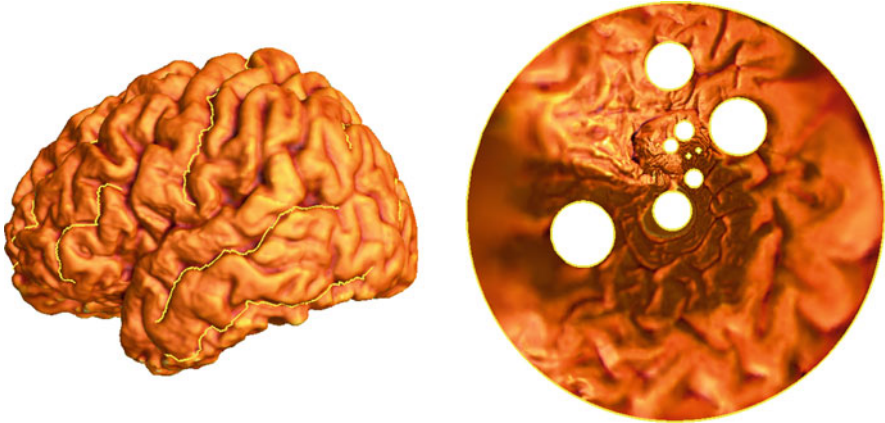


Fig. 32 Brain morphology study using conformal module

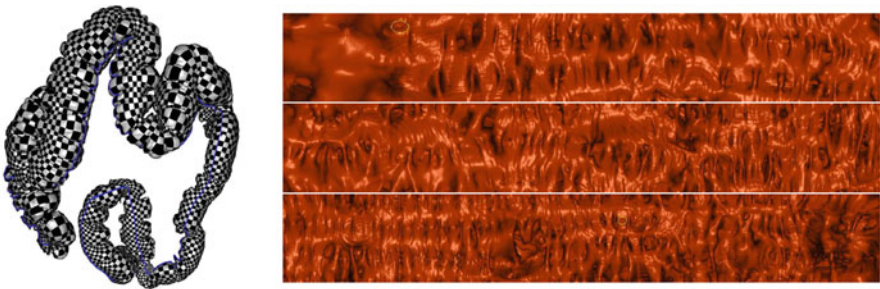


Fig. 33 Colon conformal flattening

in these folds, they are hard to be located and recognized. By using conformal geometric methods, one can flatten the whole colon surface onto a planar rectangle, as shown in the right frame. Then all the haustral folds are expanded, all polyps are exposed, and abnormalities can be found efficiently on the planar image. Furthermore, in practice, the colon surface will be scanned twice with supine and prone positions. Because the colon surface is highly elastic, there will be large deformations between the two scans. Conformal colon flattening can be applied to find a good registration between the supine and prone colon surfaces (Zeng et al. 2010; Zeng and Gu 2013). Today, the conformal colon flattening technique has already been widely used in clinical practice. More applications and algorithmic details for virtual colonoscopy can be found in Saad Nadeem et al. (2017) and Ma et al. (2019).

Conclusion

This chapter introduces the fundamental concepts and theorems in computational conformal geometry, such as discrete surface Ricci flow theory and discrete uniformization theorem; then explains main computational algorithms based on harmonic maps, Holomorphic differentials, and surface Ricci flow; finally, demonstrates the direct applications, including shape classification and surface registration in computer vision, brain mapping, and virtual colonoscopy in medical imaging.

Computational conformal geometry has emerged as an interdisciplinary field between mathematics and computer science. The theories and algorithms have played important roles in many engineering and medical fields. We expect to see more exciting breakthroughs in near future.

References

- Bobenko, A.I., Pinkall, U., Springborn, B.A.: Discrete conformal maps and ideal hyperbolic polyhedra. *Geom. Topol.* **19**(4), 2155–2215 (2015)
- Chen, W., Zhang, M., Lei, N., Gu, D.X.: Dynamic unified surface ricci flow. *Geom. Imag. Comput.* **3**(1), 31–56 (2016)
- Chow, B., Luo, F.: Combinatorial Ricci flows on surfaces. *J. Differ. Geom.* **63**(1), 97–129 (2003)
- de Verdière, Y.C.: Un principe variationnel pour les empilements de cercles. *Invent. Math.* **104**(3), 655–669 (1991)
- Glickenstein, D.: Discrete conformal variations and scalar curvature on piecewise flat two- and three-dimensional manifolds. *J. Differ. Geom.* **87**(2), 201–237 (2011)
- Gotsman, C., Gu, X., Sheffer, A.: Fundamentals of spherical parameterization for 3d meshes. *ACM Trans. Graph. (TOG)* **22**(3), 358–363 (2003)
- Gu, X., Guo, R., Luo, F., Sun, J., Wu, T.: A discrete uniformization theorem for polyhedral surfaces (II). *J. Differ. Geom. (JDG)* **109**(3), 431–466 (2018a)
- Gu, X., Luo, F., Sun, J., Wu, T.: A discrete uniformization theorem for polyhedral surfaces (I). *J. Differ. Geom. (JDG)* **109**(2), 223–256 (2018b)
- Gu, X., Luo, F., Wu, T.: Convergence of discrete conformal geometry and computation of uniformization maps. *Asian J. Math. (AJM)* **23**(1), 21–34 (2019)
- Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imag. (TMI)* **23**(8), 949–958 (2004)
- Gu, X., Yau, S.-T.: Global conformal surface parameterization. In: *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 127–137. Eurographics Association (2003)
- Gu, X., Yau, S.-T.: *Computational Conformal Geometry*. Advanced Lectures in Mathematics, vol. 3. International Press and Higher Education Press, Boston (2007)
- Gu, X., Yau, S.-T.: *Computational Conformal Geometry – Theory*. International Press and Higher Education Press, Boston (2020)
- Gu, X.D., Zeng, W., Luo, F., Yau, S.-T.: Numerical computation of surface conformal mappings. *Comput. Methods Funct. Theory* **11**(2), 747–787 (2012)
- Jin, M., Kim, J., Luo, F., Gu, X.: Discrete surface ricci flow. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **14**(5), 1030–1043 (2008)
- Jin, M., Wang, Y., Gu, X., Yau, S.-T., et al.: Optimal global conformal surface parameterization for visualization. *Commun. Inf. Syst.* **4**(2), 117–134 (2004)

- Jin, M., Zeng, W., Ding, N., Gu, X.: Computing fenchel-nielsen coordinates in teichmuller shape space. *Commun. Inf. Syst.* **9**(2), 213–234 (2009a)
- Jin, M., Zeng, W., Luo, F., Gu, X.: Computing teichmuller shape space. *IEEE Trans. Vis. Comput. Graph.* **15**(3), 504–517 (2009b)
- Lei, N., Zheng, X., Jiang, J., Lin, Y.-Y., Gu, D.X.: Quadrilateral and hexahedral mesh generation based on surface foliation theory. *Comput. Methods Appl. Mech. Eng.* **316**, 758–781 (2017a)
- Lei, N., Zheng, X., Luo, Z., Gu, D.X.: Quadrilateral and hexahedral mesh generation based on surface foliation theory II. *Comput. Methods Appl. Mech. Eng.* **321**, 406–426 (2017b)
- Lui, L.M., Gu, X., Yau, S.-T.: Convergence of an iterative algorithm for teichmüller maps via harmonic energy optimization. *Math. Comput.* **84**(296), 2823–2842 (2015)
- Lui, L.M., Wong, T.W., Zeng, W., Gu, X., Thompson, P.M., Chan, T.F., Yau, S.T.: Detection of shape deformities using yamabe flow and beltrami coefficients. *Inverse Probl. Imag.* **4**(2), 311–333 (2010)
- Lui, L.M., Wong, T.W., Zeng, W., Gu, X., Thompson, P.M., Chan, T.F., Yau, S.-T.: Optimization of surface registrations using beltrami holomorphic flow. *J. Sci. Comput.* **50**(3), 557–585 (2012)
- Luo, F.: Combinatorial Yamabe flow on surfaces. *Commun. Contemp. Math.* **6**(5), 765–780 (2004)
- Luo, F., Gu, X., Dai, J.: *Variational Principles for Discrete Surfaces. Advanced Lectures in Mathematics*, vol. 4. International Press and Higher Education Press, Boston (2007)
- Ma, M., Marino, J., Nadeem, S., Gu, X.: Supine to prone colon registration and visualization based on optimal mass transport. *Graph. Models* **104**, 101031 (2019)
- Ng, T.C., Gu, X., Lui, L.M.: Computing extremal teichmüller map of multiply-connected domains via beltrami holomorphic flow. *J. Sci. Comput.* **60**(2), 249–275 (2014)
- Peng, H., Wang, X., Duan, Y., Frey, S.H., Gu, X.: Brain morphometry on congenital hand deformities based on teichmüller space theory. *Comput.-Aided Des.* **58**, 84–91 (2015)
- Rodin, B., Sullivan, D.: The convergence of circle packings to the Riemann mapping. *J. Differ. Geom.* **26**(2), 349–360 (1987)
- Roček, M., Williams, R.M.: Quantum Regge calculus. *Phys. Lett. B* **104**(1), 31–37 (1981)
- Saad Nadeem, J.M., Gu, X., Kaufman, A.: Corresponding supine and prone colon visualization using eigenfunction analysis and fold modeling. *IEEE Trans. Vis. Comput. Graph.* **23**(1), 751–760 (2017)
- Shi, R., Zeng, W., Su, Z., Jiang, J., Damasio, H., Lu, Z., Wang, Y., Yau, S.-T., Gu, X.: Hyperbolic harmonic mapping for surface registration. *IEEE Trans. Pattern Anal. Mach. Intell.* (2016)
- Su, Z., Wang, Y., Shi, R., Zeng, W., Sun, J., Luo, F., Gu, X.: Optimal mass transport for shape matching and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(11), 2246–2259 (2015)
- Thurston, W.P.: *Three-Dimensional Geometry and Topology. Vol.1.* Princeton Mathematical Series, vol. 35. Princeton University Press, Princeton (1997)
- Wang, Y., Gu, X., Hayashi, K.M., Chan, T.F., Thompson, P.M., Yau, S.-T.: Brain surface conformal parameterization. In: *Proceedings of the Eighth IASTED International Conference, Computer Graphics and Imaging*. Honolulu, Hawaii, pp. 76–81 (2005)
- Proceedings of the Eighth IASTED International Conference, Computer Graphics and Imaging*, August, 2005, Honolulu, Hawaii, USA. Before references, a pa
- Wang, Y., Lui, L.M., Gu, X., Hayashi, K.M., Chan, T.F., Toga, A.W., Thompson, P.M., Yau, S.-T.: Brain surface conformal parameterization using riemann surface structure. *IEEE Trans. Med. Imag.* **26**(6), 853–865 (2007)
- Wong, T.W., Zhao, H.-K.: Computation of quasi-conformal surface maps using discrete beltrami flow. *SIAM J. Imag. Sci.* **7**(4), 2675–2699 (2014)
- Yin, X., Dai, J., Yau, S.-T., Gu, X.: Slit map: linear conformal parameterization for multiply connected domains. *Comput.-Aided Geom. Des. (CAGD)* **4975**, 410–422 (2008)
- Yu, X., Lei, N., Wang, Y., Gu, X.: Intrinsic 3D dynamic surface tracking based on dynamic ricci flow and teichmuller map. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5390–5398 (2017)
- Zeng, W., Gu, X.: *Ricci Flow for Shape Analysis and Surface Registration – Theories, Algorithms and Applications.* Springer Briefs in Mathematics. Springer, Springer (2013)

- Zeng, W., Marino, J., Gurijala, K.C., Gu, X., Kaufman, A.: Supine and prone colon registration using quasi-conformal mapping. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1348–1357 (2010)
- Zeng, W., Samaras, D., Gu, X. Ricci flow for 3D shape analysis. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **32**(4), 662–677 (2010)
- Zeng, W., Yin, X., Zhang, M., Luo, F., Gu, X.: Generalized Koebe’s method for conformal mapping multiply connected domains. In: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, pp. 89–100. ACM (2009)
- Zhang, M., Guo, R., Zeng, W., Luo, F., Yau, S.-T., Gu, X.: The unified discrete surface ricci flow. *Graph. Models* **76**(5), 321–339 (2014)



Sebastian Neumayer and Gabriele Steidl

Contents

Introduction	1792
Preliminaries	1794
Discrepancies	1797
Optimal Transport and Wasserstein Distances	1804
Regularized Optimal Transport	1805
Sinkhorn Divergence	1815
Numerical Approach and Examples	1818
Conclusions	1823
Basic Theorems	1824
References	1824

Abstract

A common way to quantify the “distance” between measures is via their discrepancy, also known as maximum mean discrepancy (MMD). Discrepancies are related to Sinkhorn divergences \mathcal{S}_ε with appropriate cost functions as $\varepsilon \rightarrow \infty$. In the opposite direction, if $\varepsilon \rightarrow 0$, Sinkhorn divergences approach another important distance between measures, namely, the Wasserstein distance or more generally optimal transport “distance.” In this chapter, we investigate the limiting process for arbitrary measures on compact sets and Lipschitz continuous cost functions. In particular, we are interested in the behavior of the corresponding optimal potentials $\hat{\varphi}_\varepsilon$, $\hat{\psi}_\varepsilon$, and $\hat{\varphi}_K$ appearing in the dual formulation of the Sinkhorn divergences and discrepancies, respectively. While part of the results is known, we provide rigorous proofs for some relations which we have not found in this generality in the literature. Finally, we demonstrate the limiting process

S. Neumayer (✉) · G. Steidl (✉)
Institute of Mathematics, TU Berlin, Berlin, Germany
e-mail: neumayer@math.tu-berlin.de; steidl@math.tu-berlin.de

by numerical examples and show the behavior of the distances when used for the approximation of measures by point measures in a process called dithering.

Keywords

Discrepancies · Duality · Interpolation · Optimal transport · Sinkhorn divergence

Introduction

The approximation of probability measures based on their discrepancies is a well-examined problem in approximation and complexity theory (Kuipers and Niederreiter 1974; Matousek 2010; Novak and Wozniakowski 2010). Discrepancies appear in a wide range of applications, e.g., in the derivation of quadrature rules (Novak and Wozniakowski 2010), the construction of designs (Delsarte et al. 1977), image dithering, and representation (Ehler et al. 2019; Gräf et al. 2013; Schmaltz et al. 2010; Teuber et al. 2011); see also Fig. 1, generative adversarial networks (Dziugaite et al. 2015) and multivariate statistical testing (Fernández et al. 2008; Gretton et al. 2007, 2012). In the last two applications, they are also called kernel-based maximum mean discrepancies (MMDs).

On the other hand, optimal transport (OT) “distances” and in particular Wasserstein distances became very popular for tackling various problems in imaging sciences, graphics, or machine learning (Cuturi and Peyré 2019). There exists a large amount of papers both on the theory and applications of OT, for image dithering

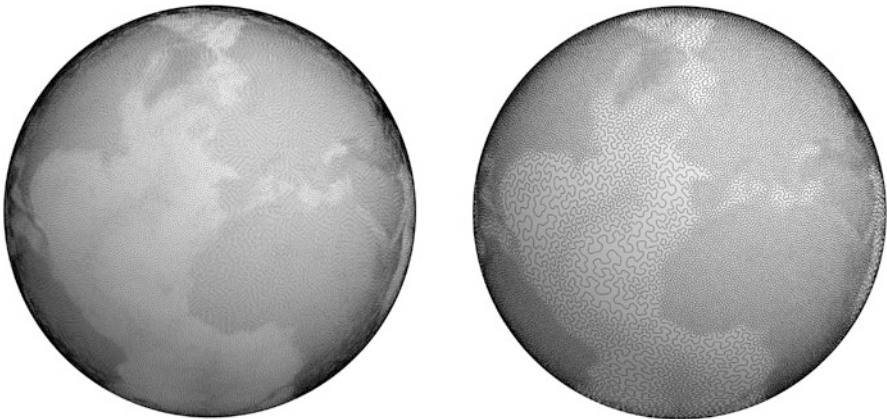


Fig. 1 Approximation of a measure on \mathbb{S}^2 by an empirical measure (Gräf et al. 2013) (left) and a measure supported on a curve (Ehler et al. 2019) (right) using discrepancies as objective function to minimize

with Wasserstein distances; see, e.g., Chauffert et al. (2017), Goes et al. (2012), and Lebrat et al. (2019).

Recently, regularized versions of OT for an efficient numerical treatment, known as Sinkhorn divergences (Cuturi 2013), were used as replacement of OT in data science. Note that such regularization ideas are also investigated in the earlier works (Rüschendorf 1995; Sinkhorn 1964; Wilson 1969; Yule 1912). For appropriately related transport cost functions and discrepancy kernels, the Sinkhorn divergences interpolate between the OT distance if the parameter goes to zero and the discrepancy if it goes to infinity (Feydy et al. 2019). In this chapter, the convergence behavior is examined for general measures on compact sets. Since cost functions applied in practice are mainly Lipschitz, we restrict our attention to such costs. This simplifies some proofs, since the theorem of Arzelà–Ascoli can be utilized. To make the paper self-contained, we provide most of the proofs although some of them are not novel and the corresponding papers are cited in the context. For estimating approximation rates when approximating measures by those of certain subsets (see, e.g., Chevallier (2018), Ehler et al. (2019), Genevay et al. (2019), and Novak and Wozniakowski (2010)), the dual form of the discrepancy, respectively, of the (regularized) Wasserstein distance, plays an important role. Therefore, we are interested in the properties of the optimal dual potentials for varying regularization parameters. In Proposition 5, we prove that the optimal dual potentials converge uniformly to certain functions as $\varepsilon \rightarrow \infty$. Then, in Corollary 2, we see that the normalized difference of these limiting functions coincides with the optimal potential in the dual form of the discrepancy if the cost function and the kernel are appropriately related. This behavior is underlined by a numerical example.

This chapter is organized as follows: section “Preliminaries” recalls basic results on measures, the Kullback–Leibler (KL) divergence, and from convex analysis. In section “Discrepancies”, we introduce discrepancies, in particular their dual formulation. Since these rely on positive definite kernels, we have a closer look at positive definite and conditionally positive definite kernels. Optimal transport and in particular Wasserstein distances are considered in section “Optimal Transport and Wasserstein Distances”. In section “Regularized Optimal Transport”, we investigate the limiting processes for the KL-regularized OT distances, when the regularization parameter goes to zero or infinity. Some results in Proposition 2 are novel in this generality; Proposition 5 seems to be new as well. Remark 3 highlights why the KL divergence should be preferred as regularizer instead of the (neg)-entropy when dealing with non-discrete measures. KL-regularized OT does not fulfill $OT_\varepsilon(\mu, \mu) = 0$, which motivates the definition of the Sinkhorn divergence S_ε in section “Sinkhorn Divergence”. Further, we prove Γ -convergence to the discrepancy as $\varepsilon \rightarrow \infty$ if the cost function of the Sinkhorn divergence is adapted to the kernel defining the discrepancy. Section “Numerical Approach and Examples” underlines the results on the limiting process by numerical examples. Further, we provide an example on the dithering of the standard Gaussian when Sinkhorn divergences with respect to different regularization parameters ε are involved. Finally, conclusions and directions of future research are given in section “Conclusions”.

Preliminaries

Measures Let \mathbb{X} be a compact Polish space (separable, complete metric space) with metric $\text{dist}_{\mathbb{X}}$. By $\mathcal{B}(\mathbb{X})$, we denote the Borel σ -algebra on \mathbb{X} and by $\mathcal{M}(\mathbb{X})$ the linear space of all finite signed Borel measures on \mathbb{X} , i.e., all $\mu : \mathcal{B}(\mathbb{X}) \rightarrow \mathbb{R}$ satisfying $\mu(\mathbb{X}) < \infty$ and for any sequence $\{B_k\}_{k \in \mathbb{N}} \subset \mathcal{B}(\mathbb{X})$ of pairwise disjoint sets the relation $\mu(\cup_{k=1}^{\infty} B_k) = \sum_{k=1}^{\infty} \mu(B_k)$. In the following, the subset of nonnegative measures is denoted by $\mathcal{M}^+(\mathbb{X})$. The *support of a measure* μ is defined as the closed set

$$\text{supp}(\mu) := \{x \in \mathbb{X} : B \subset \mathbb{X} \text{ open, } x \in B \implies \mu(B) > 0\}.$$

The *total variation* measure of $\mu \in \mathcal{M}(\mathbb{X})$ is defined by

$$|\mu|(B) := \sup \left\{ \sum_{k=1}^{\infty} |\mu(B_k)| : \bigcup_{k=1}^{\infty} B_k = B, B_k \text{ pairwise disjoint} \right\}.$$

With the norm $\|\mu\|_{\mathcal{M}} = |\mu|(\mathbb{X})$, the space $\mathcal{M}(\mathbb{X})$ becomes a Banach space. By $C(\mathbb{X})$, we denote the Banach space of continuous real-valued functions on \mathbb{X} equipped with the norm $\|\varphi\|_{C(\mathbb{X})} := \max_{x \in \mathbb{X}} |\varphi(x)|$. The space $\mathcal{M}(\mathbb{X})$ can be identified via Riesz' representation theorem with the dual space of $C(\mathbb{X})$, and the weak-* topology on $\mathcal{M}(\mathbb{X})$ gives rise to the *weak convergence of measures*. More precisely, a sequence $\{\mu_k\}_{k \in \mathbb{N}} \subset \mathcal{M}(\mathbb{X})$ converges *weakly* to μ , and we write $\mu_k \rightharpoonup \mu$, if

$$\lim_{k \rightarrow \infty} \int_{\mathbb{X}} \varphi \, d\mu_k = \int_{\mathbb{X}} \varphi \, d\mu \quad \text{for all } \varphi \in C(\mathbb{X}).$$

For a nonnegative, finite measure μ and $p \in [1, \infty)$, let $L^p(\mathbb{X}, \mu)$ be the Banach space (of equivalence classes) of complex-valued functions with norm

$$\|f\|_{L^p(\mathbb{X}, \mu)} = \left(\int_{\mathbb{X}} |f|^p \, d\mu \right)^{\frac{1}{p}} < \infty.$$

A measure $\nu \in \mathcal{M}(\mathbb{X})$ is *absolutely continuous* with respect to μ , and we write $\nu \ll \mu$ if for every $A \in \mathcal{B}(\mathbb{X})$ with $\mu(A) = 0$ we have $\nu(A) = 0$. If $\mu, \nu \in \mathcal{M}^+(\mathbb{X})$ satisfy $\nu \ll \mu$, then the *Radon-Nikodym derivative* $\sigma_{\nu} \in L^1(\mathbb{X}, \mu)$ (also denoted by $\frac{d\nu}{d\mu}$) exists and $\nu = \sigma_{\nu} \mu$. Further, $\mu, \nu \in \mathcal{M}(\mathbb{X})$ are *mutually singular*, and we write $\mu \perp \nu$ if two disjoint sets $X_{\mu}, X_{\nu} \in \mathcal{B}(\mathbb{X})$ exist such that $\mathbb{X} = X_{\mu} \cup X_{\nu}$ and for every $A \in \mathcal{B}(\mathbb{X})$ we have $\mu(A) = \mu(A \cap X_{\mu})$ and $\nu(A) = \nu(A \cap X_{\nu})$. For any $\mu, \nu \in \mathcal{M}^+(\mathbb{X})$, there exists a unique *Lebesgue decomposition* of μ with respect to ν given by $\mu = \sigma_{\mu} \nu + \mu^{\perp}$, where $\sigma \in L^1(\mathbb{X}, \nu)$ and $\mu^{\perp} \perp \nu$.

By $\mathcal{P}(\mathbb{X})$, we denote the set of Borel probability measures on \mathbb{X} , i.e., nonnegative Borel measures with $\mu(\mathbb{X}) = 1$. This set is weakly compact, i.e., compact with respect to the weak-* topology. Note that there is an ambiguity in the notation as the above usual weak-* convergence is called weak convergence in stochastics. In section “Optimal Transport and Wasserstein Distances”, we introduce a metric on $\mathcal{P}(\mathbb{X})$ such that it becomes a Polish space.

Convex analysis The following can be found, e.g., in Bredies and Lorenz (2011). Let V be a real Banach space with dual V^* , i.e., the space of real-valued continuous linear functionals on V . We use the notation $\langle v, x \rangle = v(x)$, $v \in V^*$, $x \in V$. For $F: V \rightarrow (-\infty, +\infty]$, the domain of F is given by $\text{dom}F := \{x \in V : F(x) \in \mathbb{R}\}$. If $\text{dom}F \neq \emptyset$, then F is called proper. The subdifferential of $F: V \rightarrow (-\infty, +\infty]$ at a point $x_0 \in \text{dom}F$ is defined as

$$\partial F(x_0) := \{v \in V^* : F(x) \geq F(x_0) + \langle v, x - x_0 \rangle\},$$

and $\partial F(x_0) = \emptyset$ if $x_0 \notin \text{dom}F$. The Fenchel conjugate $F^*: V^* \rightarrow (-\infty, +\infty]$ is given by

$$F^*(v) = \sup_{x \in V} \{\langle v, x \rangle - F(x)\}.$$

If $F: V \rightarrow (-\infty, +\infty]$ is convex and lower semicontinuous (lsc) at $x \in \text{dom}F$, then

$$v \in \partial F(x) \iff x \in \partial F^*(v). \tag{1}$$

By $\Gamma_0(V)$, we denote the set of proper, convex, lsc functions mapping from V to $(-\infty, +\infty]$. Let W be another real Banach space. Then, for $F \in \Gamma_0(V)$, $G \in \Gamma_0(W)$ and a linear, bounded operator $A: V \rightarrow W$ with the property that there exists $x \in \text{dom}F$ such that G is continuous at Ax , the following Fenchel-Rockafellar duality relation is fulfilled

$$\sup_{x \in V} \{-F(-x) - G(Ax)\} = \inf_{w \in W^*} \{F^*(A^*w) + G^*(w)\}; \tag{2}$$

see Ekeland and Témam (1999, Thm. 4.1, p. 61), where we consider

$$\sup_{x \in V} \{-F(-x) - G(Ax)\} = - \inf_{x \in V} \{F(-x) + G(Ax)\}$$

as primal problem with respect to the notation in Ekeland and Témam (1999). If the optimal (primal) solution \hat{x} exists, it is related to any optimal (dual) solution \hat{w} by

$$A\hat{x} \in \partial G^*(\hat{w}); \tag{3}$$

see Ekeland and Témam (1999, Prop. 4.1).

Kullback-Leibler divergence A function $f : [0, +\infty) \rightarrow [0, +\infty]$ is called *entropy function*, if it is convex, lsc, and $\text{dom } f \cap (0, +\infty) \neq \emptyset$. The corresponding recession constant is given by $f'_\infty = \lim_{x \rightarrow \infty} \frac{f(x)}{x}$. For every $\mu, \nu \in \mathcal{M}^+(\mathbb{X})$ with Lebesgue decomposition $\mu = \sigma_\mu \nu + \mu^\perp$, the *f-divergence* is defined as

$$D_f(\mu, \nu) = \int_{\mathbb{X}} f \circ \sigma_\mu \, d\nu + f'_\infty \mu^\perp(\mathbb{X}). \tag{4}$$

In case that $f'_\infty = \infty$ and $\mu^\perp(\mathbb{X}) = 0$, we make the usual convention $\infty \cdot 0 = 0$. The *f-divergence* fulfills $D_f(\mu, \nu) \geq 0$ for all $\mu, \nu \in \mathcal{M}^+(\mathbb{X})$ and neither is in general symmetric nor satisfies a triangle inequality. The associated mapping $D_f : \mathcal{M}^+(\mathbb{X}) \times \mathcal{M}^+(\mathbb{X}) \rightarrow [0, +\infty]$ is jointly convex and weakly lsc; see Liero et al. (2018, Cor. 2.9). The *f-divergence* can be written in the dual form

$$D_f(\mu, \nu) = \sup_{\varphi \in C(\mathbb{X})} \int_{\mathbb{X}} \varphi \, d\mu - \int_{\mathbb{X}} f^* \circ \varphi \, d\nu;$$

see Liero et al. (2018, Rem. 2.10). Hence, $D_f(\cdot, \nu)$ is the Fenchel conjugate of $H : C(\mathbb{X}) \rightarrow \mathbb{R}$ given by $H(\varphi) := \int_{\mathbb{X}} f^* \circ \varphi \, d\nu$. If f^* is differentiable, we directly deduce from (1) that

$$\varphi \in \partial_\mu D_f(\mu, \nu) \iff \mu = \nabla H(\varphi) \iff \mu = \nabla f^* \circ \varphi \, \nu. \tag{5}$$

In the following, we focus on the *Shannon-Boltzmann entropy* function and its Fenchel conjugate given by

$$f(x) = x \log(x) - x + 1 \quad \text{and} \quad f^*(x) = \exp(x) - 1$$

with the agreement $0 \log 0 = 0$. The corresponding *f-divergence* is the *Kullback-Leibler divergence* $\text{KL} : \mathcal{M}^+(\mathbb{X}) \times \mathcal{M}^+(\mathbb{X}) \rightarrow [0, +\infty]$. For $\mu, \nu \in \mathcal{M}^+(\mathbb{X})$ with existing Radon-Nikodym derivative $\sigma_\mu = \frac{d\mu}{d\nu}$ of μ with respect to ν , formula (4) can be written as

$$\text{KL}(\mu, \nu) := \int_{\mathbb{X}} \log(\sigma_\mu) \, d\mu + \nu(\mathbb{X}) - \mu(\mathbb{X}). \tag{6}$$

In case that the above Radon-Nikodym derivative does not exist, (4) implies $\text{KL}(\mu, \nu) = +\infty$. For $\mu, \nu \in \mathcal{P}(\mathbb{X})$, the last two summands in (6) cancel each other. Hence, we have for discrete measures $\mu = \sum_{j=1}^n \mu_j \delta_{x_j}$ and $\nu = \sum_{j=1}^n \nu_j \delta_{x_j}$ with $\mu_j, \nu_j \geq 0$ and $\sum_{j=1}^n \mu_j = \sum_{j=1}^n \nu_j = 1$ that

$$\text{KL}(\mu, \nu) = \sum_{j=1}^n \log \left(\frac{\mu_j}{\nu_j} \right) \mu_j.$$

Further, the KL divergence is strictly convex with respect to the first variable. Due to the Fenchel conjugate pairing

$$H(\varphi) = \int_{\mathbb{X}} \exp(\varphi) - 1 \, d\nu \quad \text{and} \quad H^*(\mu) = \text{KL}(\mu, \nu), \quad (7)$$

the derivative relation (5) simplifies to

$$\varphi \in \partial_{\mu} \text{KL}(\mu, \nu) \quad \Leftrightarrow \quad \mu = e^{\varphi} \nu \quad \Leftrightarrow \quad \varphi = \log \left(\frac{d\mu}{d\nu} \right). \quad (8)$$

Finally, note that the KL divergence and the total variation norm $\|\cdot\|_{\mathcal{M}}$ are related by the *Pinsker inequality* $\|\mu - \nu\|_{\mathcal{M}}^2 \leq \text{KL}(\mu, \nu)$.

Discrepancies

In this section, we introduce the notation of discrepancies and have a closer look at (conditionally) positive definite kernels. In particular, we emphasize how conditionally positive definite kernels can be modified to positive definite ones.

Let $\sigma_{\mathbb{X}} \in \mathcal{M}(\mathbb{X})$ be nonnegative with $\text{supp}(\sigma_{\mathbb{X}}) = \mathbb{X}$. The given definition of discrepancies is based on symmetric, positive definite, continuous kernels. There is a close relation to general discrepancies related to measures on $\mathcal{B}(\mathbb{X})$; see Novak and Wozniakowski (2010). Recall that a symmetric function $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ is *positive definite* if for any finite number $n \in \mathbb{N}$ of points $x_j \in \mathbb{X}$, $j = 1, \dots, n$, the relation

$$\sum_{i,j=1}^n a_i a_j K(x_i, x_j) \geq 0$$

is satisfied for all $(a_j)_{j=1}^n \in \mathbb{R}^n$ and *strictly positive definite* if strict inequality holds for all $(a_j)_{j=1}^n \neq 0$. Assuming that $K \in C(\mathbb{X} \times \mathbb{X})$ is symmetric, positive definite, we know by Mercer's theorem (Cucker and Smale 2002; Mercer 1909; Steinwart and Scovel 2011) that there exists an orthonormal basis $\{\phi_k : k \in \mathbb{N}\}$ of $L^2(\mathbb{X}, \sigma_{\mathbb{X}})$ and nonnegative coefficients $\{\alpha_k\}_{k \in \mathbb{N}} \in \ell_1$ such that K has the Fourier expansion

$$K(x, y) = \sum_{k=0}^{\infty} \alpha_k \phi_k(x) \overline{\phi_k(y)} \quad (9)$$

with absolute and uniform convergence of the right-hand side. If $\alpha_k > 0$ for some $k \in \mathbb{N}_0$, the corresponding function ϕ_k is continuous. Every function $f \in L^2(\mathbb{X}, \sigma_{\mathbb{X}})$ has a Fourier expansion

$$f = \sum_{k=0}^{\infty} \hat{f}_k \phi_k, \quad \hat{f}_k := \int_{\mathbb{X}} f \overline{\phi_k} \, d\sigma_{\mathbb{X}}.$$

Moreover, for $k \in \mathbb{N}_0$ with $\alpha_k > 0$, the *Fourier coefficients* of $\mu \in \mathcal{P}(\mathbb{X})$ are well-defined by

$$\hat{\mu}_k := \int_{\mathbb{X}} \overline{\phi_k} \, d\mu.$$

The kernel K gives rise to a *reproducing kernel Hilbert space* (RKHS). More precisely, the function space

$$H_K(\mathbb{X}) := \left\{ f \in L^2(\mathbb{X}, \sigma_{\mathbb{X}}) : \sum_{k=0}^{\infty} \alpha_k^{-1} |\hat{f}_k|^2 < \infty \right\}$$

equipped with the inner product and the corresponding norm

$$\langle f, g \rangle_{H_K(\mathbb{X})} = \sum_{k=0}^{\infty} \alpha_k^{-1} \hat{f}_k \overline{\hat{g}_k}, \quad \|f\|_{H_K(\mathbb{X})} = \langle f, f \rangle_{H_K(\mathbb{X})}^{\frac{1}{2}} \tag{10}$$

forms a Hilbert space with reproducing kernel, i.e.,

$$\begin{aligned} K(x, \cdot) &\in H_K(\mathbb{X}) && \text{for all } x \in \mathbb{X}, \\ f(x) &= \langle f, K(x, \cdot) \rangle_{H_K(\mathbb{X})} && \text{for all } f \in H_K(\mathbb{X}), x \in \mathbb{X}. \end{aligned} \tag{11}$$

Note that $f \in H_K(\mathbb{X})$ implies $\hat{f}_k = 0$ if $\alpha_k = 0$, in which case we make the convention $\alpha_k^{-1} \hat{f}_k = 0$ in (10). Indeed, $H_K(\mathbb{X})$ is the closure of the linear span of $\{K(x_j, \cdot) : x_j \in \mathbb{X}\}$ with respect to the norm (10). The space $H_K(\mathbb{X})$ is continuously embedded in $C(\mathbb{X})$, and hence point evaluations in $H_K(\mathbb{X})$ are continuous. Since the series in (9) converges uniformly and the functions ϕ_k are continuous, the function

$$\|K(x, \cdot)\|_{H_K(\mathbb{X})} = \left\| \sum_{k=0}^{\infty} \alpha_k \phi_k(x) \overline{\phi_k(\cdot)} \right\|_{H_K(\mathbb{X})} = \left(\sum_{k=0}^{\infty} \alpha_k |\phi_k(x)|^2 \right)^{\frac{1}{2}}$$

is also continuous so that we have $\int_{\mathbb{X}} \|K(x, \cdot)\|_{H_K(\mathbb{X})} \, d\mu(x) < \infty$. By the definition of Bochner integrals (see Hytönen et al. (2016, Prop. 1.3.1)), we have for any $\mu \in \mathcal{P}(\mathbb{X})$ that

$$\int_{\mathbb{X}} K(x, \cdot) \, d\mu(x) \in H_K(\mathbb{X}). \tag{12}$$

For $\mu, \nu \in \mathcal{M}(\mathbb{X})$, the *discrepancy* $\mathcal{D}_K(\mu, \nu)$ is defined as norm of the linear operator $T : H_K \rightarrow \mathbb{R}$ with $\varphi \mapsto \int_{\mathbb{X}} \varphi \, d\xi$,

$$\mathcal{D}_K(\mu, \nu) = \max_{\|\varphi\|_{H_K(\mathbb{X})} \leq 1} \int_{\mathbb{X}} \varphi \, d\xi, \tag{13}$$

where $\xi := \mu - \nu$; see Gnewuch (2012) and Novak and Wozniakowski (2010). If $\mu_n \rightharpoonup \mu$ and $\nu_n \rightharpoonup \nu$ as $n \rightarrow \infty$, then also $\mu_n \otimes \nu_n \rightharpoonup \mu \otimes \nu$. Thus, continuity of K implies that $\lim_{n \rightarrow \infty} \mathcal{D}_K(\mu_n, \nu_n) = \mathcal{D}_K(\mu, \nu)$. Since

$$\int_{\mathbb{X}} \varphi \, d\xi = \int_{\mathbb{X}} \langle \varphi, K(x, \cdot) \rangle_{H_K(\mathbb{X})} \, d\xi(x) = \left\langle \varphi, \int_{\mathbb{X}} K(x, \cdot) \, d\xi(x) \right\rangle_{H_K(\mathbb{X})},$$

we obtain by Schwarz' inequality that the optimal dual potential (up to the sign) is given by

$$\hat{\varphi}_K = \frac{\int_{\mathbb{X}} K(x, \cdot) \, d\xi(x)}{\|\int_{\mathbb{X}} K(x, \cdot) \, d\xi(x)\|_{H_K(\mathbb{X})}} = \frac{\int_{\mathbb{X}} K(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) \, d\nu(x)}{\|K(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) \, d\nu(x)\|_{H_K(\mathbb{X})}}. \tag{14}$$

In the following, it is always clear from the context if the Fourier transform of the function or the optimal dual potential is meant. Further, Riesz' representation theorem implies

$$\mathcal{D}_K(\mu, \nu) = \max_{\|\varphi\|_{H_K(\mathbb{X})} \leq 1} \int_{\mathbb{X}} \varphi \, d\xi = \left\| \int_{\mathbb{X}} K(x, \cdot) \, d\xi(x) \right\|_{H_K(\mathbb{X})},$$

so that we conclude by Fubini's theorem and (11) that

$$\begin{aligned} \mathcal{D}_K^2(\mu, \nu) &= \left\| \int_{\mathbb{X}} K(x, \cdot) \, d\xi(x) \right\|_{H_K(\mathbb{X})}^2 = \int_{\mathbb{X}^2} K \, d(\xi \otimes \xi) \\ &= \int_{\mathbb{X}^2} K \, d(\mu \otimes \mu) + \int_{\mathbb{X}^2} K \, d(\nu \otimes \nu) - 2 \int_{\mathbb{X}^2} K \, d(\mu \otimes \nu). \end{aligned} \tag{15}$$

By (9), we finally get

$$\mathcal{D}_K^2(\mu, \nu) = \sum_{k=0}^{\infty} \alpha_k |\hat{\mu}_k - \hat{\nu}_k|^2, \tag{16}$$

where the summation runs over all $k \in \mathbb{N}_0$ with $\alpha_k > 0$.

Remark 1 (Relation to attraction-repulsion functionals). We briefly consider the relation to attraction-repulsion functionals motivated from electrostatic halftoning; see Schmaltz et al. (2010) and Teuber et al. (2011). Let $\nu = w \, dx$ be fixed, for example, a continuous (normalized) image with gray values in $[0, 1]$ represented by

$w: \mathbb{X} \rightarrow [0, 1]$, where pure black is the largest value of w and white the smallest one. Then, looking for a discrete measure $\mu = \frac{1}{M} \sum_{j=1}^M \delta(\cdot - p_j)$ that approximates ν by minimizing the squared discrepancy is equivalent to solving the minimization problem

$$\arg \min_{p \in \mathbb{R}^M} \left\{ \underbrace{\frac{1}{2M} \sum_{i,j=1}^M K(p_i, p_j)}_{\text{repulsion}} - \underbrace{\sum_{i=1}^M \int_{\mathbb{X}} w(x) K(x, p_i)}_{\text{attraction}} \right\}.$$

For $K(x, y) = h(\|x - y\|)$ and a decreasing function $h: [0, +\infty) \rightarrow \mathbb{R}$, it becomes clear that

- the first term is minimal if the points are far away from each other, implying a *repulsion*;
- the second (negative) term becomes maximal if for large $w(x)$, there are many points positioned in this area; so it can be considered as an *attraction* steered by w .

Kernels In this paragraph, we want to have a closer look at appropriate kernels. Recall that for symmetric, positive definite kernels $K_i \in C(\mathbb{X} \times \mathbb{X})$, $i = 1, 2$, and $\alpha > 0$, the kernels αK_1 , $K_1 + K_2$, $K_1 \cdot K_2$, and $\exp(K_1)$ are again positive definite; see Steinwart and Christmann (2008, Lems. 4.5 and 4.6).

Of special interest are so-called radial kernels of the form

$$K(x, y) := h(\text{dist}_{\mathbb{X}}(x, y)),$$

where $h: [0, +\infty) \rightarrow \mathbb{R}$. In the following, the discussion is restricted to compact sets \mathbb{X} in \mathbb{R}^d and the Euclidean distance $\text{dist}_{\mathbb{X}}(x, y) = \|x - y\|$. Many results on positive definite functions on \mathbb{R}^d go back to Schoenberg (1938) and Micchelli (1986). For a good overview, we refer to Wendland (2004), where some of the following statements can be found. Clearly, restricting positive definite kernels on \mathbb{R}^d to compact subsets \mathbb{X} results in positive definite kernels on \mathbb{X} . The radial kernels related to the Gaussian, which are quite popular in MMDs, and the inverse multiquadric given by

$$h(r) = e^{-r^2/c^2} \quad \text{and} \quad h(r) = (c^2 + r^2)^{-p}, \quad c, p > 0,$$

are known to be strictly positive definite on \mathbb{R}^d for every $d \in \mathbb{N}$. Further, the following compactly supported functions h give rise to positive definite kernels in \mathbb{R}^d

$$h(r) = (1 - r)_+^p, \quad p \geq \left\lfloor \frac{d}{2} \right\rfloor + 1, \tag{17}$$

where $\lfloor a \rfloor$ denotes the largest integer less or equal than $a \in \mathbb{R}$ and $a_+ := \max(a, 0)$.

In connection with Wasserstein distances, we are interested in (negative) powers of distances $K(x, y) = \|x - y\|^p$, $p > 0$, related to the functions $h(r) = r^p$. Unfortunately, all these functions are not positive definite! By (17), we know that $\tilde{K}(x, y) = 1 - |x - y|$ is positive definite in one dimension $d = 1$. A more general result for the Euclidean distance is given in the following proposition:

Proposition 1. *Let $K(x, y) = -\|x - y\|$. For every compact set $\mathbb{X} \subset \mathbb{R}^d$, there exists a constant $C > 0$ such that the function*

$$\tilde{K}(x, y) := C - \|x - y\|$$

is positive definite on \mathbb{X} . Further, for $\mu, \nu \in \mathcal{P}(\mathbb{X})$, it holds

$$\mathcal{D}_{\tilde{K}}^2(\mu, \nu) = \mathcal{D}_K^2(\mu, \nu) \quad \text{and} \quad \hat{\varphi}_{\tilde{K}} = \hat{\varphi}_K.$$

Proof. In Gräf (2013, Cor. 2.15), it was shown that \tilde{K} is positive definite. The rest follows in a straightforward way from (15) and (14) regarding that μ and ν are probability measures.

Some interesting functions such as negative powers of Euclidean distances or the smoothed distance function $\sqrt{c^2 + \|x - y\|^2}$, $0 < c \ll 1$, are conditionally positive definite. Let $\Pi_{m-1}(\mathbb{R}^d)$ denote the $\binom{d+m-1}{d}$ -dimensional space of polynomials on \mathbb{R}^d of absolute degree (sum of exponents) $\leq m - 1$. A function $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ is *conditionally positive definite of order m* if for all points $x_1, \dots, x_n \in \mathbb{R}^d$, $n \in \mathbb{N}$, the relation

$$\sum_{i,j=1}^n a_i a_j K(x_i, x_j) \geq 0 \tag{18}$$

holds true for all $a_1, \dots, a_n \in \mathbb{R}$ satisfying

$$\sum_{i=1}^n a_i P(x_i) = 0 \quad \text{for all} \quad P \in \Pi_{m-1}(\mathbb{R}^d).$$

If strong inequality holds in (18) except for $a_i = 0$ for all $i = 1, \dots, n$, then K is called *strictly conditionally positive definite of order m* . In particular, for $m = 1$, the condition (18) relaxes to $\sum_{i=1}^n a_i = 0$.

The radial kernels related to the following functions are strictly conditionally positive definite of order m on \mathbb{R}^d :

$$\begin{aligned} h(r) &= (-1)^{\lceil p \rceil} (c^2 + r^2)^p, & p > 0, p \notin \mathbb{N}, m = \lceil p \rceil, \\ h(r) &= (-1)^{\lceil p/2 \rceil} r^p, & p > 0, p \notin 2\mathbb{N}, m = \lceil p/2 \rceil, \\ h(r) &= (-1)^{k+1} r^{2k} \log(r), & k \in \mathbb{N}, m = k + 1, \end{aligned}$$

where $\lceil a \rceil$ denotes the smallest integer larger or equal than $a \in \mathbb{R}$. The first group of functions is called multiquadric and the last group is known as thin plate splines. In connection with Wasserstein distances, the second group of functions is of interest.

By the following lemma, it is easy to turn conditionally positive definite functions into positive definite ones. However, only for conditionally positive definite functions of order $m = 1$ that the discrepancy remains the same.

Lemma 1. *Let $\Xi := \{u_k : k = 1, \dots, N\}$ with $N := \binom{d+m-1}{m-1}$ be a set of points such that $P(u_k) = 0$ for all $k = 1, \dots, N$, $P \in \Pi_{m-1}(\mathbb{R}^d)$, is only fulfilled for the zero polynomial. Denote by $\{P_k : k = 1, \dots, N\}$ the set of Lagrangian basis polynomials with respect to Ξ , i.e., $P_k(u_j) = \delta_{jk}$. Let $K \in C(\mathbb{X} \times \mathbb{X})$ be a symmetric conditionally positive definite kernel of order m .*

(i) *Then*

$$\begin{aligned} \tilde{K}(x, y) &:= K(x, y) - \sum_{j=1}^N P_j(x)K(u_j, y) - \sum_{k=1}^N P_k(y)K(x, u_k) \\ &\quad + \sum_{j,k=1}^N P_j(x)P_k(y)K(u_j, u_k) \end{aligned}$$

is a positive definite kernel.

(ii) *If μ and ν have the same moments up to order $m - 1$, then they satisfy*

$$\mathcal{D}_{\tilde{K}}^2(\mu, \nu) = \mathcal{D}_K^2(\mu, \nu).$$

(iii) *In particular, we have for $m = 1$, $\mu, \nu \in \mathcal{P}(\mathbb{X})$ and any fixed $u \in \mathbb{X}$ that*

$$\tilde{K}(x, y) = K(x, y) - K(u, y) - K(x, u) + K(u, u) \tag{19}$$

and

$$\begin{aligned} \mathcal{D}_{\tilde{K}}^2(\mu, \nu) &= \mathcal{D}_K^2(\mu, \nu), \\ \hat{\varphi}_{\tilde{K}} &= \frac{\int_{\mathbb{X}} K(x, \cdot) d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) d\nu(x) + c_\nu - c_\mu}{\|\int_{\mathbb{X}} K(x, \cdot) d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) d\nu(x) + c_\nu - c_\mu\|_{H_K(\mathbb{X})}}, \end{aligned}$$

where

$$c_\mu := \int_{\mathbb{X}} K(x, u) \, d\mu(x) \quad \text{and} \quad c_\nu := \int_{\mathbb{X}} K(x, u) \, d\nu(x). \tag{20}$$

Proof.

- (i) This part follows a straightforward computation; see Wendland (2004, Thm. 10.18).
- (ii) Assuming that μ and ν have the same moments up to order $m - 1$, i.e.,

$$p_j = \int_{\mathbb{X}} P_j(x) \, d\mu(x) = \int_{\mathbb{X}} P_j(x) \, d\nu(x), \quad j = 1, \dots, N,$$

and abbreviating for the symmetric kernels

$$c_{\mu,j} := \int_{\mathbb{X}} K(u_j, y) \, d\mu(y), \quad c_{\nu,j} := \int_{\mathbb{X}} K(u_j, y) \, d\nu(y),$$

we obtain by definition of \tilde{K} that

$$\begin{aligned} & \mathcal{D}_{\tilde{K}}^2(\mu, \nu) \\ &= \int_{\mathbb{X}^2} \tilde{K} \, d(\mu \otimes \mu) + \int_{\mathbb{X}^2} \tilde{K} \, d(\nu \otimes \nu) - 2 \int_{\mathbb{X}^2} \tilde{K} \, d(\mu \otimes \nu) \\ &= \mathcal{D}_{\tilde{K}}^2(\mu, \nu) - \sum_{j=1}^N p_j(c_{\mu,j} + c_{\nu,j}) - \sum_{k=1}^N p_j(c_{\mu,k} + c_{\nu,k}) + 2 \sum_{j,k=1}^N p_j p_k K(u_j, u_k) \\ &\quad + \sum_{j=1}^N p_j(c_{\mu,j} + c_{\nu,j}) + \sum_{k=1}^N p_j(c_{\mu,k} + c_{\nu,k}) - 2 \sum_{j,k=1}^N p_j p_k K(u_j, u_k) \\ &= \mathcal{D}_{\tilde{K}}^2(\mu, \nu). \end{aligned}$$

- (iii) Let $m = 1$. Then we have for the optimal dual potential in (14) related to $\mathcal{D}_{\tilde{K}}$ that

$$\begin{aligned} \hat{\varphi}_{\tilde{K}} &= \frac{\int_{\mathbb{X}} \tilde{K}(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} \tilde{K}(x, \cdot) \, d\nu(x)}{\| \int_{\mathbb{X}} \tilde{K}(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} \tilde{K}(x, \cdot) \, d\nu(x) \|_{H_K(\mathbb{X})}} \\ &= \frac{\int_{\mathbb{X}} K(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) \, d\nu(x) + c_\nu - c_\mu}{\| \int_{\mathbb{X}} K(x, \cdot) \, d\mu(x) - \int_{\mathbb{X}} K(x, \cdot) \, d\nu(x) + c_\nu - c_\mu \|_{H_K(\mathbb{X})}}. \end{aligned}$$

Optimal Transport and Wasserstein Distances

The following discussion about optimal transport is based on Ambrosio et al. (2005), Cuturi and Peyré (2019), and Santambrogio (2015), where many aspects simplify due to the compactness of \mathbb{X} and the assumption that the cost c is Lipschitz continuous. Let $\mu, \nu \in \mathcal{P}(\mathbb{X})$ and $c \in C(\mathbb{X} \times \mathbb{X})$ be a nonnegative, symmetric, and Lipschitz continuous function. Then, the *Kantorovich problem of optimal transport* (OT) reads

$$\text{OT}(\mu, \nu) := \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{X}^2} c \, d\pi, \tag{21}$$

where $\Pi(\mu, \nu)$ denotes the set of joint probability measures π on \mathbb{X}^2 with marginals μ and ν . In our setting, the OT functional $\pi \mapsto \int_{\mathbb{X}^2} c \, d\pi$ is weakly continuous, (21) has a solution, and every such minimizer $\hat{\pi}$ is called optimal transport plan. In general, we cannot expect the optimal transport plan to be unique. However, if \mathbb{X} is a compact subset of a separable Hilbert space, $c(x, y) = \|x - y\|_{\mathbb{X}}^p$, $p \in (1, \infty)$, and either μ or ν is *regular* (see Ambrosio et al. (2005, Def. 6.2.2) for the technical definition), then (21) has a unique solution. Instead of giving the exact definition, we want to remark that for $\mathbb{X} = \mathbb{R}^d$ the regular measures are precisely the ones which have a density with respect to the Lebesgue measure.

The c -transform $\varphi^c \in C(\mathbb{X})$ of $\varphi \in C(\mathbb{X})$ is defined as

$$\varphi^c(y) = \min_{x \in \mathbb{X}} \{c(x, y) - \varphi(x)\}.$$

Note that φ^c has the same Lipschitz constant as c . A function $\varphi^c \in C(\mathbb{X})$ is called c -concave if it is the c -transform of some function $\varphi \in C(\mathbb{X})$.

The dual formulation of the OT problem (21) reads

$$\text{OT}(\mu, \nu) = \max_{\substack{(\varphi, \psi) \in C(\mathbb{X})^2 \\ \varphi(x) + \psi(y) \leq c(x, y)}} \int_{\mathbb{X}} \varphi \, d\mu + \int_{\mathbb{X}} \psi \, d\nu. \tag{22}$$

Maximizing pairs are essentially of the form $(\varphi, \psi) = (\hat{\varphi}, \hat{\varphi}^c)$ for some c -concave function $\hat{\varphi}$ and fulfill $\hat{\varphi}(x) + \hat{\varphi}^c(y) = c(x, y)$ in $\text{supp}(\hat{\pi})$, where $\hat{\pi}$ is any optimal transport plan. The function $\hat{\varphi}$ is called (Kantorovich) potential for the couple (μ, ν) . If $(\hat{\varphi}, \hat{\psi})$ is an optimal pair, clearly also $(\hat{\varphi} - C, \hat{\psi} + C)$ with $C \in \mathbb{R}$ is optimal, and manipulations outside of $\text{supp}(\mu)$ and $\text{supp}(\nu)$ do not change the functional value. But even if we exclude such manipulations, the optimal dual potentials are in general not unique as Example 1 shows.

Example 1. We choose $\mathbb{X} = [0, 1]$, $c(x, y) = |x - y|$, $\mu = \delta_{0.2} + \delta_{1.2}$, and $\nu = \delta_{0.1} + \delta_{0.9}$. Then, $\text{OT}(\mu, \nu) = 0.1$ with the unique optimal transport plan $\hat{\pi} = \frac{1}{2}\delta_{0.1} + \frac{1}{2}\delta_{1.0.9}$. Optimal dual potentials are given by

$$\hat{\varphi}_1(x) = \begin{cases} 0.1 - x & \text{for } x \in [0, 0.1], \\ x - 0.9 & \text{for } x \in [0.9, 1], \\ 0 & \text{else,} \end{cases} \text{ and } \hat{\varphi}_2(x) = \begin{cases} 0.2 - x & \text{for } x \in [0, 0.2], \\ x - 0.9 & \text{for } x \in [0.9, 1], \\ 0 & \text{else.} \end{cases}$$

Clearly, these potentials do not differ only by a constant.

Remark 2. Note that the space $C(\mathbb{X})^2$ in the dual problem could also be replaced with $C(\text{supp}(\mu)) \times C(\text{supp}(\nu))$. Using the Tietze extension theorem, any feasible point of the restricted problem can be extended to a feasible point of the original problem, and hence the problems coincide. If the problem is restricted, all other concepts have to be adapted accordingly.

For $p \in [1, \infty)$, the p -Wasserstein distance W_p between $\mu, \nu \in \mathcal{P}(\mathbb{X})$ is defined by

$$W_p(\mu, \nu) := \left(\min_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{X}^2} \text{dist}(x, y)^p d\pi(x, y) \right)^{\frac{1}{p}}.$$

It is a metric on $\mathcal{P}(\mathbb{X})$, which metrizes the weak topology. Indeed, due to compactness of \mathbb{X} , we have that $\mu_k \rightharpoonup \mu$ if and only if $\lim_{k \rightarrow \infty} W_p(\mu_k, \mu) = 0$.

For $1 \leq p \leq q < \infty$, it holds $W_p \leq W_q$. The distance W_1 is also called *Kantorovich-Rubinstein distance* or *Earth’s mover distance*. Here, it holds $\varphi^c = -\varphi$ and the dual problem reads

$$W_1(\mu, \nu) = \max_{|\varphi|_{\text{Lip}(\mathbb{X})} \leq 1} \int_{\mathbb{X}} \varphi d\xi, \quad \xi := \mu - \nu,$$

where the maximum is taken over all Lipschitz continuous functions with Lipschitz constant bounded by 1. This looks similar to the discrepancy (13), but the space of test functions is larger for W_1 .

The distance W_1 is related to W_p by

$$W_1(\mu, \nu) \leq W_p(\mu, \nu) \leq C W_1(\mu, \nu)^{\frac{1}{p}}$$

with a constant $0 \leq C < \infty$ depending on $\text{diam}(\mathbb{X})$ and p .

Regularized Optimal Transport

In this section, we give a self-contained introduction to continuous *regularized optimal transport*. For $\mu, \nu \in \mathcal{P}(\mathbb{X})$ and $\varepsilon > 0$, regularized OT is defined as

$$\text{OT}_\varepsilon(\mu, \nu) := \min_{\pi \in \Pi(\mu, \nu)} \left\{ \int_{\mathbb{X}^2} c \, d\pi + \varepsilon \text{KL}(\pi, \mu \otimes \nu) \right\}. \tag{23}$$

Compared to the original OT problem, we will see in the numerical part that OT_ε can be efficiently solved numerically; see also Cuturi and Peyré (2019). Moreover, OT_ε has the following properties:

Lemma 2.

- (i) *There is a unique minimizer $\hat{\pi}_\varepsilon \in \mathcal{P}(\mathbb{X}^2)$ of (23) with finite value.*
- (ii) *The function OT_ε is weakly continuous and Fréchet differentiable.*
- (iii) *For any $\mu, \nu \in \mathcal{P}(\mathbb{X})$ and $\varepsilon_1, \varepsilon_2 \in [0, \infty]$ with $\varepsilon_1 \leq \varepsilon_2$, it holds*

$$\text{OT}_{\varepsilon_1}(\mu, \nu) \leq \text{OT}_{\varepsilon_2}(\mu, \nu).$$

Proof.

- (i) First, note that $\mu \otimes \nu$ is a feasible point and hence the infimum is finite. Existence of minimizers follows as the functional is weakly lsc and $\Pi(\mu, \nu) \subset \mathcal{P}(\mathbb{X}^2)$ is weakly compact. Uniqueness follows since $\text{KL}(\cdot, \mu \otimes \nu)$ is strictly convex.
- (ii) The proof uses the dual formulation in Proposition 3; see Feydy et al. (2019, Prop. 2).
- (iii) Let $\hat{\pi}_{\varepsilon_2}$ be the minimizer for $\text{OT}_{\varepsilon_2}(\mu, \nu)$. Then, it holds

$$\begin{aligned} \text{OT}_{\varepsilon_2}(\mu, \nu) &= \int_{\mathbb{X}^2} c \, d\hat{\pi}_{\varepsilon_2} + \varepsilon_2 \text{KL}(\hat{\pi}_{\varepsilon_2}, \mu \otimes \nu) \\ &\geq \int_{\mathbb{X}^2} c \, d\hat{\pi}_{\varepsilon_2} + \varepsilon_1 \text{KL}(\hat{\pi}_{\varepsilon_2}, \mu \otimes \nu) \geq \text{OT}_{\varepsilon_1}(\mu, \nu). \end{aligned}$$

Note that in special cases, e.g., for absolutely continuous measures (see Carlier et al. (2017) and Léonard (2012)), it is possible to show convergence of the optimal solutions $\hat{\pi}_\varepsilon$ to an optimal solution of $\text{OT}(\mu, \nu)$ as $\varepsilon \rightarrow 0$. However, we are not aware of a fully general result. An extension of entropy regularization to unbalanced OT is discussed in Chizat et al. (2018).

Originally, entropic regularization was proposed in Cuturi (2013) for *discrete* probability measures with the negative entropy E (see also Peyré (2015)),

$$\begin{aligned} \tilde{\text{OT}}_\varepsilon(\mu, \nu) &:= \min_{\pi \in \Pi(\mu, \nu)} \left\{ \int_{\mathbb{X}^2} c \, d\pi + \varepsilon E(\pi) \right\}, \\ E(\pi) &:= \sum_{i,j=1}^n \log(p_{ij}) p_{ij} = \text{KL}(\pi, \lambda \otimes \lambda), \end{aligned}$$

where λ denotes the counting measure. For $\pi \in \Pi(\mu, \nu)$, it is easy to check that

$$\begin{aligned} E(\pi) &= \text{KL}(\pi, \mu \otimes \nu) + \sum_{i,j=1}^n \log(\mu_i \nu_j) \mu_i \nu_j \\ &= \text{KL}(\pi, \mu \otimes \nu) + \text{KL}(\mu \otimes \nu, \lambda \otimes \lambda), \end{aligned}$$

i.e., the minimizers are independent of the chosen regularization. For non-discrete measures, special care is necessary as the following remark shows:

Remark 3 (KL($\pi, \mu \otimes \nu$) versus $E(\pi)$ regularization). Since the entropy is only defined for measures with densities, we consider compact sets $\mathbb{X} \subset \mathbb{R}^d$ equipped with the normalized Lebesgue measure λ and $\mu, \nu \ll \lambda$ with densities $\sigma_\mu, \sigma_\nu \in L^1(\mathbb{X})$. For $\pi \ll \lambda \otimes \lambda$ with density σ_π , the entropy is defined by

$$E(\pi) = \int_{\mathbb{X}^2} \log(\sigma_\pi) \sigma_\pi \, d(\lambda \otimes \lambda) = \text{KL}(\pi, \lambda \otimes \lambda).$$

Note that for any $\pi \in \Pi(\mu, \nu)$, we have

$$\pi \ll \mu \otimes \nu \iff \pi \ll \lambda \otimes \lambda,$$

where the right implication follows directly and the left one can be seen as follows: If $\pi \ll \lambda \otimes \lambda$ with density $\sigma_\pi \in L^1(\mathbb{X} \times \mathbb{X})$, then

$$0 = \int_{\{z \in \mathbb{X} : \sigma_\mu(z) = 0\}} \int_{\mathbb{X}} \sigma_\pi(x, y) \, dy \, dx.$$

Consequently, we get $\sigma_\pi(x, y) = 0$ a.e. on $\{z \in \mathbb{X} : \sigma_\mu(z) = 0\} \times \mathbb{X}$ (for any representative of σ_μ). The same reasoning is applicable to $\mathbb{X} \times \{z \in \mathbb{X} : \sigma_\nu(z) = 0\}$. Thus,

$$\pi = \sigma_\pi (\lambda \otimes \lambda) = \frac{\sigma_\pi(x, y)}{\sigma_\mu(x)\sigma_\nu(y)} (\mu \otimes \nu),$$

where the quotient is defined as zero if σ_μ or σ_ν vanish. Hence, the left implication also holds true.

If $\text{KL}(\mu \otimes \nu, \lambda \otimes \lambda) < \infty$, we conclude for any $\pi \ll \lambda \otimes \lambda$ with $\pi \in \Pi(\mu, \nu)$ that the following expressions are well-defined

$$\begin{aligned} &\text{KL}(\pi, \lambda \otimes \lambda) - \text{KL}(\mu \otimes \nu, \lambda \otimes \lambda) \\ &= \int_{\mathbb{X}^2} \log(\sigma_\pi) \, d\pi - \int_{\mathbb{X}^2} \log\left(\frac{d(\mu \otimes \nu)}{d(\lambda \otimes \lambda)}\right) \, d(\mu \otimes \nu) \end{aligned}$$

$$\begin{aligned}
 &= \text{KL}(\pi, \mu \otimes \nu) + \int_{\mathbb{X}^2} \log(\sigma_\mu(x)\sigma_\nu(y)) \, d\pi(x, y) \\
 &\quad - \int_{\mathbb{X}^2} \log(\sigma_\mu(x)\sigma_\nu(y)) \, d\mu(x) \, d\nu(y) \\
 &= \text{KL}(\pi, \mu \otimes \nu).
 \end{aligned}$$

Consequently, in this case we also have $\widetilde{\text{OT}}_\varepsilon(\mu, \nu) = \text{OT}_\varepsilon(\mu, \nu) + \varepsilon \text{KL}(\mu \otimes \nu, \lambda \otimes \lambda)$. The crux is the condition $\text{KL}(\mu \otimes \nu, \lambda \otimes \lambda) < \infty$, which is equivalent to μ, ν having finite entropy, i.e., σ_μ, σ_ν are in a so-called Orlicz space $L \log L$ (Navrotskaya and Rabier 2013). The authors in Clason et al. (2019) considered the entropy as regularization (with continuous cost function) and pointed out that $\widetilde{\text{OT}}_\varepsilon(\mu, \nu)$ admits a (finite) minimizer exactly in this case. However, we have seen that we can avoid this existence trouble if we regularize with $\text{KL}(\pi, \mu \otimes \nu)$ instead, which therefore seems to be a more natural choice. A comparison of the settings and a more general existence discussion based on merely continuous cost functions can be also found in Di Marino and Gerolin (2019).

Another possibility is to use quadratic regularization instead; see Lorenz et al. (2021) for more details. In connection with discrepancies, we are especially interested in the limiting case $\varepsilon \rightarrow \infty$. The next proposition is basically known; see Cuturi and Peyré (2019) and Feydy et al. (2019). However, we have not found it in this generality in the literature.

Proposition 2.

(i) *It holds $\lim_{\varepsilon \rightarrow \infty} \text{OT}_\varepsilon(\mu, \nu) = \text{OT}_\infty(\mu, \nu)$, where*

$$\text{OT}_\infty(\mu, \nu) := \int_{\mathbb{X}^2} c \, d(\mu \otimes \nu).$$

(ii) *It holds $\lim_{\varepsilon \rightarrow 0} \text{OT}_\varepsilon(\mu, \nu) = \text{OT}(\mu, \nu)$.*

Proof.

(i) For $\pi = \mu \otimes \nu$, we have

$$\int_{\mathbb{X}^2} c \, d\pi + \varepsilon \text{KL}(\pi, \mu \otimes \nu) = \text{OT}_\varepsilon(\mu, \nu)$$

and consequently $\limsup_{\varepsilon \rightarrow \infty} \text{OT}_\varepsilon(\mu, \nu) \leq \text{OT}_\infty(\mu, \nu)$. In particular, the optimal transport plan $\hat{\pi}_\varepsilon$ satisfies $\limsup_{\varepsilon \rightarrow \infty} \varepsilon \text{KL}(\hat{\pi}_\varepsilon, \mu \otimes \nu) \leq \text{OT}_\infty(\mu, \nu)$. Since KL is weakly lsc, we conclude that the sequence of minimizers $\hat{\pi}_\varepsilon$ satisfies $\hat{\pi}_\varepsilon \rightharpoonup \mu \otimes \nu$ as $\varepsilon \rightarrow \infty$. Hence, we obtain the desired result from

$$\begin{aligned} \liminf_{\varepsilon \rightarrow \infty} \text{OT}_\varepsilon(\mu, \nu) &= \liminf_{\varepsilon \rightarrow \infty} \int_{\mathbb{X}^2} c \, d\hat{\pi}_\varepsilon + \varepsilon \text{KL}(\hat{\pi}_\varepsilon, \mu \otimes \nu) \\ &\geq \liminf_{\varepsilon \rightarrow \infty} \int_{\mathbb{X}^2} c \, d\hat{\pi}_\varepsilon = \text{OT}_\infty(\mu, \nu). \end{aligned}$$

(ii) This part is more involved and follows from Proposition 5 (ii).

Similar as OT in (22), its regularized version OT_ε can be written in dual form; see Chizat et al. (2018) and Clason et al. (2019).

Proposition 3. *The (pre-)dual problem of OT_ε is given by*

$$\begin{aligned} \text{OT}_\varepsilon(\mu, \nu) = \sup_{(\varphi, \psi) \in C(\mathbb{X})^2} &\left\{ \int_{\mathbb{X}} \varphi \, d\mu + \int_{\mathbb{X}} \psi \, d\nu \right. \\ &\left. - \varepsilon \int_{\mathbb{X}^2} \exp\left(\frac{\varphi(x) + \psi(y) - c(x, y)}{\varepsilon}\right) - 1 \, d(\mu \otimes \nu) \right\}. \end{aligned} \tag{24}$$

If optimal dual solutions $\hat{\varphi}_\varepsilon$ and $\hat{\psi}_\varepsilon$ exist, they are related to the optimal transport plan $\hat{\pi}_\varepsilon$ by

$$\hat{\pi}_\varepsilon = \exp\left(\frac{\hat{\varphi}_\varepsilon(x) + \hat{\psi}_\varepsilon(y) - c(x, y)}{\varepsilon}\right) \mu \otimes \nu. \tag{25}$$

Proof. Let us consider $F \in \Gamma_0(C(\mathbb{X})^2)$, $G \in \Gamma_0(C(\mathbb{X}^2))$ with Fenchel conjugates $F^* \in \Gamma_0(\mathcal{M}(\mathbb{X})^2)$, $G^* \in \Gamma_0(\mathcal{M}(\mathbb{X}^2))$ together with a linear bounded operator $A : C(\mathbb{X})^2 \rightarrow C(\mathbb{X}^2)$ with adjoint operator $A^* : \mathcal{M}(\mathbb{X}^2) \rightarrow \mathcal{M}(\mathbb{X})^2$ defined by

$$\begin{aligned} F(\varphi, \psi) &= \int_{\mathbb{X}} \varphi \, d\mu + \int_{\mathbb{X}} \psi \, d\nu, \\ G(\varphi) &= \varepsilon \int_{\mathbb{X}^2} \exp\left(\frac{\varphi - c}{\varepsilon}\right) - 1 \, d(\mu \otimes \nu), \\ A(\varphi, \psi)(x, y) &= \varphi(x) + \psi(y). \end{aligned}$$

Then, (24) has the form of the left-hand side in (2). Incorporating (7), we get

$$G^*(\pi) = \int_{\mathbb{X}} c \, d\pi + \varepsilon \text{KL}(\pi, \mu \otimes \nu).$$

Using the indicator function ι_C defined by $\iota_C(x) := 0$ for $x \in C$ and $\iota_C(x) := +\infty$ otherwise, we have

$$\begin{aligned}
 F^*(A^*\pi) &= \sup_{(\varphi, \psi) \in C(\mathbb{X})^2} \langle A^*\pi, (\varphi, \psi) \rangle - \int_{\mathbb{X}} \varphi \, d\mu - \int_{\mathbb{X}} \psi \, d\nu \\
 &= \sup_{(\varphi, \psi) \in C(\mathbb{X})^2} \langle \pi, \varphi(x) + \psi(y) \rangle - \int_{\mathbb{X}} \varphi \, d\mu - \int_{\mathbb{X}} \psi \, d\nu \\
 &= \iota_{\Pi(\mu, \nu)}(\pi).
 \end{aligned}$$

Now, the duality relation follows from (2).

If the optimal solution $(\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon)$ exists, we can apply (3) and (8) to obtain

$$\hat{\varphi}_\varepsilon(x) + \hat{\psi}_\varepsilon(y) = c + \log \left(\frac{d\hat{\tau}_\varepsilon}{d(\mu \otimes \nu)} \right),$$

which yields (25).

Remark 4. Using the Tietze extension theorem, we could also replace the space $C(\mathbb{X})^2$ by $C(\text{supp}(\mu)) \times C(\text{supp}(\nu))$.

Note that the last term in (24) is a smoothed version of the associated constraint $\varphi(x) + \psi(y) \leq c(x, y)$ appearing in (22). Clearly, the values of φ and ψ are only relevant on $\text{supp}(\mu)$ and $\text{supp}(\nu)$, respectively. Further, for any $\varphi, \psi \in C(\mathbb{X})$ and $C \in \mathbb{R}$, the potentials $\varphi + C, \psi - C$ realize the same value in (24).

For fixed φ or ψ , the corresponding maximizing potentials in (24) are given by

$$\hat{\psi}_{\varphi, \varepsilon} = T_{\mu, \varepsilon}(\varphi) \text{ on } \text{supp}(\nu) \quad \text{and} \quad \hat{\varphi}_{\psi, \varepsilon} = T_{\nu, \varepsilon}(\psi) \text{ on } \text{supp}(\mu),$$

respectively. Here, $T_{\mu, \varepsilon}: C(\mathbb{X}) \rightarrow C(\mathbb{X})$ is defined as

$$T_{\mu, \varepsilon}(\varphi)(x) := -\varepsilon \log \left(\int_{\mathbb{X}} \exp \left(\frac{\varphi(y) - c(x, y)}{\varepsilon} \right) d\mu(y) \right). \tag{26}$$

Therefore, any pair of optimal potentials $\hat{\varphi}_\varepsilon$ and $\hat{\psi}_\varepsilon$ must satisfy

$$\hat{\psi}_\varepsilon = T_{\mu, \varepsilon}(\hat{\varphi}_\varepsilon) \text{ on } \text{supp}(\nu), \quad \hat{\varphi}_\varepsilon = T_{\nu, \varepsilon}(\hat{\psi}_\varepsilon) \text{ on } \text{supp}(\mu).$$

For every $\varphi \in C(\mathbb{X})$ and $C \in \mathbb{R}$, it holds $T_{\mu, \varepsilon}(\varphi + C) = T_{\mu, \varepsilon}(\varphi) + C$. Hence, $T_{\mu, \varepsilon}$ can be interpreted as an operator on the quotient space $C(\mathbb{X})/\mathbb{R}$, where $f_1, f_2 \in C(\mathbb{X})$ are equivalent if they differ by a real constant. This space can be equipped with the *oscillation norm*

$$\|f\|_{\circ, \infty} := \frac{1}{2}(\max f - \min f),$$

and for $f \in C(\mathbb{X})/\mathbb{R}$, there is a representative $\bar{f} \in C(\mathbb{X})$ with $\|f\|_{\circ, \infty} = \|\bar{f}\|_\infty$. Finally, it is possible to restrict the domain of $T_{\mu, \varepsilon}$ to $C(\text{supp}(\mu))$ and

$C(\text{supp}(\mu))/\mathbb{R}$, respectively. This interpretation is useful for showing convergence of the Sinkhorn algorithm. In the next lemma, we collect a few properties of $T_{\mu,\varepsilon}$; see also Genevay et al. (2019) and Vialard (2019).

Lemma 3.

- (i) For any measure $\mu \in P(\mathbb{X})$, $\varepsilon > 0$, and $\varphi \in C(\mathbb{X})$, the function $T_{\mu,\varepsilon}(\varphi) \in C(\mathbb{X})$ has the same Lipschitz constant as c and satisfies

$$T_{\mu,\varepsilon}(\varphi)(x) \in \left[\min_{y \in \text{supp}(\mu)} c(x, y) - \varphi(y), \max_{y \in \text{supp}(\mu)} c(x, y) - \varphi(y) \right]. \quad (27)$$

- (ii) For fixed $\mu \in \mathcal{P}(\mathbb{X})$, the operator $T_{\mu,\varepsilon}: C(\text{supp}(\mu)) \rightarrow C(\mathbb{X})$ is 1-Lipschitz. Additionally, the operator $T_{\mu,\varepsilon}: C(\text{supp}(\mu))/\mathbb{R} \rightarrow C(\mathbb{X})/\mathbb{R}$ is κ -Lipschitz with $\kappa < 1$.

Proof.

- (i) For $x_1, x_2 \in \mathbb{X}$ (possibly changing the naming of the variables), we obtain

$$\begin{aligned} & |T_{\mu,\varepsilon}(\varphi)(x_1) - T_{\mu,\varepsilon}(\varphi)(x_2)| \\ &= \varepsilon \left| \log \int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_2, y)}{\varepsilon}\right) d\mu(y) - \log \int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_1, y)}{\varepsilon}\right) d\mu(y) \right| \\ &= \varepsilon \log \left(\frac{\int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_2, y)}{\varepsilon}\right) d\mu(y)}{\int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_1, y)}{\varepsilon}\right) d\mu(y)} \right). \end{aligned}$$

Incorporating the L -Lipschitz continuity of c , we get

$$\exp\left(\frac{c(x_1, y) - c(x_2, y)}{\varepsilon}\right) \leq \exp\left(\frac{|c(x_1, y) - c(x_2, y)|}{\varepsilon}\right) \leq \exp\left(\frac{L}{\varepsilon}|x_1 - x_2|\right),$$

so that

$$\begin{aligned} & \int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_2, y)}{\varepsilon}\right) d\mu(y) \\ & \leq \exp\left(\frac{L}{\varepsilon}|x_1 - x_2|\right) \int_{\mathbb{X}} \exp\left(\frac{\varphi(y) - c(x_1, y)}{\varepsilon}\right) d\mu(y). \end{aligned}$$

Thus, $T_{\mu,\varepsilon}(\varphi)$ is Lipschitz continuous

$$|T_{\mu,\varepsilon}(\varphi)(x_1) - T_{\mu,\varepsilon}(\varphi)(x_2)| \leq \varepsilon \log \left(\exp\left(\frac{L}{\varepsilon}|x_1 - x_2|\right) \right) = L|x_1 - x_2|.$$

Finally, (27) follows directly from (26) since μ is a probability measure.

(ii) For any $x \in \mathbb{X}$ and $\varphi_1, \varphi_2 \in C(\text{supp}(\mu))$, it holds

$$\begin{aligned} T_{\mu,\varepsilon}(\varphi_1)(x) - T_{\mu,\varepsilon}(\varphi_2)(x) &= \int_0^1 \frac{d}{dt} T_{\mu,\varepsilon}(\varphi_1 + t(\varphi_2 - \varphi_1))(x) dt \quad (28) \\ &= \int_0^1 \int_{\mathbb{X}} (\varphi_1(z) - \varphi_2(z)) \rho_{t,x}(z) d\mu(z) dt \end{aligned}$$

with

$$\rho_{t,x} := \frac{\exp((t\varphi_2 + (1-t)\varphi_1 - c(x, \cdot)/\varepsilon))}{\int_{\mathbb{X}} \exp((t\varphi_2(z) + (1-t)\varphi_1(z) - c(x, z)/\varepsilon)) d\mu(z)}.$$

This directly implies

$$\begin{aligned} &\|T_{\mu,\varepsilon}(\varphi_1) - T_{\mu,\varepsilon}(\varphi_2)\|_\infty \\ &\leq \sup_{x \in \text{supp}(\mu)} \int_0^1 \int_{\mathbb{X}} |\varphi_1(z) - \varphi_2(z)| \rho_{t,x}(z) d\mu(z) dt \leq \|\varphi_1 - \varphi_2\|_\infty. \end{aligned}$$

In order to show the second claim, we choose representatives φ_1 and φ_2 such that $\|\varphi_1 - \varphi_2\|_\infty = \|\varphi_1 - \varphi_2\|_{\circ,\infty}$. Given $x, y \in \mathbb{X}$, we conclude using (28) that

$$\begin{aligned} &\frac{1}{2}(T_{\mu,\varepsilon}(\varphi_1)(x) - T_{\mu,\varepsilon}(\varphi_2)(x) - T_{\mu,\varepsilon}(\varphi_1)(y) + T_{\mu,\varepsilon}(\varphi_2)(y)) \\ &= \frac{1}{2} \int_0^1 \int_{\mathbb{X}} (\varphi_1(z) - \varphi_2(z)) (\rho_{t,x}(z) - \rho_{t,y}(z)) d\mu(z) dt \\ &\leq \|\varphi_1 - \varphi_2\|_{\circ,\infty} \frac{1}{2} \int_0^1 \|\rho_{t,x} - \rho_{t,y}\|_{L^1(\mu)} dt. \quad (29) \end{aligned}$$

For all $z \in \mathbb{X}$ with $p_{t,x}(z) \geq p_{t,y}(z)$, we can estimate

$$p_{t,x}(z) - p_{t,y}(z) \leq p_{t,x}(z)(1 - \exp(-2L \text{diam}(\mathbb{X})/\varepsilon))$$

and similarly for $z \in \mathbb{X}$ with $p_{t,y}(z) \geq p_{t,x}(z)$. Hence, we obtain

$$\begin{aligned} \|\rho_{t,x} - \rho_{t,y}\|_{L^1(\mu)} &\leq \int_{\mathbb{X}} (1_{\{p_{t,x} \geq p_{t,y}\}} p_{t,x} + 1_{\{p_{t,y} > p_{t,x}\}} p_{t,y}) \\ &\quad \times (1 - \exp(-2L \text{diam}(\mathbb{X})/\varepsilon)) d\mu \\ &\leq 2(1 - \exp(-2L \text{diam}(\mathbb{X})/\varepsilon)). \end{aligned}$$

Finally, inserting this into (29) implies

$$\|T_{\mu,\varepsilon}(\varphi_1) - T_{\mu,\varepsilon}(\varphi_2)\|_{0,\infty} \leq (1 - \exp(-2L \operatorname{diam}(\mathbb{X})/\varepsilon)) \|\varphi_1 - \varphi_2\|_{0,\infty}.$$

Now, we are able to prove existence of an optimal solution $(\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon)$.

Proposition 4. *The optimal potentials $\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon \in C(\mathbb{X})$ exist and are unique on $\operatorname{supp}(\mu)$ and $\operatorname{supp}(\nu)$, respectively (up to the additive constant).*

Proof. Let $\varphi_n, \psi_n \in C(\mathbb{X})$ be maximizing sequences of (24). Using the operator $T_{\mu,\varepsilon}$, these can be replaced by

$$\tilde{\psi}_n = T_{\mu,\varepsilon}(\varphi_n) \quad \text{and} \quad \tilde{\varphi}_n = T_{\nu,\varepsilon} \circ T_{\mu,\varepsilon}(\varphi_n),$$

which are Lipschitz continuous with the same constant as c by Lemma 3 (i) and therefore uniformly equi-continuous. Next, we can choose some $x_0 \in \operatorname{supp}(\mu)$ and w.l.o.g. assume $\tilde{\psi}_n(x_0) = 0$. Due to the uniform Lipschitz continuity, the potentials $\tilde{\psi}_n$ are uniformly bounded, and by (27), the same holds true for $\tilde{\varphi}_n$. Now, the theorem of Arzelà–Ascoli implies that both sequences contain convergent subsequences. Since the functional in (24) is continuous, we can readily infer the existence of optimal potentials $\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon \in C(\mathbb{X})$. Due to the uniqueness of $\hat{\pi}_\varepsilon$, (25) implies that $\hat{\varphi}_\varepsilon|_{\operatorname{supp}(\mu)}$ and $\hat{\psi}_\varepsilon|_{\operatorname{supp}(\nu)}$ are uniquely determined up to an additive constant.

Combining the optimality condition (26) and (24), we directly obtain for any pair of optimal solutions

$$\operatorname{OT}_\varepsilon(\mu, \nu) = \int_{\mathbb{X}} \hat{\varphi}_\varepsilon \, d\mu + \int_{\mathbb{X}} \hat{\psi}_\varepsilon \, d\nu. \tag{30}$$

Adding, e.g., the additional constraint

$$\int_{\mathbb{X}} \varphi \, d\mu = \frac{1}{2} \operatorname{OT}_\infty(\mu, \nu), \tag{31}$$

the restricted optimal potentials $\hat{\varphi}_\varepsilon|_{\operatorname{supp}(\mu)}$ and $\hat{\psi}_\varepsilon|_{\operatorname{supp}(\nu)}$ are unique. The next proposition investigates the limits of the potentials as $\varepsilon \rightarrow 0$ and $\varepsilon \rightarrow \infty$.

Proposition 5.

- (i) *If (31) is satisfied, the restricted potentials $\hat{\varphi}_\varepsilon|_{\operatorname{supp}(\mu)}$ and $\hat{\psi}_\varepsilon|_{\operatorname{supp}(\nu)}$ converge uniformly for $\varepsilon \rightarrow \infty$ to*

$$\hat{\varphi}_\infty(x) = \int_{\mathbb{X}} c(x, y) \, d\nu(y) - \frac{1}{2} \operatorname{OT}_\infty(\mu, \nu),$$

$$\hat{\psi}_\infty(y) = \int_{\mathbb{X}} c(x, y) \, d\mu(x) - \frac{1}{2} \text{OT}_\infty(\mu, \nu),$$

respectively.

- (ii) For $\varepsilon \rightarrow 0$, every accumulation point of $(\hat{\varphi}_\varepsilon|_{\text{supp}(\mu)}, \hat{\psi}_\varepsilon|_{\text{supp}(\nu)})$ can be extended to an optimal dual pair for $\text{OT}(\mu, \nu)$ satisfying (31). In particular, $\lim_{\varepsilon \rightarrow 0} \text{OT}_\varepsilon(\mu, \nu) = \text{OT}(\mu, \nu)$.

Proof.

- (i) Since \mathbb{X} is bounded, the Lipschitz continuity of the potentials together with (31) implies that all $\hat{\varphi}_\varepsilon$ are uniformly bounded on $\text{supp}(\mu)$. Then, we conclude for $y \in \text{supp}(\nu)$ using l’Hôpital’s rule, dominated convergence, and (31) that

$$\begin{aligned} \lim_{\varepsilon \rightarrow \infty} \hat{\psi}_\varepsilon(y) &= \lim_{\varepsilon \rightarrow \infty} - \frac{\int_{\mathbb{X}} (\hat{\varphi}_\varepsilon(x) - c(x, y)) \exp((\hat{\varphi}_\varepsilon(x) - c(x, y))/\varepsilon) \, d\mu(x)}{\int_{\mathbb{X}} \exp((\hat{\varphi}_\varepsilon(x) - c(x, y))/\varepsilon) \, d\mu(x)} \\ &= \lim_{\varepsilon \rightarrow \infty} \int_{\mathbb{X}} c(x, y) \exp((\hat{\varphi}_\varepsilon(x) - c(x, y))/\varepsilon) \\ &\quad - \hat{\varphi}_\varepsilon(x) \exp((\hat{\varphi}_\varepsilon(x) - c(x, y))/\varepsilon) \, d\mu(x) \\ &= \int_{\mathbb{X}} c(x, y) \, d\mu(x) \\ &\quad - \lim_{\varepsilon \rightarrow \infty} \int_{\mathbb{X}} \hat{\varphi}_\varepsilon(x) \left(\exp((\hat{\varphi}_\varepsilon(x) - c(x, y))/\varepsilon) - 1 \right) + \hat{\varphi}_\varepsilon(x) \, d\mu(x) \\ &= \int_{\mathbb{X}} c(x, y) \, d\mu(x) - \frac{1}{2} \text{OT}_\infty(\mu, \nu). \end{aligned}$$

Again, a similar reasoning, incorporating (27), can be applied for $\hat{\varphi}_\varepsilon$. Finally, note that pointwise convergence of uniformly Lipschitz continuous functions on compact sets implies uniform convergence.

- (ii) By continuity of the integral, we can directly infer that (31) is satisfied for any accumulation point. Note that for any fixed $\varphi \in C(\mathbb{X})$, $x \in \mathbb{X}$, and $\varepsilon \rightarrow 0$, it holds

$$T_{\mu, \varepsilon}(\varphi)(x) \rightarrow \min_{y \in \text{supp}(\mu)} c(x, y) - \varphi(y);$$

see Feydy et al. (2019, Prop. 9), which by uniform Lipschitz continuity of $T_{\mu, \varepsilon}(\varphi)$ directly implies the convergence in $C(\mathbb{X})$. Let $\{(\hat{\varphi}_{\varepsilon_j}, \hat{\psi}_{\varepsilon_j})\}_j$ be a subsequence converging to $(\hat{\varphi}_0, \hat{\psi}_0) \in C(\text{supp}(\mu)) \times C(\text{supp}(\nu))$. Then, we have

$$\begin{aligned}\hat{\psi}_0 &= \lim_{j \rightarrow \infty} \hat{\psi}_{\varepsilon_j} = \lim_{j \rightarrow \infty} T_{\mu, \varepsilon_j}(\hat{\varphi}_{\varepsilon_j}) \\ &= \lim_{j \rightarrow \infty} \left(T_{\mu, \varepsilon_j}(\hat{\varphi}_{\varepsilon_j}) - T_{\mu, \varepsilon_j}(\hat{\varphi}_0) + T_{\mu, \varepsilon_j}(\hat{\varphi}_0) \right).\end{aligned}$$

By Lemma 3 (ii), it holds

$$\|T_{\mu, \varepsilon_j}(\hat{\varphi}_{\varepsilon_j}) - T_{\mu, \varepsilon_j}(\hat{\varphi}_0)\|_\infty \leq \|\hat{\varphi}_{\varepsilon_j} - \hat{\varphi}_0\|_\infty,$$

and we conclude

$$\hat{\psi}_0 = \lim_{j \rightarrow \infty} T_{\mu, \varepsilon_j}(\hat{\varphi}_0) = \min_{y \in \text{supp}(\mu)} c(\cdot, y) - \hat{\varphi}_0(y).$$

Similarly, we get

$$\hat{\varphi}_0 = \min_{y \in \text{supp}(v)} c(\cdot, y) - \hat{\psi}_0(y).$$

Thus, $(\hat{\varphi}_0, \hat{\psi}_0)$ can be extended to a feasible point in $C(\mathbb{X})^2$ of (22) by Remark 2.

Due to continuity of (30) and since OT_ε is monotone in ε , this implies

$$\lim_{j \rightarrow \infty} \text{OT}_{\varepsilon_j}(\mu, v) = \int_{\mathbb{X}} \hat{\varphi}_0 \, d\mu + \int_{\mathbb{X}} \hat{\psi}_0 \, dv \leq \text{OT}(\mu, v) \leq \lim_{j \rightarrow \infty} \text{OT}_{\varepsilon_j}(\mu, v).$$

Hence, the extended potentials are optimal for (22). Since the subsequence choice was arbitrary, this also shows Proposition 2 (ii).

So far we cannot show the convergence of the potentials for $\varepsilon \rightarrow 0$ for the fully general case. Essentially, our approach would require that all $T_{\mu, \varepsilon}$ are contractive with a uniform constant $\beta < 1$, which is not the case. Note that if we assume that the unregularized potentials satisfying (31) are unique, then (ii) directly implies convergence of the restricted dual potentials; see also Berman (2020, Thm. 3.3) and Cominetti and San Martín (1994). Nevertheless, we always observed convergence in our numerical examples.

Sinkhorn Divergence

The regularized functional OT_ε is biased, i.e., in general $\min_v \text{OT}_\varepsilon(v, \mu) \neq \text{OT}_\varepsilon(\mu, \mu)$. Hence, the usage as distance measure is meaningless, which motivates the introduction of the *Sinkhorn divergence*

$$S_\varepsilon(\mu, \nu) = \text{OT}_\varepsilon(\mu, \nu) - \frac{1}{2} \text{OT}_\varepsilon(\mu, \mu) - \frac{1}{2} \text{OT}_\varepsilon(\nu, \nu).$$

Indeed, it was shown that S_ε is nonnegative and biconvex and metrizes the convergence in law under mild assumptions (Feydy et al. 2019). Clearly, we have $S_0 = \text{OT}$. By (14) and Proposition 5, we obtain the following corollary:

Corollary 1. *Assume that $K \in C(\mathbb{X} \times \mathbb{X})$ is symmetric and positive definite. Set $c(x, y) := -K(x, y)$. Then, it holds $S_\infty(\mu, \nu) = \frac{1}{2} \mathcal{D}_K^2(\mu, \nu)$ and the optimal dual potential $\hat{\varphi}_K$ realizing $\mathcal{D}_K(\mu, \nu)$ is related to the uniform limits $\hat{\varphi}_\infty, \hat{\psi}_\infty$ of $\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon$ in $\text{OT}_\varepsilon(\mu, \nu)$ with constraint (31) by*

$$\hat{\varphi}_K = \frac{\hat{\varphi}_\infty - \hat{\psi}_\infty}{\|\hat{\varphi}_\infty - \hat{\psi}_\infty\|_{H_K(\mathbb{X})}}.$$

Note that (12) already implies that for the chosen c , it holds $\hat{\varphi}_\infty, \hat{\psi}_\infty \in H_K(\mathbb{X})$. By Corollary 1, we have for $c(x, y) := -K(x, y)$ that $S_\infty(\mu, \nu) = \frac{1}{2} \mathcal{D}_K^2(\mu, \nu)$ if $K \in C(\mathbb{X} \times \mathbb{X})$ is symmetric, positive definite. For the cost $c(x, y) = \|x - y\|^p$ of the classical p -Wasserstein distance, we have already seen in section “Discrepancies” that $K(x, y) = -c(x, y)$ is not positive definite. However, at least for $p = 1$ the kernel is conditionally positive definite of order 1 and can be tuned by Proposition 1 to a positive definite kernel by adding a constant, which changes the value of neither the discrepancy nor the optimal dual potential. More generally, we have the following corollary:

Corollary 2. *Let $K \in C(\mathbb{X} \times \mathbb{X})$ be symmetric, conditionally positive definite of order 1, and let \tilde{K} be the corresponding positive definite kernel in (19). Then we have for $c = -\tilde{K}$ that*

$$S_\infty(\mu, \nu) = \frac{1}{2} \mathcal{D}_K^2(\mu, \nu)$$

and for the optimal dual potentials

$$\begin{aligned} \hat{\varphi}_\infty(x) &= \int_{\mathbb{X}} -K(x, y) \, d\nu(y) + \frac{1}{2} \int_{\mathbb{X}^2} K \, d(\mu \otimes \nu) + K(x, \xi) \\ &\quad + \frac{1}{2}(c_\nu - c_\mu - K(\xi, \xi)), \\ \hat{\psi}_\infty(y) &= \int_{\mathbb{X}} -K(x, y) \, d\mu(x) + \frac{1}{2} \int_{\mathbb{X}^2} K \, d(\mu \otimes \nu) + K(\xi, y) \\ &\quad + \frac{1}{2}(c_\mu - c_\nu - K(\xi, \xi)), \end{aligned}$$

with some fixed $\xi \in \mathbb{X}$ and c_μ, c_ν defined as in (20).

Proof. By Corollary 1 and Lemma 1, we obtain

$$S_\infty(\mu, \nu) = \frac{1}{2} \mathcal{D}_{\tilde{K}}(\mu, \nu)^2 = \frac{1}{2} \mathcal{D}_K(\mu, \nu)^2.$$

The second claim follows by Proposition 5.

In the following, we want to characterize the convergence of the functional $S_\varepsilon(\cdot, \nu)$ in the limiting cases $\varepsilon \rightarrow 0$ and $\varepsilon \rightarrow \infty$ for fixed $\nu \in \mathcal{P}(\mathbb{X})$. Recall that a sequence $\{F_n\}_{n \in \mathbb{N}}$ of functionals $F_n: \mathcal{P}(\mathbb{X}) \rightarrow (-\infty, +\infty]$ is said to Γ -converge to $F: \mathcal{P}(\mathbb{X}) \rightarrow (-\infty, +\infty]$ if the following two conditions are fulfilled for every $\mu \in \mathcal{P}(\mathbb{X})$ (see Braides (2002)):

- (i) $F(\mu) \leq \liminf_{n \rightarrow \infty} F_n(\mu_n)$ whenever $\mu_n \rightarrow \mu$,
- (ii) there is a sequence $\{\mu_n\}_{n \in \mathbb{N}}$ with $\mu_n \rightarrow \mu$ and $\limsup_{n \rightarrow \infty} F_n(\mu_n) \leq F(\mu)$.

The importance of Γ -convergence relies in the fact that every cluster point of minimizers of $\{F_n\}_{n \in \mathbb{N}}$ is a minimizer of F .

Proposition 6. *It holds $S_\varepsilon(\cdot, \nu) \xrightarrow{\Gamma} S_\infty(\cdot, \nu)$ as $\varepsilon \rightarrow \infty$ and $S_\varepsilon(\cdot, \nu) \xrightarrow{\Gamma} \text{OT}(\cdot, \nu)$ as $\varepsilon \rightarrow 0$.*

Proof. In both cases, the lim sup-inequality follows from Proposition 2 by choosing for some fixed $\mu \in \mathcal{P}(\mathbb{X})$ the constant sequence $\mu_n = \mu, n \in \mathbb{N}$.

Concerning the lim inf-inequality, we first treat the case $\varepsilon \rightarrow \infty$. Let $\mu_n \rightarrow \mu$ and $\varepsilon_n \rightarrow \infty$. Since $\text{OT}_\varepsilon(\mu, \nu)$ is increasing with ε , it holds for every fixed $m \in \mathbb{N}$ that

$$\begin{aligned} \liminf_{n \rightarrow \infty} S_{\varepsilon_n}(\mu_n, \nu) &= \liminf_{n \rightarrow \infty} \left(\text{OT}_{\varepsilon_n}(\mu_n, \nu) - \frac{1}{2} \text{OT}_{\varepsilon_n}(\mu_n, \mu_n) - \frac{1}{2} \text{OT}_{\varepsilon_n}(\nu, \nu) \right) \\ &\geq \liminf_{n \rightarrow \infty} \left(\text{OT}_m(\mu_n, \nu) - \frac{1}{2} \text{OT}_\infty(\mu_n, \mu_n) \right) - \frac{1}{2} \text{OT}_\infty(\nu, \nu). \end{aligned}$$

Due to the weak continuity of OT_m and OT_∞ , we obtain

$$\liminf_{n \rightarrow \infty} S_{\varepsilon_n}(\mu_n, \nu) \geq \text{OT}_m(\mu, \nu) - \frac{1}{2} \text{OT}_\infty(\mu, \mu) - \frac{1}{2} \text{OT}_\infty(\nu, \nu).$$

Letting $m \rightarrow \infty$, Proposition 2 implies the lim inf-inequality.

Next, we consider $\varepsilon \rightarrow 0$. Let $\mu_n \rightarrow \mu$ and $\varepsilon_n \rightarrow 0$. With similar arguments as above, we obtain for any fixed $m \in \mathbb{N}$ that

$$\liminf_{n \rightarrow \infty} S_{\varepsilon_n}(\mu_n, \nu) \geq \liminf_{n \rightarrow \infty} \left(\text{OT}(\mu_n, \nu) - \frac{1}{2} \text{OT}_m(\mu_n, \mu_n) \right) - \frac{1}{2} \text{OT}_m(\nu, \nu)$$

and weak continuity of OT_m and OT implies

$$\liminf_{n \rightarrow \infty} S_{\varepsilon_n}(\mu_n, \nu) \geq \text{OT}(\nu, \mu) - \frac{1}{2} \text{OT}_m(\mu, \mu) - \frac{1}{2} \text{OT}_m(\nu, \nu).$$

Using again Proposition 2, we verify the lim inf-inequality.

Numerical Approach and Examples

In this section, we discuss the Sinkhorn algorithm for computing OT_ε based on the (pre)-dual form (24) and show some numerical examples. As pointed out in Remark 4, we can restrict the potentials and the update operator (26) to $\text{supp}(\mu)$ and $\text{supp}(\nu)$, respectively. In particular, this restriction results in a discrete problem if both input measures are atomic. For a fixed starting iterate $\psi^{(0)}$, the Sinkhorn algorithm iterates are defined as

$$\begin{aligned} \varphi^{(i+1)} &= T_{\nu, \varepsilon}(\psi^{(i)}), \\ \psi^{(i+1)} &= T_{\mu, \varepsilon}(\varphi^{(i+1)}). \end{aligned}$$

Equivalently, we could rewrite the scheme with just one potential and the following update $\psi^{(i+1)} = T_{\mu, \varepsilon} \circ T_{\nu, \varepsilon}(\psi^{(i)})$. According to Lemma 3, the operator $T_{\mu, \varepsilon} \circ T_{\nu, \varepsilon}$ is contractive, and hence the Banach fixed point theorem implies that the algorithm converges linearly. Note that it suffices to enforce the additional constraint (31) after the Sinkhorn scheme by adding an appropriately chosen constant. Then, the value of $\text{OT}_\varepsilon(\mu, \nu)$ can be computed from the optimal potentials using (30). Here, we do not want to go into more detail on implementation issues, since this is not the main scope of this chapter. The numerical examples merely serve as an illustration of the theoretical results. All computations in this section are performed using GEOMLOSS, a publicly available PyTorch implementation for regularized optimal transport. Implementation details can be found in Feydy et al. (2019) and in the corresponding GitHub repository.

Demonstration of convergence results In the following, we present a numerical toy example for illustrating the convergence results from the previous sections. First, we want to verify the interpolation behavior of $S_\varepsilon(\mu, \nu)$ between $\text{OT}(\mu, \nu)$ and $\mathcal{D}_K(\mu, \nu)$. We choose $\mathbb{X} = [0, 1]$, $c(x, y) = |x - y|$ and the probability measures μ and ν depicted in Fig. 2. The resulting energies $S_\varepsilon(\mu, \nu)$ in log-scale are plotted in the same figure.

We observe that the values converge as shown in Proposition 2 and that the change mainly happens in the interval $[10^{-2}, 10^1]$. Additionally, the numerical results indicate $S_{\varepsilon_1}(\mu, \nu) \leq S_{\varepsilon_2}(\mu, \nu)$ for $\varepsilon_1 > \varepsilon_2$, which is the opposite behavior as for OT_ε where the energies increase; see Lemma 2 (iii). So far we are not aware of any theoretical result in this direction for $S_\varepsilon(\mu, \nu)$.

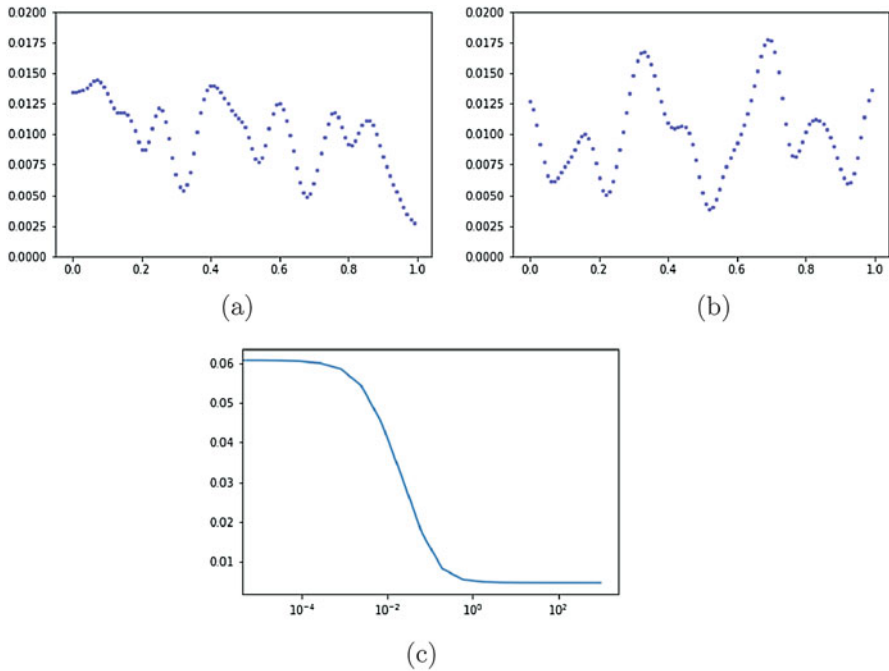


Fig. 2 Energy values between S_0 and S_∞ for two given measures on $[0, 1]$ and cost function $c(x, y) = |x - y|$. Every blue dot corresponds to the position and the weight of a Dirac measure. (a) Measure μ . (b) Measure ν . (c) Values $S_\epsilon(\mu, \nu)$ for increasing ϵ

Next, we investigate the behavior of the corresponding optimal potentials $\hat{\varphi}_\epsilon$ and $\hat{\psi}_\epsilon$ in (24). The convergence of the potentials as shown in Proposition 5 (iii) is numerically verified in Fig. 3. Further, the corresponding potentials $\hat{\varphi}_\epsilon$ are depicted in Fig. 4, and the differences $\hat{\varphi}_\epsilon - \hat{\psi}_\epsilon$ are depicted in Fig. 5. According to Corollary 1, this difference is related to the optimal potential $\hat{\varphi}_K$ in the dual formulation of the related discrepancy. The shape of the potentials ranges from something almost linear for small ϵ to something more quadratic for large ϵ . Again, we observe that the changes mainly happen for ϵ in the interval $[10^{-2}, 10^1]$ and that numerical instabilities start to occur for $\epsilon > 10^3$. For small values of ϵ , we actually observe numerical convergence and that the relation $\hat{\psi}_\epsilon \approx -\hat{\varphi}_\epsilon$ holds true; see Fig. 3c. This fits the theoretical findings for $W_1(\mu, \nu)$ in section “Optimal Transport and Wasserstein Distances”.

Dithering results Now, we want to take a short glimpse at a more involved problem. In the following, we investigate the influence of using S_ϵ with different values ϵ as approximation quality measure in dithering. For this purpose, we choose $\mathbb{X} = [-1, 1]^2$, $c(x, y) = |x - y|$, and $\mu = C \exp(-9\|x\|^2/2)(\lambda \otimes \lambda)$, where $C \in \mathbb{R}$ is a normalizing constant. In order to deal with a fully discrete problem, μ is

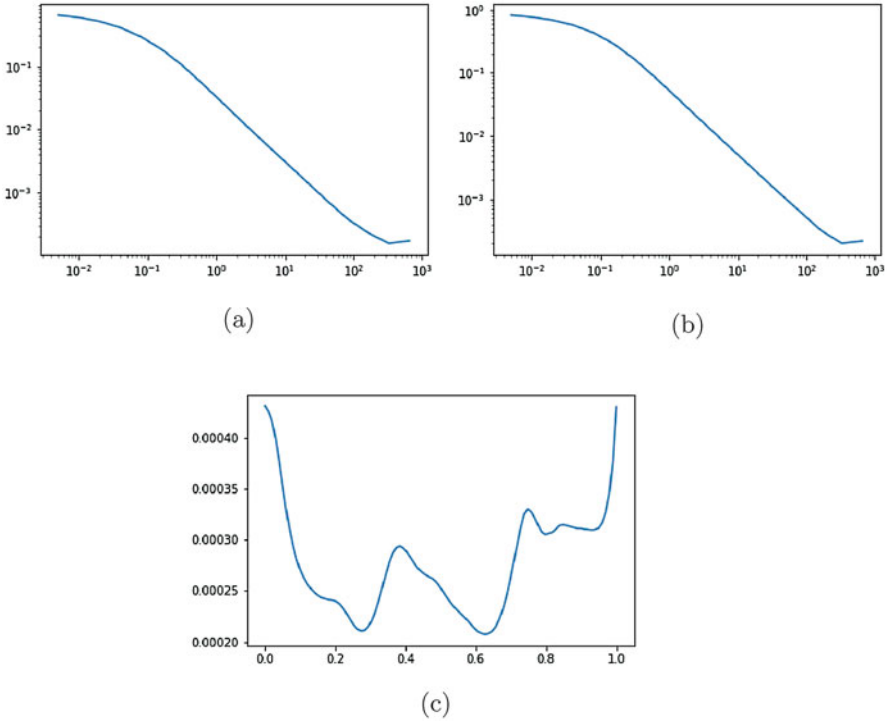


Fig. 3 Numerical verification of Prop. 5 and of $\hat{\psi}_\varepsilon \approx -\hat{\varphi}_\varepsilon$ for small ε . (a) $\sup_{\text{supp}(\mu)} |\hat{\varphi}_\varepsilon - \hat{\varphi}_\infty|$ for increasing values of ε . (b) $\sup_{\text{supp}(\nu)} |\hat{\psi}_\varepsilon - \hat{\psi}_\infty|$ for increasing values of ε . (c) $\hat{\varphi}_{1e-4} + \hat{\psi}_{1e-4}$

approximated by an atomic measure with 90×90 spikes on a regular grid. Then, we approximate μ with a measure $\nu \in \mathcal{P}_{\text{emp}}^{400}(\mathbb{X})$ (empirical measure with 400 spikes) in terms of the following objective function

$$\min_{\nu \in \mathcal{P}_{\text{emp}}^{400}(\mathbb{X})} S_\varepsilon(\mu, \nu). \tag{32}$$

For solving this problem, we can equivalently minimize over the positions of the equally weighted Dirac spikes in ν . Hence, we need the gradient of S_ε with respect to these positions. If $\varepsilon = \infty$, this gradient is given by an analytic expression. Otherwise, we can apply automatic differentiation tools to the Sinkhorn algorithm in order to compute a numerical gradient; see Feydy et al. (2019) for more details. Here, it is important to ensure high enough numerical precision and to perform enough Sinkhorn iterations. In any case, the gradient serves as input for the L-BFGS-B (quasi-Newton) method in which the Hessian is approximated in a memory-efficient way (Byrd et al. 1995). The numerical results are depicted in Fig. 6, where all examples are iterated to high numerical precision. Numerically,

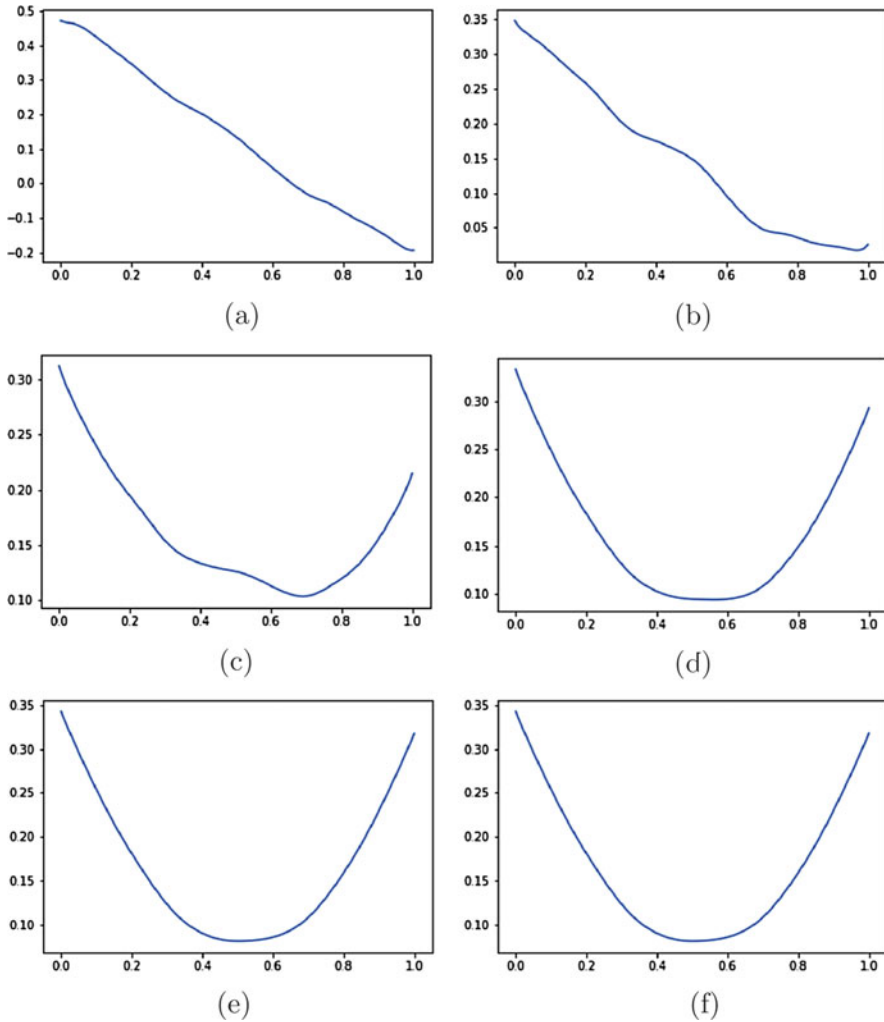


Fig. 4 Optimal potentials $\hat{\varphi}_\varepsilon$ in $\text{OT}_\varepsilon(\mu, \nu)$ for increasing values of ε . (a) $\hat{\varphi}_{0.02}$. (b) $\hat{\varphi}_{0.08}$. (c) $\hat{\varphi}_{0.32}$. (d) $\hat{\varphi}_{1.28}$. (e) $\hat{\varphi}_{81.92}$. (f) $\hat{\varphi}_\infty$

we nicely observe the convergence of $S_\varepsilon(\mu, \hat{\nu})$ in the limits $\varepsilon \rightarrow 0$ and $\varepsilon \rightarrow \infty$ as implied from the Γ -convergence result in Proposition 6. Visually, the result using Fourier methods is most appealing. Differences could be caused by the different numerical approaches. In particular, the minimization of (32) is quite challenging, and our applied approach is pretty straight forward without including any special knowledge about the problem. Noteworthy, the Fourier method uses a truncation of $S_\infty = \frac{1}{2} \mathcal{D}_K^2$ in the Fourier domain (see (16)), namely,

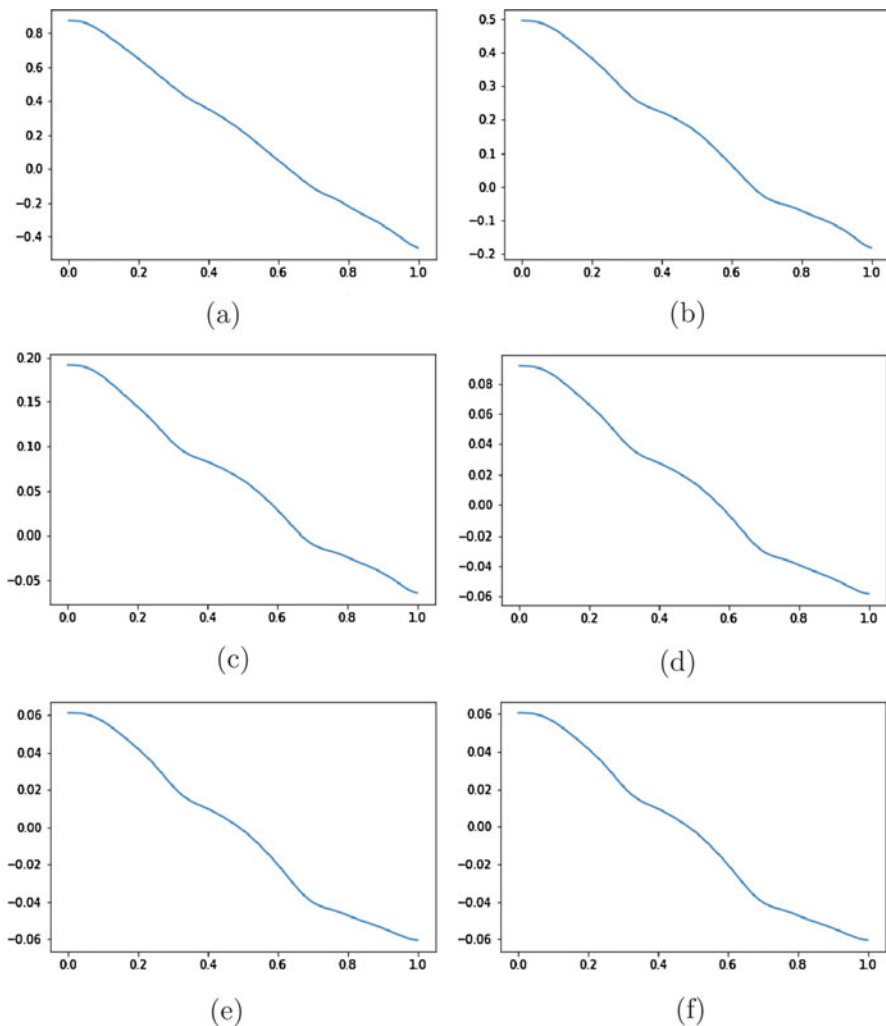


Fig. 5 Difference $\hat{\varphi}_\varepsilon - \hat{\psi}_\varepsilon$ of optimal potentials in $OT_\varepsilon(\mu, \nu)$ for increasing ε , where the normalized function $\hat{\varphi}_\infty - \hat{\psi}_\infty$ coincides with the optimal dual potential $\hat{\varphi}_K$ in the discrepancy by Corollary 2. (a) $\hat{\varphi}_{0.02} - \hat{\psi}_{0.02}$. (b) $\hat{\varphi}_{0.08} - \hat{\psi}_{0.08}$. (c) $\hat{\varphi}_{0.32} - \hat{\psi}_{0.32}$. (d) $\hat{\varphi}_{1.28} - \hat{\psi}_{1.28}$. (e) $\hat{\varphi}_{81.92} - \hat{\psi}_{81.92}$. (f) $\hat{\varphi}_\infty - \hat{\psi}_\infty$

$$\sum_{k=0}^N \alpha_k |\hat{\mu}_k - \hat{\nu}_k|^2, \quad N := 128$$

as target functional; see Gräf et al. (2013). The value of S_∞ for the Fourier method is slightly larger than the result using optimization of S_∞ directly. Since the computational cost increases as ε gets smaller, we suggest to choose $\varepsilon \approx 1$ or

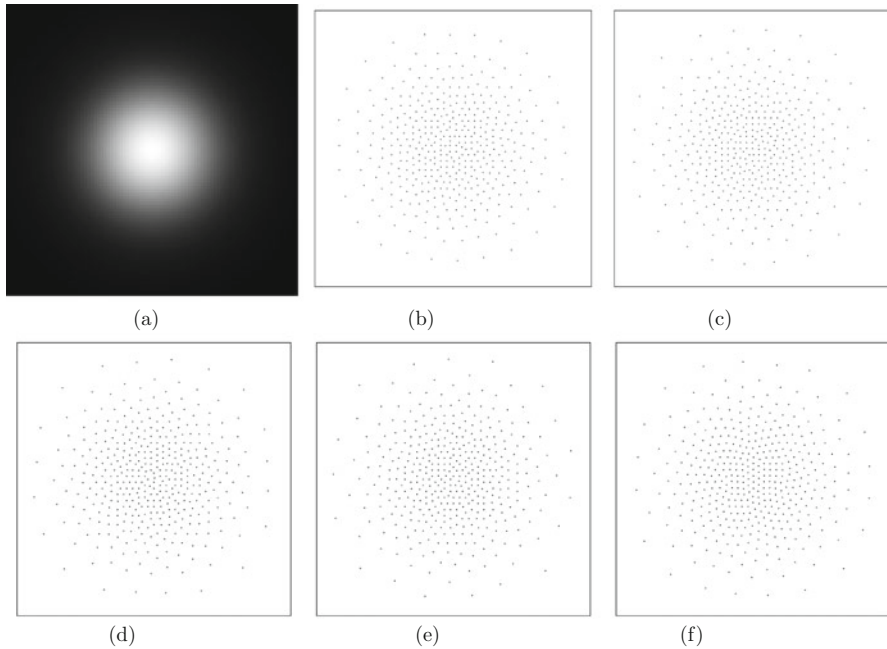


Fig. 6 Optimal approximations $\hat{\nu}$ and corresponding energies $S_{\varepsilon}(\mu, \hat{\nu})$ for increasing ε . **(a)** Fixed measure μ . **(b)** $S_{0.03}(\mu, \hat{\nu}) = 1.303e^{-3}$. **(c)** $S_{0.15}(\mu, \hat{\nu}) = 1.071e^{-4}$. **(d)** $S_{1.25}(\mu, \hat{\nu}) = 1.491e^{-5}$. **(e)** $S_{\infty}(\mu, \hat{\nu}) = 1.118e^{-5}$. **(f)** Fourier formulation (Ehler et al. 2019), $S_{\infty}(\mu, \hat{\nu}) = 1.156e^{-5}$

to directly stick with discrepancies. This also avoids that the approximation rates suffer from the so-called curse of dimensionality.

Finally, note that we sampled μ with a lot more points than we used for the dithering. If not enough points are used, we would observe clustering of the dithered measure around the positions of μ . One possibility to avoid such a behavior for S_{ε} could be to use the semi-discrete approach described in Genevay et al. (2016), avoiding any sampling of the measure μ . In the Fourier-based approach, this issue was less pronounced.

Conclusions

In this chapter, we examined the behavior of the Sinkhorn divergences S_{ε} as $\varepsilon \rightarrow \infty$ and $\varepsilon \rightarrow 0$, with focus on the first case, which leads to discrepancies for appropriate cost functions and kernels. We considered a quite general scenario of measures involving, e.g., convex combinations of measures with densities and point measures (spikes). Besides application questions, some open theoretical problems are left. While OT_{ε} is monotone increasing in ε for any cost function c , we observed numerically for $c(x, y) = \|x - y\|$ that S_{ε} is monotone decreasing. Further, in

Proposition 5 (ii), we were not able to show convergence of the whole sequence of optimal potentials $\{(\hat{\varphi}_\varepsilon, \hat{\psi}_\varepsilon)\}_\varepsilon$ without further assumptions so far.

Basic Theorems

We frequently apply the theorem of Arzelà–Ascoli. By definition, a sequence $\{f_n\}_{n \in \mathbb{N}}$ of continuous functions on \mathbb{X} is *uniformly bounded*, if there exists a constant $M \geq 0$ independent of n and x such that for all f_n and all $x \in \mathbb{X}$ it holds $|f_n(x)| \leq M$. The sequence is said to be *uniformly equi-continuous* if, for every $\varepsilon > 0$, there exists a $\delta > 0$ such that for all functions f_n

$$|f_n(x) - f_n(y)| < \varepsilon$$

whenever $d_{\mathbb{X}}(x, y) < \delta$.

Theorem 1 (Arzelà–Ascoli). *Let $\{f_n\}_{n \in \mathbb{N}}$ be a uniformly bounded and uniformly equi-continuous sequence of continuous functions on \mathbb{X} . Then, the sequence has a uniformly convergent subsequence.*

For the dual problems, we want to extend continuous functions from $A \subset \mathbb{X}$ to the whole space, which is possible by the following theorem. In the standard version, the theorem comes without the bounds, but they can be included directly since min and max of two continuous functions are again continuous functions.

Theorem 2 (Tietze Extension Theorem). *Let a closed subset $A \subset \mathbb{X}$ and a continuous function $f: A \rightarrow \mathbb{R}$ be given. If $g, h \in C(\mathbb{X})$ are such that $g \leq h$ and $g(x) \leq f(x) \leq h(x)$ for all $x \in A$, then there exists a continuous function $F: \mathbb{X} \rightarrow \mathbb{R}$ such that $F(x) = f(x)$ for all $x \in A$ and $g(x) \leq F(x) \leq h(x)$ for all $x \in \mathbb{X}$.*

References

- Ambrosio, L., Gigli, N., Savaré, G.: Gradient Flows in Metric Spaces and in the Space of Probability Measures. Birkhäuser, Basel (2005)
- Berman, R.J.: The Sinkhorn algorithm, parabolic optimal transport and geometric Monge–Ampère equations. *Numer. Math.* **145**(4), 771–836 (2020)
- Braides, A.: Γ -Convergence for Beginners. Oxford University Press, Oxford (2002)
- Bredies, K., Lorenz, D.: Mathematische Bildverarbeitung. Vieweg+Teuber, Wiesbaden (2011)
- Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **16**(5), 1190–1208 (1995)
- Carlier, G., Duval, V., Peyré, G., Schmitzer, B.: Convergence of entropic schemes for optimal transport and gradient flows. *SIAM J. Math. Anal.* **49**(2), 1385–1418 (2017)
- Chauffert, N., Ciuciu, P., Kahn, J., Weiss, P.: A projection method on measures sets. *Constr. Approx.* **45**(1), 83–111 (2017)

- Chevallier, J.: Uniform decomposition of probability measures: quantization, clustering and rate of convergence. *J. Appl. Probab.* **55**(4), 1037–1045 (2018)
- Chizat, L., Peyré, G., Schmitzer, B., Vialard, F.-X.: Scaling algorithms for unbalanced optimal transport problems. *Math. Comput.* **87**(314), 2563–2609 (2018)
- Clason, C., Lorenz, D., Mahler, H., Wirth, B.: Entropic regularization of continuous optimal transport problems. arXiv:1906.01333 (2019)
- Cominetti, R., San Martín, J.: Asymptotic analysis of the exponential penalty trajectory in linear programming. *Math. Program.* **67**(1–3), 169–187 (1994)
- Cucker, F., Smale, S.: On the mathematical foundations of learning. *Bull. Am. Math. Soc.* **39**(1), 1–49 (2002)
- Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: *Advances in Neural Information Processing Systems*, vol. 26, pp. 2292–2300 (2013)
- Cuturi, M., Peyré, G.: Computational optimal transport. *Found. Trends Mach. Learn.* **11**(5–6), 355–607 (2019)
- Delsarte, P., Goethals, J.M., Seidel, J.J.: Spherical codes and designs. *Geom. Dedicata* **6**, 363–388 (1977)
- Di Marino, S., Gerolin, A.: An optimal transport approach for the Schrödinger bridge problem and convergence of Sinkhorn algorithm. arXiv:1911.06850 (2019)
- Dziugaite, G.K., Roy, D.M., Ghahramani, Z.: Training generative neural networks via maximum mean discrepancy optimization. In: *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pp. 258–267 (2015)
- Ehler, M., Gräf, M., Neumayer, S., Steidl, G.: Curve based approximation of measures on manifolds by discrepancy minimization. arXiv:1910.06124 (2019)
- Ekeland, I., Témam, R.: *Convex Analysis and Variational Problems*. SIAM, Philadelphia (1999)
- Fernández, V.A., Gamero, M.J., García, J.M.: A test for the two-sample problem based on empirical characteristic functions. *Comput. Stat. Data Anal.* **52**(7), 3730–3748 (2008)
- Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S., Trounev, A., Peyré, G.: Interpolating between optimal transport and MMD using Sinkhorn divergences. In: *Proceedings of Machine Learning Research*, vol. 89, pp. 2681–2690. PMLR (2019)
- Genevay, A., Cuturi, M., Peyré, G., Bach, F.: Stochastic optimization for large-scale optimal transport. In: *Advances in Neural Information Processing Systems*, vol. 29, pp. 3440–3448 (2016)
- Genevay, A., Chizat, L., Bach, F., Cuturi, M., Peyré, G.: Sample complexity of Sinkhorn divergences. In: *Proceedings of Machine Learning Research*, vol. 89, pp. 1574–1583. PMLR (2019)
- Gnewuch, M.: Weighted geometric discrepancies and numerical integration on reproducing kernel Hilbert spaces. *J. Complex.* **28**(1), 2–17 (2012)
- Goes, F.D., Breeden, K., Ostromoukhov, V., Desbrun, M.: Blue noise through optimal transport. *ACM Trans. Graph.* **31**(6), 171–182 (2012)
- Gräf, M.: *Efficient Algorithms for the Computation of Optimal Quadrature Points on Riemannian Manifolds*. PhD thesis, TU Chemnitz (2013)
- Gräf, M., Potts, M., Steidl, G.: Quadrature errors, discrepancies and their relations to halftoning on the torus and the sphere. *SIAM J. Sci. Comput.* **34**(5), 2760–2791 (2013)
- Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. In: *Advances in Neural Information Processing Systems*, vol. 19, pp. 513–520 (2007)
- Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. *J. Mach. Learn. Res.* **13**(1), 723–773 (2012)
- Hytönen, T., van Neerven, J., Veraar, M., Weis, L.: *Analysis in Banach Spaces-Vol. I: Martingales and Littlewood-Paley Theory. A Series of Modern Surveys in Mathematics*, vol. 63. Springer, Cham (2016)
- Kuipers, L., Niederreiter, H.: *Uniform Distribution of Sequences*. Wiley, New York (1974)
- Lebrat, L., de Gournay, F., Kahn, J., Weiss, P.: Optimal transport approximation of 2-dimensional measures. *SIAM J. Imaging Sci.* **12**(2), 762–787 (2019)

- Léonard, C.: From the Schrödinger problem to the Monge–Kantorovich problem. *J. Funct. Anal.* **262**(4), 1879–1920 (2012)
- Liero, M., Mielke, A., Savaré, G.: Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Invent. Math.* **211**(3), 969–1117 (2018)
- Lorenz, D., Manns, P., Meyer, C.: Quadratically regularized optimal transport. *J. Math. Anal. Appl.* **494**, 124432 (2021)
- Matousek, J.: *Geometric Discrepancy. Algorithms and Combinatorics*, vol. 18. Springer, Berlin (2010)
- Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. R. Soc. Lond. Ser. A* **209**(441–458), 415–446 (1909)
- Micchelli, C.A.: Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.* **2**(1), 11–22 (1986)
- Navrotskaya, I., Rabier, P.J.: $L \log L$ and finite entropy. *Adv. Nonlinear Anal.* **2**(4), 379–387 (2013)
- Novak, E., Wozniakowski, H.: *Tractability of Multivariate Problems. Volume II. EMS Tracts in Mathematics*, vol. 12. EMS Publishing House, Zürich (2010)
- Peyré, G.: Entropic Wasserstein gradient flows. *SIAM J. Imaging Sci.* **8**(4), 2323–2351 (2015)
- Rüschendorf, L.: Convergence of the iterative proportional fitting procedure. *Ann. Stat.* **23**(4), 1160–1174 (1995)
- Santambrogio, F.: *Optimal Transport for Applied Mathematicians. Progress in Nonlinear Differential Equations and Their Applications*, vol. 87. Birkhäuser, Basel (2015)
- Schmaltz, C., Gwosdek, P., Bruhn, A., Weickert, J.: Electrostatic half-toning. *Comput. Graph. For.* **29**(8), 2313–2327 (2010)
- Schoenberg, I.J.: Metric spaces and completely monotone functions. *Ann. Math.* **39**(4), 811–841 (1938)
- Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.* **35**(2), 876–879 (1964)
- Steinwart, I., Christmann, A.: *Support Vector Machines*. Springer, New York (2008)
- Steinwart, I., Scovel, C.: Mercer’s theorem on general domains: on the interaction between measures, kernels, and RKHSs. *Constr. Approx.* **35**(3), 363–417 (2011)
- Teuber, T., Steidl, G., Gwosdek, P., Schmaltz, C., Weickert, J.: Dithering by differences of convex functions. *SIAM J. Imaging Sci.* **4**(1), 79–108 (2011)
- Vialard, F.-X.: *An elementary introduction to entropic regularization and proximal methods for numerical optimal transport. Lecture* (2019)
- Wendland, H.: *Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics*, vol. 17. Cambridge University Press, Cambridge (2004)
- Wilson, A.G.: The use of entropy maximising models in the theory of trip distribution, mode split and route split. *J. Transp. Econ. Policy* **3**(1), 108–126 (1969)
- Yule, G.U.: On the methods of measuring association between two attributes. *J. R. Stat. Soc.* **75**(6), 579–652 (1912)



Compensated Convex-Based Transforms for Image Processing and Shape Interrogation

51

Antonio Orlando, Elaine Crooks, and Kewei Zhang

Contents

Introduction	1828
Related Areas: Semiconvex Envelope	1832
Related Areas: Proximity Hull	1833
Related Areas: Mathematical Morphology	1834
Related Areas: Quadratic Envelopes	1835
Outline of the Chapter	1836
Notation and Preliminaries	1836
Compensated Convexity-Based Transforms	1843
Smoothing Transform	1843
Stable Ridge/Edge Transform	1844
Stable Multiscale Intersection Transform of Smooth Manifolds	1853
Stable Multiscale Medial Axis Map	1856
Approximation Transform	1859
Numerical Algorithms	1862
Convex-Based Algorithms	1862
Moreau Envelope-Based Algorithms	1865
Numerical Examples	1867
Prototype Example: Upper Transform of a Singleton Set of \mathbb{R}^2	1867

A. Orlando
CONICET, Departamento de Bioingeniería, Universidad Nacional de Tucumán, Tucumán,
Argentina
e-mail: aorlando@herrera.unt.edu.ar

E. Crooks
Department of Mathematics, Swansea University, Swansea, UK
e-mail: e.c.m.crooks@swansea.ac.uk

K. Zhang (✉)
School of Mathematical Sciences, University of Nottingham, Nottingham, UK
e-mail: kewei.zhang@nottingham.ac.uk

Intersection of Sampled Smooth Manifolds.....	1868
Approximation Transform.....	1871
Conclusions.....	1881
References.....	1883

Abstract

This paper reviews some recent applications of the theory of the compensated convex transforms or of the proximity hull as developed by the authors to image processing and shape interrogation with special attention given to the Hausdorff stability and multiscale properties. This paper contains also numerical experiments that demonstrate the performance of our methods compared to the state-of-art ones.

Keywords

Compensated convex transform · Moreau envelope · Proximity hull · Mathematical morphology · Hausdorff-Lipschitz continuity · Image processing · Shape interrogation · Scattered data

2000 Mathematics Subjects Classification number

90C25 · 90C26 · 49J52 · 52A41 · 65K10 · 62H35 · 14J17 · 58K25 · 53-XX · 65D17 · 53A05 · 26B25 · 52B55 · 65D18

Introduction

The compensated convex transforms were introduced in Zhang (2008a,b) for the purpose of tight approximation of functions defined in \mathbb{R}^n , and their definitions were originally motivated by the translation method (Tartar 1985) in the study of the quasiconvex envelope in the vectorial calculus of variations (see Dacorogna (2008) and references therein) and in the variational approach of material microstructure (Ball and James 1987). Thanks to their smoothness and tight approximation property, these transforms provide geometric convexity-based techniques for general functions that yield novel methods for identifying singularities in functions (Zhang et al. 2015a,b,c, 2016b) and new tools for function and image interpolation and approximation (Zhang et al. 2016a, 2018). In this paper, we present some of the applications that have been tackled by this theory up to date. These range from the detection of features in images or data (Zhang et al. 2015b,c, 2016b) to multiscale medial axis extraction (Zhang et al. 2015a), to surface reconstruction from level sets, to approximation of scattered data and noise removal from images, and to image inpainting (Zhang et al. 2016a, 2018).

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfies the following growth condition

$$f(x) \geq -A_1|x|^2 - A_2 \quad \text{for any } x \in \mathbb{R}^n, \quad (1)$$

for some constants $A_1, A_2 \geq 0$, the quadratic lower compensated convex transform (lower transform for short) for a given $\lambda > A_1$ is defined in Zhang (2008a) by

$$C_\lambda^l(f)(x) = \text{co} \left[\lambda |\cdot|^2 + f \right] (x) - \lambda |x|^2 \quad x \in \mathbb{R}^n, \tag{2}$$

where $|x|$ is the Euclidean norm of $x \in \mathbb{R}^n$ and $\text{co}[g]$ the convex envelope (Hiriart-Urruty and Lemaréchal 2001; Rockafellar 1970) of a function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ bounded below. Similarly, given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying the growth condition

$$f(x) \leq A_1 |x|^2 + A_2 \quad \text{for any } x \in \mathbb{R}^n, \tag{3}$$

for some constants $A_1, A_2 \geq 0$, the quadratic upper compensated convex transform (upper transform for short) for a given $\lambda > A_1$ is defined (Zhang 2008a) by

$$\begin{aligned} C_\lambda^u(f)(x) &= -C_\lambda^l(-f)(x) \\ &= \lambda |x|^2 - \text{co} \left[\lambda |\cdot|^2 - f \right] (x) \quad x \in \mathbb{R}^n. \end{aligned} \tag{4}$$

It is not difficult to verify that if f meets both (1) and (3), for instance, if f is bounded, there holds

$$C_\lambda^l(f)(x) \leq f(x) \leq C_\lambda^u(f)(x) \quad x \in \mathbb{R}^n,$$

thus, the lower and upper compensated convex transforms are λ -parametrized families of transforms that approximate f from below and above, respectively. Furthermore, they have smoothing effects and are tight approximations of f in the sense that if f is $C^{1,1}$ in a neighborhood of x_0 , there is a finite $\Lambda > 0$, such that $f(x_0) = C_\lambda^l(f)(x_0)$ (respectively, $f(x_0) = C_\lambda^u(f)(x_0)$) whenever $\lambda \geq \Lambda$. This approximation property, which we refer to as tight approximation, is pivotal in the developments of the theory, because it allows the transforms to be used for detecting singularities of functions by exploiting the fact that it is only when a point x is close to a singularity point of f we might find that the values of $C_\lambda^l(f)(x)$ and $C_\lambda^u(f)(x)$ might be different from that of $f(x)$ (Zhang et al. 2015b). Figure 1 visualizes the smoothing and tight approximation of the mixed transform $C_\lambda^u(C_\lambda^l(f))$ of the squared-distance function f to a four-point set. Given the type of singularity of f , we apply the lower transform to f which smoothes the ‘‘concave’’-like singularity followed by the upper transform that smoothes the ‘‘convex’’-like singularity of $C_\lambda^l(f)$ which are unaltered with respect to the original function f . This can be appreciated by the graph of the pointwise error $e(x) = |f(x) - C_\lambda^u(C_\lambda^l(f))(x)|$ for $x \in \Omega$ which is zero everywhere but in a neighborhood of the singularities of f .

The transforms additionally satisfy the locality property that the values of $C_\lambda^l(f)$, $C_\lambda^u(f)$ at $x \in \mathbb{R}^n$ depend only on the values of f in a neighborhood of x and are translation invariant in the sense that $C_\lambda^l(f)$, $C_\lambda^u(f)$ are unchanged if the ‘‘weight’’ $|\cdot|^2$ in the formula (2) and (4) is replaced by $|\cdot - x_0|^2$ for any shift $x_0 \in \mathbb{R}^n$.

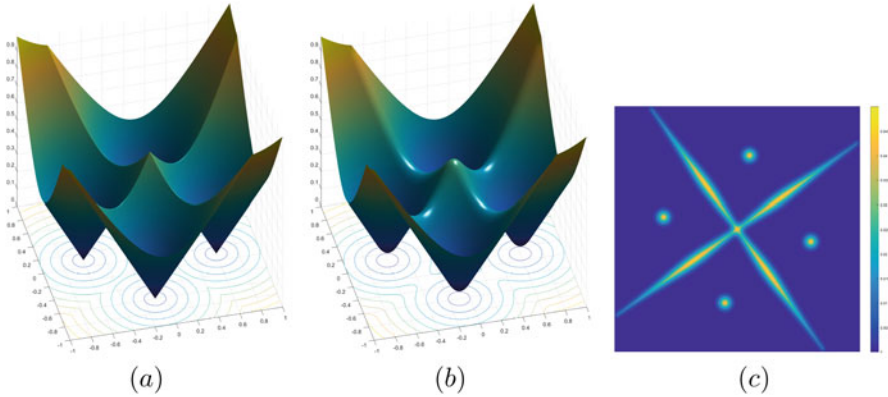


Fig. 1 Graph of (a) a squared-distance function f to a four-point set, (b) its mixed transform $C_\lambda^u(C_\lambda^l(f))$ and (c) the pointwise error $e = |f - C_\lambda^u(C_\lambda^l(f))|$

These last two properties make the explicit calculation of transforms tractable for specific prototype functions f , which facilitate the creation of dedicated extractors for a variety of different types of singularity using customized combinations of the transforms.

These new geometric approaches enjoy key advantages over previous image and data processing techniques (Chan and Shen 2005; Schönlieb 2015). The curvature parameter λ provides scales for features that allow users to select which size of feature they wish to detect, and the techniques are blind and global, in the sense that images/data are treated as a global object with no a priori knowledge required of, e.g., feature location. Figure 2 displays the λ –scale dependence in the case of the medial axis where λ is associated with the scale of the different branches, whereas Fig. 3 shows the multiscale feature for given λ associated with the height of the different branches of the multiscale medial axis map.

Many of the methods can also be shown to be stable under perturbation and different sampling techniques. Most significantly, Hausdorff stability results can be rigorously proven for many of the methods. For example, the Hausdorff-Lipschitz continuity estimate (Zhang et al. 2015b)

$$|C_\lambda^u(\chi_E)(x) - C_\lambda^u(\chi_F)(x)| \leq 2\sqrt{\lambda} \text{dist}_{\mathcal{H}}(E, F), \quad x \in \mathbb{R}^n,$$

shows that the upper transform C_λ^u is Hausdorff stable against sampling of geometric shapes defined by their characteristic functions. Such stability is particularly important for the extraction of information when “point clouds” represent sampled domains. If a geometric shape is densely sampled, then from a human vision point of view, one can typically still identify geometric features of the sample and sketch its boundary. From the mathematical/computer science perspective, however, feature identification from sampled domains is challenging, and usually methods are justified only by either ad hoc arguments or numerical experiments. Figure 4 displays an instance of this property where we show the edges of the

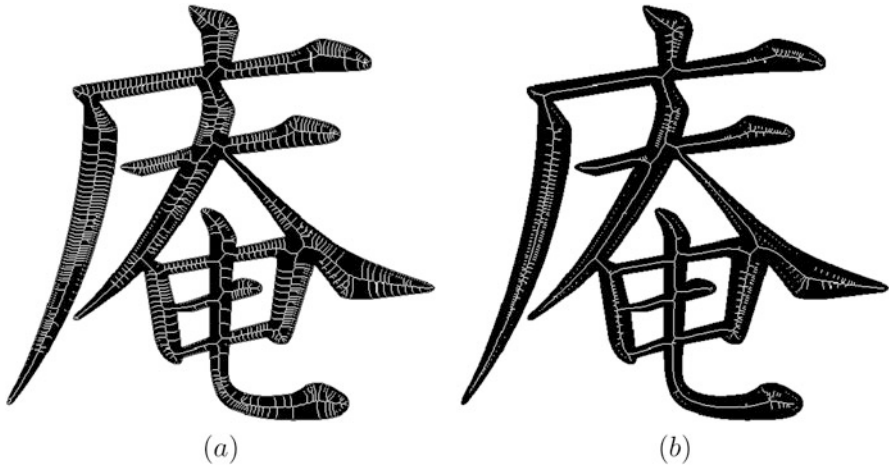


Fig. 2 Support of the multiscale medial axis map (suplevel set with level $t = 10^{-8} \max_{x \in \mathbb{R}^2} M_\lambda(\cdot; K)$) with the “spurious” branches generated by pixelation of the boundary for (a) $\lambda = 1$ and for (b) $\lambda = 8$

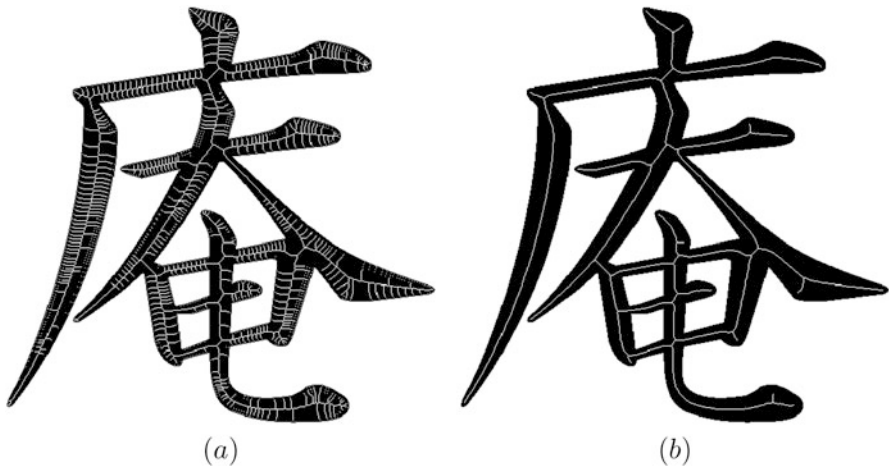


Fig. 3 Selection of branches via the suplevel set of the multiscale medial axis map for $\lambda = 1$ using different values of the threshold t , (a) $t = 10^{-3} \max_{x \in \mathbb{R}^2} M_\lambda(\cdot; K)$ and (b) $t = 2 \cdot 10^{-2} \max_{x \in \mathbb{R}^2} M_\lambda(\cdot; K)$

continuous nonnegative function $f(x, y) = \text{dist}^2((x, y), \partial\Omega)$, with $(x, y) \in \Omega = ([-1.5, 1.5] \times [-1.5, 1.5]) \setminus ([-1.5, 0.5] \times [-1.5, -0.5])$ and of its sparse sampling $f \cdot \chi_A$ where $A \subset \Omega$ is a sparse set (see Fig. 4a, b, respectively). Due to the Hausdorff stability of the stable ridge transform, we are able to recover an approximation of the ridges from the sampled image (compare Fig. 4c, d).

Via fast and robust numerical implementations of the transforms (Zhang et al. 2021), this theory also gives rise to a highly effective computational toolbox for

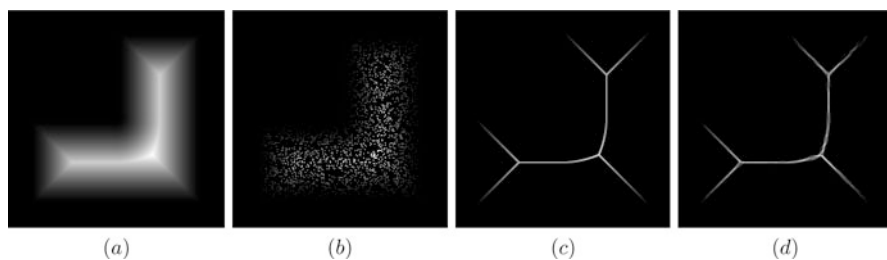


Fig. 4 (a) Image of $f(x, y)$; (b) sampled image of $f(x, y)$ by random salt and pepper noise; (c) stable ridges of $f(x, y)$; (d) stable ridges from sampled image

applications. The efficiency of the numerical computations benefits greatly from the locality property, which holds despite the global nature of the convex envelope itself.

Before we describe the applications of this theory, we provide next alternative characterizations of the compensated convex transforms.

Related Areas: Semiconvex Envelope

Given the definitions (2) and (4), lower and upper compensated convex transforms can be considered as parameterized semiconvex and semiconcave envelopes, respectively, for a given function. The notions of semiconvex and semiconcave functions go back at least to Reshetnyak (1956) and have since been studied by many authors in different contexts (see, e.g., Alberti et al. 1992; Cannarsa and Sinestrari 2004; Lasry and Lions 1986). Let $\Omega \subseteq \mathbb{R}^n$ be an open set; we recall that a function $f : \Omega \rightarrow \mathbb{R} \cup \{+\infty\}$ is semiconvex if there is a constant $C \geq 0$ such that $f(x) = g(x) - C|x|^2$ with g a convex function. More general weight functions, such as $|x|\sigma(|x|)$, for example, are also used in the literature for defining more general semiconvex functions (Alberti et al. 1992). Since general DC functions (difference of convex functions) (Hartman 1959) and semiconvex/semiconcave functions are locally Lipschitz functions in their essential domains (Cannarsa and Sinestrari 2004, Theorem 2.1.7), Rademacher's theorem implies that they are differentiable almost everywhere. Fine properties for the singular sets of convex/concave and semiconvex/semiconcave functions have been studied extensively (Alberti et al. 1992; Cannarsa and Sinestrari 2004) showing that the singular set of a semiconvex/semiconcave function is rectifiable. By applying results and tools of the theory of compensated convex transforms, it is possible therefore to study how such functions can be effectively approximated by smooth functions; whether all singular points are of the same type, that is, whether for semiconcave (semiconvex) functions, all singular points are geometric 'ridge' ('valley') points; how singular sets can be effectively extracted beyond the definition of differentiability; and how the information concerning "strengths" of different singular points can be effectively measured. These are all questions relevant to applications in image processing and

computer-aided geometric design. An instance of this study, for example, has been carried out in Zhang et al. (2015a, 2016b) to study the singular set of the Euclidean squared-distance function $\text{dist}^2(\cdot, \Omega^c)$ to the complement of a bounded open domain $\Omega \subset \mathbb{R}^n$ (called the medial axis (Blum 1967) of the domain Ω) and of the weighted squared-distance function.

Related Areas: Proximity Hull

Another characterization of the compensated convex transforms is in terms of the critical mixed Moreau envelopes, given that

$$C_\lambda^l(f)(x) = M^\lambda(M_\lambda(f))(x), \quad C_\lambda^u(f)(x) = M_\lambda(M^\lambda(f))(x), \quad (5)$$

where the Moreau lower and upper envelopes (Moreau 1965) are defined, in our notation, respectively, by

$$\begin{aligned} M_\lambda(f)(x) &= \inf\{f(y) + \lambda|y - x|^2, y \in \mathbb{R}^n\}, \\ M^\lambda(f)(x) &= \sup\{f(y) - \lambda|y - x|^2, y \in \mathbb{R}^n\}, \end{aligned} \quad (6)$$

with f satisfying the growth condition (1) and (3), respectively. Moreau envelopes play important roles in optimization, nonlinear analysis, optimal control, and Hamilton-Jacobi equations, both theoretically and computationally (Crandall et al. 1992; Cannarsa and Sinestrari 2004; Hiriart-Urruty and Lemaréchal 2001; Rockafellar and Wets 1998). The mixed Moreau envelopes $M^\tau(M_\lambda(f))$ and $M_\tau(M^\lambda(f))$ coincide with the Lasry-Lions double envelopes $(f_\lambda)^\tau$ and $(f^\lambda)_\tau$ defined in Lasry and Lions (1986) by (16) and (17), respectively, in the case of $\lambda = \tau$ and are also referred to in Rockafellar and Wets (1998) as proximal hull and upper proximal hull, respectively. They have been extensively studied and used as approximation and smoothing methods of not necessarily convex functions (Attouch and Aze 1993; Cannarsa and Sinestrari 2004; Hare 2009; Parikh and Boyd 2013). In particular, in the partial differential equation literature, the focus of the study of the mixed Moreau envelopes $M^\tau(M_\lambda(f))$ and $M_\tau(M^\lambda(f))$ for the case $\tau > \lambda$ is known, under suitable growth conditions, as the Lasry-Lions regularizations of f of parameter λ and τ . In this case, the mixed Moreau envelopes are both $C^{1,1}$ functions (Attouch and Aze 1993; Cannarsa and Sinestrari 2004; Lasry and Lions 1986). However, crucially they are not “tight approximations” of f , in contrast with our lower and upper transforms $C_\lambda^l(f)(x)$ and $C_\lambda^u(f)(x)$ (Zhang 2008a). Generalized inf and sup convolutions have also been considered, for instance, in Cannarsa and Sinestrari (2004) and Rockafellar and Wets (1998). However, due to the way these regularization operators are defined, proof of mathematical and geometrical results to describe how such approximations work has usually been challenging, making their analysis and applications very difficult. As a result, the study of the proximal

hull using the characterization in terms of the compensated convex transform would make them much more accessible and feasible for real-world applications.

Related Areas: Mathematical Morphology

Moreau lower and upper envelopes have also been employed in mathematical morphology in the 1990s (Jackway 1992; van den Boomgaard 1992), to define gray scale erosion and dilation morphological operators, whereas the critical mixed Moreau envelopes $M^\lambda(M_\lambda(f))$ and $M_\lambda(M^\lambda(f))$ are gray scale opening and closing morphological operators (Serra 1982; Soille 2004). In convex analysis, the infimal convolution of f with g is denoted as $f \square g$ and is defined as (Rockafellar 1970; Rockafellar and Wets 1998)

$$(f \square g)(x) = \inf_y \{f(y) + g(x - y)\}.$$

This is closely related to the erosion of f by g , given that

$$(f \square g)(x) = f(x) \ominus (-g(-x)).$$

Thus, if we denote by $b_\lambda(x) = -\lambda|x|^2$ the quadratic structuring function, introduced for the first time in Jackway (1992) and van den Boomgaard (1992), then with the notation of Serra (1982) and Soille (2004), we have

$$\begin{aligned} M_\lambda(f)(x) &= \inf_{y \in \mathbb{R}^n} \{f(y) - b_\lambda(y - x)\} =: f \ominus b_\lambda, \\ M^\lambda(f)(x) &= \sup_{y \in \mathbb{R}^n} \{f(y) + b_\lambda(y - x)\} =: f \oplus b_\lambda \end{aligned} \quad (7)$$

so that (5) can be written alternatively as

$$C_\lambda^l(f) = (f \ominus b_\lambda) \oplus b_\lambda \quad \text{and} \quad C_\lambda^u(f) = (f \oplus b_\lambda) \ominus b_\lambda. \quad (8)$$

The application of $M^\lambda(M_\lambda(f))$ and $M_\lambda(M^\lambda(f))$ in mathematical morphology (Serra 1982; Soille 2004), however, has not met with corresponding success, nor have its properties been fully explored. This is in contrast with the rôle, recognized since its introduction, that is played by paraboloid structuring functions in defining morphological scale-spaces in image analysis (Jackway 1992; van den Boomgaard 1992; Lindeberg 2011; Maragos and Schafer 1987; Weickert 1998). For this and related topics concerning the morphological scale-space representation produced by quadratic structuring functions, we refer to the pioneering works Jackway (1992) and van den Boomgaard (1992). Here, we would like only to observe that through identity (5), we have a direct characterization of the quadratic structuring-based opening and closing morphological operators, either in terms of the convex envelope (see (2) and (4)) or in terms of envelope from below/above with parabolas

(see (9) and (10)). Such characterizations will allow us to derive various new geometric and stability properties for opening and closing morphological operators. Furthermore, when we apply compensated convex transforms to extract singularities from characteristic functions of compact geometric sets, our operations can be viewed as the application of morphological operations devised for “gray scale images” to “binary images.” As a result, it might look not efficient to apply more involved operations for processing binary images, when in the current literature Serra (1982) and Soille (2004) there are “binary” set theoretic morphological operations that have been specifically designed for the tasks under examination. Nevertheless, an advantage of adopting our approach is that the compensated convex transforms of characteristic functions are (Lipschitz) continuous; therefore, applying a combination of transforms will produce a landscape of various levels (heights) that can be designed to highlight a specific type of singularity. We can then extract multiscale singularities by taking thresholds at different levels. In fact, the graphs of functions obtained by combinations of compensated convex transforms contain much more geometric information than binary operations that produce simply a yes or no answer. Also, for “thin” geometric structures, such as curves and surfaces, it is difficult to design “binary” morphological operations to be Hausdorff stable.

Related Areas: Quadratic Envelopes

From definition (2), it also follows that $C_\lambda^l(f)(x)$ is the envelope of all the quadratic functions with fixed quadratic term $\lambda|x|^2$ that are less than or equal to f , that is,

$$C_\lambda^l(f)(x) = \sup \left\{ -\lambda|x|^2 + \ell(x) : -\lambda|y|^2 + \ell(y) \leq f(y) \text{ for all } y \in \mathbb{R}^n \text{ and } \ell \text{ affine} \right\}, \quad (9)$$

whereas from (4) it follows that $C_\lambda^u(f)(x)$ is the envelope of all the quadratic functions with fixed quadratic term $\lambda|x|^2$ that are greater than or equal to f , that is,

$$C_\lambda^u(f)(x) = \inf \left\{ \lambda|x|^2 + \ell(x) : f(y) \leq \lambda|y|^2 + \ell(y) \text{ for all } y \in \mathbb{R}^n \text{ and } \ell \text{ affine} \right\}. \quad (10)$$

This characterization was first given in Zhang et al. (2015b, Eq. (1.4)) and can be derived by noting that since the convex envelope of a function g can be characterized as the pointwise supremum of the family $\text{Aff}(\mathbb{R}^n)$ of all the affine functions which are majorized by g , we have then

$$\begin{aligned}
C_\lambda^l(f)(x) &= \text{co}[f + \lambda|\cdot|](x) - \lambda|x|^2 \\
&= \sup_{\ell \in \text{Aff}(\mathbb{R}^n)} \left\{ \ell(x) : \ell(y) \leq f(y) + \lambda|y|^2 \text{ for any } y \in \mathbb{R}^n \right\} - \lambda|x|^2 \\
&= \sup_{\ell \in \text{Aff}(\mathbb{R}^n)} \left\{ \ell(x) - \lambda|x|^2 : \ell(y) - \lambda|y|^2 \leq f(y) \text{ for any } y \in \mathbb{R}^n \right\},
\end{aligned} \tag{11}$$

which is (9). As stated before, (11) can be in turn related directly to the Moreau's mixed envelope. The characterization (9) has been recently also reproposed by Carlsson (2019) for the study of low-rank approximation and compressed sensing.

It is instructive to compare this characterization with (75) below about the Moreau envelope as lower envelope of parabolas with given curvature λ .

Outline of the Chapter

The plan of this paper is as follows: After this general introduction, we will introduce relevant notation and recall basic results in convex analysis and compensated convex transforms in the next section. In section “[Compensated Convexity-Based Transforms](#)”, we introduce the different compensated convex-based transforms that we have been developing. Their definition can be either motivated by a mere application of key properties of the basic transforms, namely, the lower and upper transform, or by an ad hoc designed combinations of the basic transforms so to create a singularity at the location of the feature of interest. Section “[Numerical Algorithms](#)” introduces some of the numerical schemes that can be used for the numerical realization of the compensated convex-based transforms, namely, of the basic transform given by the lower compensated convex transform. We will therefore describe the convex-based and Moreau-based algorithms, which can be both used according to whether we refer to the definition (2) or the characterization (5) of the lower compensated convex transform. Section “[Numerical Examples](#)” contains some representative applications of the transformations introduced in this paper. More specifically, we will consider an application to shape interrogation by considering the problem of identifying the location of intersections of manifolds represented by point clouds and applications of our approximation compensated convex transform to the reconstruction of surfaces using level lines and isolated points, image inpainting, and salt & pepper noise removal.

Notation and Preliminaries

Throughout the paper \mathbb{R}^n denotes the n -dimensional Euclidean space, whereas $|x|$ and $x \cdot y$ are the standard Euclidean norm and inner product, respectively, for $x, y \in \mathbb{R}^n$. Given a non-empty subset K of \mathbb{R}^n , K^c denotes the complement of K in \mathbb{R}^n , i.e., $K^c = \mathbb{R}^n \setminus K$, \overline{K} its closure, $\text{co}[K]$ the convex hull of K , that is, the

smallest (with respect to inclusion) convex set that contains the set K and χ_K its characteristic function, that is, $\chi_K(x) = 1$ if $x \in K$ and $\chi_K(x) = 0$ if $x \in K^c$. The Euclidean distance transform of a non-empty set $K \subset \mathbb{R}^n$ is the function that, at any point $x \in \mathbb{R}^n$, associates the Euclidean distance of x to K , which is defined as $\inf\{|x - y|, y \in K\}$ and is denoted as $\text{dist}(x, K)$. Let $\delta > 0$, the open δ -neighborhood K^δ of K is then defined by $K^\delta = \{x \in \mathbb{R}^n, \text{dist}(x, K) < \delta\}$ and is an open set. For $x \in \mathbb{R}^n$ and $r > 0$, $B(x; r)$ indicates the open ball with center x and radius r , whereas $S(x; r)$ denotes the sphere with center x and radius r , that is, $S(x; r) = \partial B(x; r)$ is the boundary of $B(x; r)$. The suplevel set of a function $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ of level α is the set

$$S_\alpha f = \{x \in \Omega : f(x) \geq \alpha\}, \tag{12}$$

whereas the level set of f with level α is also defined by (12) with the inequality sign replaced by the equality sign. Finally, we use the notation Df to denote the derivative of f .

Next, we list some basic properties of compensated convex transforms. Without loss of generality, these properties are stated mainly for the lower compensated convex transform given that it is then not difficult to derive the corresponding results for the upper compensated convex transform using (4). Only in the case f is the characteristic function of a set K , i.e., $f = \chi_K$, we will refer explicitly to $C_\lambda^n(\chi_K)$ given that $C_\lambda^l(\chi_K)(x) = 0$ for any $x \in \mathbb{R}^n$ if K is, e.g., a finite set. For details and proofs, we refer to Zhang (2008a) and Zhang et al. (2015b) and references therein, whereas for the relevant notions of convex analysis, we refer to Hiriart-Urruty and Lemaréchal (2001) and Rockafellar (1970).

Definition 1. Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bounded below, the convex envelope $\text{co}[f]$ is the largest convex function not greater than f .

This is a global notion. By Carathéodory’s theorem (Hiriart-Urruty and Lemaréchal 2001; Rockafellar 1970), we have

$$\text{co}[f](x_0) = \inf_{\substack{x_i \in \mathbb{R}^n \\ i=1, \dots, n+1}} \left\{ \sum_{i=1}^{n+1} \lambda_i f(x_i) : \sum_{i=1}^{n+1} \lambda_i = 1, \sum_{i=1}^{n+1} \lambda_i x_i = x_0, \lambda_i \geq 0 \ i = 1, \dots, n + 1 \right\}, \tag{13}$$

that is, the convex envelope of f at a point $x_0 \in \mathbb{R}^n$ depends on the values of f on its whole domain of definition, namely, \mathbb{R}^n in this case. We will however introduce also a local version of this concept which will be used to formulate the locality property of the compensated convex transform and is fundamental for our applications.

Definition 2. Let $r > 0, x_0 \in \mathbb{R}^n$. Assume $f : B(x_0; r) \rightarrow \mathbb{R}$ to be bounded from below. Then the value of the local convex envelope of f at x_0 in $B(x_0; r)$ is defined by

$$\text{co}_{\overline{B}(x_0; r)}[f](x_0) = \inf_{\substack{x_i \in B(x_0; r) \\ i=1, \dots, n+1}} \left\{ \sum_{i=1}^{n+1} \lambda_i f(x_i) : \sum_{i=1}^{n+1} \lambda_i = 1, \sum_{i=1}^{n+1} \lambda_i x_i = x_0, \right. \\ \left. \lambda_i \geq 0 \ i = 1, \dots, n+1 \right\}. \tag{14}$$

Unlike the global definition, the infimum in (14) is taken only over convex combinations in $B(x_0; r)$ rather than in \mathbb{R}^n .

As part of the convex analysis reminder, we also recall the definition of the Legendre-Fenchel transform.

Definition 3. Let $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, $f \not\equiv +\infty$, and there is an affine function minorizing f on \mathbb{R}^n . The conjugate (or Legendre-Fenchel transform) of f is

$$f^* : s \in \mathbb{R}^n \rightarrow f^*(s) = \sup_{x \in \mathbb{R}^n} \{x \cdot s - f(x)\}, \tag{15}$$

and the biconjugate of f is $(f^*)^*$.

We have then the following results:

Proposition 1. For f satisfying the conditions of Definition 3, the conjugate f^* is a lower semicontinuous convex function, and $(f^*)^*$ is equal to the lower semicontinuous convex envelope of f .

Before stating the properties of interest of the compensated convex transforms, we describe the relationship between the compensated convex transforms and other infimal convolutions.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfy (1) and (3). As we have mentioned in the introduction, concepts closely related to the compensated convex transforms are the Lasry-Lions regularizations for parameters λ and τ with $0 < \tau < \lambda$, which are defined in Lasry and Lions (1986) as follows:

$$(f_\lambda)^\tau(x) = \sup_{y \in \mathbb{R}^n} \inf_{u \in \mathbb{R}^n} \{f(u) + \lambda|u - y|^2 - \tau|y - x|^2\} \\ = M^\tau(M_\lambda(f))(x), \tag{16}$$

and

$$(f^\lambda)_\tau(x) = \inf_{y \in \mathbb{R}^n} \sup_{u \in \mathbb{R}^n} \{f(u) - \lambda|u - y|^2 + \tau|y - x|^2\} \\ = M_\tau(M^\lambda(f))(x). \tag{17}$$

Both $(f_\lambda)^\tau$ and $(f^\lambda)_\tau$ approach f from below and above, respectively, as the parameters λ and τ go to $+\infty$. If $\lambda = \tau$, then $(f_\lambda)^\lambda = M^\lambda(M_\lambda(f))$ is called proximal hull of f , whereas $(f^\lambda)_\lambda = M_\lambda(M^\lambda(f))$ is referred to as the upper proximal hull of f . It is not difficult to verify that whenever $\tau > \lambda > 0$, the following relation holds between the compensated convex transforms, the Moreau envelopes, and the Lasry-Lions regularizations of f (Zhang 2008a),

$$M_\lambda(f)(x) \leq M^\lambda(M_\tau(f))(x) \leq C_\lambda^l(f)(x) \leq f(x) \quad \text{for } x \in \mathbb{R}^n,$$

and $f(x) \leq C_\tau^u(f)(x) \leq M_\lambda(M^\tau(f))(x) \leq M^\tau(f)(x) \quad \text{for } x \in \mathbb{R}^n.$ (18)

Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we recall also that the lower semicontinuous envelope of f is defined in Hiriart-Urruty and Lemaréchal (2001) and Rockafellar (1970) by

$$\underline{f} : x \in \mathbb{R}^n \mapsto \underline{f}(x) = \liminf_{y \rightarrow x} f(y),$$
 (19)

and since there holds

$$C_\lambda^l(f)(x) = C_\lambda^l(\underline{f})(x) \quad \text{for } x \in \mathbb{R}^n,$$
 (20)

without loss of generality, in the following we can assume that the functions are lower semicontinuous.

The monotonicity and approximation properties of $C_\lambda^l(f)$ with respect to λ are described by the following results:

Proposition 2. *Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies (1), then for all $A_1 < \lambda < \tau < \infty$, we have*

$$C_\lambda^l(f)(x) \leq C_\tau^l(f)(x) \leq f(x) \quad \text{for } x \in \mathbb{R}^n,$$
 (21)

and for $\lambda > A_1$

$$\lim_{\lambda \rightarrow \infty} C_\lambda^l(f)(x) = f(x) \quad \text{for } x \in \mathbb{R}^n.$$
 (22)

The approximation of f from below by $C_\lambda^l(f)$ given by (22) can be better specified, given that $C_\lambda^l(f)$ realizes a “tight” approximation of the function f in the following sense (see Zhang 2008a, Theorem 2.3(iv)).

Proposition 3. *Let $f \in C^{1,1}(\overline{B}(x_0; r))$, $x_0 \in \mathbb{R}^n$, $r > 0$. Then for sufficiently large $\lambda > 0$, we have that $f(x_0) = C_\lambda^l(f)(x_0)$. If the gradient of f is Lipschitz in \mathbb{R}^n with Lipschitz constant L , then $C_\lambda^l(f)(x) = f(x)$ for all $x \in \mathbb{R}^n$ whenever $\lambda \geq L$.*

The property of “tight” approximation plays an important role in the definition of the transforms introduced in section “Compensated Convexity-Based Transforms”.

Related to this property is the density property of the lower compensated transform established in Zhang et al. (2015b) that can be viewed as a tight approximation for general bounded functions.

Theorem 1. *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is bounded, satisfying $|f(x)| \leq M$ for some $M > 0$ and for all $x \in \mathbb{R}^n$. Let $\lambda > 0$, $x_0 \in \mathbb{R}^n$ and define $R_{\lambda, M} = (2 + \sqrt{2})\sqrt{M/\lambda}$. Then there are $x_i \in \bar{B}(x_0; R_{\lambda, M})$, with $x_i \neq x_0$, and $\lambda_i \geq 0$ for $i = 1, \dots, n + 1$, satisfying $\sum_{i=1}^{n+1} \lambda_i = 1$ and $\sum_{i=1}^{n+1} \lambda_i x_i = x_0$, such that*

$$C_\lambda^l(f)(x_i) = \underline{f}(x_i) \quad \text{for } i = 1, \dots, n + 1.$$

Since the lower transform satisfies

$$C_\lambda^l(f) \leq \underline{f} \leq f,$$

if we consider the following set

$$T_l(f, \lambda) = \{x \in \mathbb{R}^n : C_\lambda^l(f)(x) = \underline{f}(x)\},$$

as a result of Theorem 1, the set of points at which the lower compensated convex transform equals the original function satisfies a density property, that is, the closed $R_{\lambda, M}$ -neighborhoods of $T_l(f, \lambda)$ covers \mathbb{R}^n . For any point $x_0 \in \mathbb{R}^n$, the point x_0 is contained in the local convex hull $\text{co} [T_l(f, \lambda) \cap \bar{B}(x_0; R_{\lambda, M})]$. Furthermore, if f is bounded and continuous, $T_l(f, \lambda)$ is exactly the set of points at which f is λ -semiconvex (Cannarsa and Sinestrari 2004), i.e., points x_0 where

$$f(x) \geq f(x_0) + \ell(x) - \lambda|x - x_0|^2 \quad \text{for all } x \in \mathbb{R}^n$$

with ℓ an affine function satisfying $\ell(x_0) = 0$ and condition (1) holds for f .

A fundamental property for the applications is the locality of the compensated convex transforms. For a lower semicontinuous function that is in addition bounded on any bounded set, the locality property was established for this general case in Zhang (2008a). We next report its version for a bounded function which is relevant for the applications to image processing and shape interrogation (Zhang et al. 2015b).

Theorem 2. *Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is bounded, satisfying $|f(x)| \leq M$ for some $M > 0$ and for all $x \in \mathbb{R}^n$. Let $\lambda > 0$ and $x_0 \in \mathbb{R}^n$, then the following locality properties hold,*

$$C_\lambda^l(f)(x_0) = \inf \left\{ \sum_{i=1}^{n+1} \lambda_i (f(x_i) + \lambda |x_i - x_0|^2), \quad \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1, \sum_{i=1}^{n+1} \lambda_i x_i = x_0, \right. \\ \left. |x_i - x_0| \leq R_{\lambda, M} \right\}, \tag{23}$$

where $R_{\lambda, M}$ is the same as in Theorem 1.

Since the convex envelope is affine invariant, it is not difficult to realize that there holds

$$C_\lambda^l(f)(x_0) = \text{co}[\lambda|\cdot - x_0|^2 + f](x_0) \quad \text{for } x_0 \in \mathbb{R}^n; \tag{24}$$

thus condition (23) can be equivalently written as

$$C_\lambda^l(f)(x_0) = \text{co}_{\overline{B}(x_0; R_{\lambda, M})} [\lambda|\cdot - x_0|^2 + f](x_0). \tag{25}$$

Despite the definition of $C_\lambda^l(f)$ involves the convex envelope of $f + \lambda|\cdot|^2$, the value of the lower transform for a bounded function at a point depends on the values of the function in its $R_{\lambda, M}$ -neighborhood. Therefore, when λ is large, the neighborhood will be very small. If f is globally Lipschitz, this result is a special case of Lemma 3.5.7 at p. 72 of Cannarsa and Sinestrari (2004).

The following property shows that the mapping $f \rightarrow C_\lambda^l(f)$ is nondecreasing, that is, we have

Proposition 4. *If $f \leq g$ in \mathbb{R}^n and satisfy (1), then*

$$C_\lambda^l(f)(x) \leq C_\lambda^l(g)(x) \quad \text{for } x \in \mathbb{R}^n \text{ and } \lambda \geq \max\{A_{1,f}, A_{1,g}\}.$$

We conclude this section by stating some results on the Hausdorff stability of the compensated convex transforms. This is the relevant concept of stability we use to assess the change of the transformations with respect to perturbations of the set; thus, it refers to the behavior of the compensated convex transform of the characteristic functions of subsets K of \mathbb{R}^n . We first state a result that highlights the geometric structure of the upper transform of χ_K .

Theorem 3 (Expansion Theorem). *Let $E \subset \mathbb{R}^n$ be a non-empty set and let $\lambda > 0$ be fixed, and then*

$$C_\lambda^u(\chi_E)(x) \begin{cases} = 1, & \text{if } x \in \bar{E}, \\ = 0, & \text{if } x \in (\bar{E}^{1/\sqrt{\lambda}})^c, \\ \in (0, 1), & \text{if } x \in E^{1/\sqrt{\lambda}} \setminus \bar{E}. \end{cases}$$

Next, we recall the definition of Hausdorff distance from Ambrosio and Tilli (2004).

Definition 4. Let E, F be non-empty subsets of \mathbb{R}^n . The Hausdorff distance between E and F is defined by

$$\text{dist}_{\mathcal{H}}(E, F) = \inf \left\{ \delta > 0 : F \subset E^\delta \text{ and } E \subset F^\delta \right\}.$$

This definition is also equivalent to saying that

$$\text{dist}_{\mathcal{H}}(E, F) = \max \left\{ \sup_{x \in E} \text{dist}(x, F), \sup_{x \in F} \text{dist}(x, E) \right\}.$$

It is well-known and easy to prove that the Euclidean distance function $\text{dist}(x, K)$ is Hausdorff-Lipschitz continuous in the sense that for given K and $S \subset \mathbb{R}^n$ non-empty compact sets, we have

$$|\text{dist}(x, K) - \text{dist}(x, S)| \leq \text{dist}_{\mathcal{H}}(K, S).$$

In order to study the Hausdorff-Lipschitz continuity of the upper compensated convex transform of characteristic functions of compact sets, we introduce the distance-based function $D_\lambda^2(x, K)$ defined by

$$D_\lambda^2(x, K) = \left(\max \left\{ 0, 1 - \sqrt{\lambda} \text{dist}(x, K) \right\} \right)^2, \quad x \in \mathbb{R}^n. \tag{26}$$

Clearly, we have $0 \leq D_\lambda^2(x, K) \leq 1$ in \mathbb{R}^n . More precisely, we have

$$D_\lambda^2(x, K) \begin{cases} = 1, & \text{if } x \in K, \\ = 0, & \text{if } \text{dist}(x, K) \geq \frac{1}{\sqrt{\lambda}}, \\ \in (0, 1), & \text{if } 0 < \text{dist}(x, K) < \frac{1}{\sqrt{\lambda}}. \end{cases} \tag{27}$$

Suppose $E, F \subset \mathbb{R}^n$ are two non-empty closed sets. It is, then, easy to see that

(i) if $E \subset F$,

$$D_\lambda^2(x, E) \leq D_\lambda^2(x, F), \quad x \in \mathbb{R}^n; \tag{28}$$

(ii) for $x \in \mathbb{R}^n$, if $E \cap \bar{B}(x, 1/\sqrt{\lambda}) \neq \emptyset$, then

$$D_\lambda^2(x, E) = D_\lambda^2(x, E \cap \bar{B}(x, 1/\sqrt{\lambda})). \tag{29}$$

For a given non-empty closed set K , by definition of the function $D_\lambda^2(x, K)$, we have

$$0 \leq \chi_K(x) \leq D_\lambda^2(x, K) \leq 1, \quad x \in \mathbb{R}^n .$$

The following result establishes the relationship between the upper transform of $\chi_K(x)$ and $D_\lambda^2(x, K)$ and it was established in Zhang et al. (2015b).

Proposition 5. *Let $K \subset \mathbb{R}^n$ be a non-empty closed set and assume $\lambda > 0$. Then, there holds*

$$C_\lambda^u(\chi_K)(x) = C_\lambda^u(D_\lambda^2(\cdot, K))(x), \quad x \in \mathbb{R}^n . \tag{30}$$

The Hausdorff-Lipschitz continuity of $C_\lambda^u(\chi_K)(x)$ and $C_\lambda^u(D_\lambda^2(\cdot, K))(x)$ were also established in Zhang et al. (2015b).

Theorem 4. *Let $E, F \subset \mathbb{R}^n$ be non-empty compact sets and let $\lambda > 0$ be fixed, then for all $x \in \mathbb{R}^n$,*

$$|D_\lambda^2(x, E) - D_\lambda^2(x, F)| \leq 2\sqrt{\lambda} \text{dist}_{\mathcal{H}}(E, F), \tag{31}$$

$$|C_\lambda^u(D_\lambda^2(\cdot, E))(x) - C_\lambda^u(D_\lambda^2(\cdot, F))(x)| \leq 2\sqrt{\lambda} \text{dist}_{\mathcal{H}}(E, F). \tag{32}$$

Consequently,

$$|C_\lambda^u(\chi_E)(x) - C_\lambda^u(\chi_F)(x)| \leq 2\sqrt{\lambda} \text{dist}_{\mathcal{H}}(E, F). \tag{33}$$

Compensated Convexity-Based Transforms

The lower compensated convex transform (2) and the upper compensated convex transform (4) represent building blocks for defining novel transformations to smooth functions, to identify singularities in functions, and to interpolate and approximate data. For the creation of these transformations, we follow mainly two approaches. One approach makes a direct use of the basic transforms to single out singularities of the function or to smooth and/or approximate the function. By contrast, the other approach realizes a suitably designed combination of the basic transforms that creates the singularity at the location of the feature of interest.

Smoothing Transform

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfy a growth condition of the form

$$|f(x)| \leq C_1|x|^2 + C_2 \tag{34}$$

for some $C_1, C_2 > 0$, then given $\lambda, \tau > C_1$, we can define two (quadratic) mixed compensated convex transform as follows:

$$C_{\tau,\lambda}^{u,l}(f)(x) := C_{\tau}^u(C_{\lambda}^l(f))(x) \quad \text{and} \quad C_{\lambda,\tau}^{l,u}(f)(x) := C_{\lambda}^l(C_{\tau}^u(f))(x), \quad x \in \mathbb{R}^n. \tag{35}$$

From (4), we have that for every $\lambda, \tau > C_1$

$$C_{\tau,\lambda}^{u,l}(f)(x) = -C_{\tau,\lambda}^{l,u}(-f)(x). \tag{36}$$

Hence, properties of $C_{\tau,\lambda}^{l,u}(f)$ follow from those for $C_{\tau,\lambda}^{u,l}(f)$, and we can thus state appropriate results only for $C_{\tau,\lambda}^{u,l}(f)$. In this case, then, whenever $\tau, \lambda > C_1$ we have that $C_{\tau,\lambda}^{u,l}(f) \in C^{1,1}(\mathbb{R}^n)$. As a result, if f is bounded, then $C_{\tau,\lambda}^{u,l}(f) \in C^{1,1}(\mathbb{R}^n)$ and $C_{\tau,\lambda}^{l,u}(f) \in C^{1,1}(\mathbb{R}^n)$ for all $\lambda > 0$ and $\tau > 0$. This is important in applications of the mixed transforms to image processing, because there the function representing the image takes a value from a fixed range at each pixel point and so is always bounded. The regularizing effect of the mixed transform is visualized in Fig. 5 where we display $C_{\lambda,\tau}^{l,u}(f)$ of the no-differentiable function $f(x, y) = |x| - |y|$, $(x, y) \in [-1, 1] \times [-1, 1]$ and of $f(x, y) + n(x, y)$ with $n(x, y)$ a bivariate normal distribution with mean value equal to 0.05. The level lines of $C_{\lambda,\tau}^{l,u}(f)$ and $C_{\lambda,\tau}^{l,u}(f + n)$ displayed in Fig. 5b and d, respectively, are smooth curves.

Finally, as a consequence of the approximation result (22) and likewise result for $C_{\tau}^u(f)$ (see Proposition 2), it is then not trivial to establish a similar approximation result also for the mixed transforms and verify that there are $\tau_j, \lambda_j \rightarrow \infty$ as $j \rightarrow \infty$ such that on every compact subset of \mathbb{R}^n , there holds

$$C_{\tau_j}^u(C_{\lambda_j}^l(f)) \rightarrow f \quad \text{uniformly as } j \rightarrow \infty. \tag{37}$$

Stable Ridge/Edge Transform

The ridge, valley, and edge transforms introduced in Zhang et al. (2015b) are basic operations for extracting geometric singularities. The key property is the tight approximation of the compensated convex transforms (see Proposition 3) and the approximation to f from below by $C_{\lambda}^l(f)$ and above by $C_{\lambda}^u(f)$, respectively.

Basic Transforms

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfy the growth condition (34). The ridge $R_{\lambda}(f)$, the valley $V_{\lambda}(f)$, and the edge transforms $E_{\lambda}(f)$ of scale $\lambda > C_1$ are defined, respectively, by

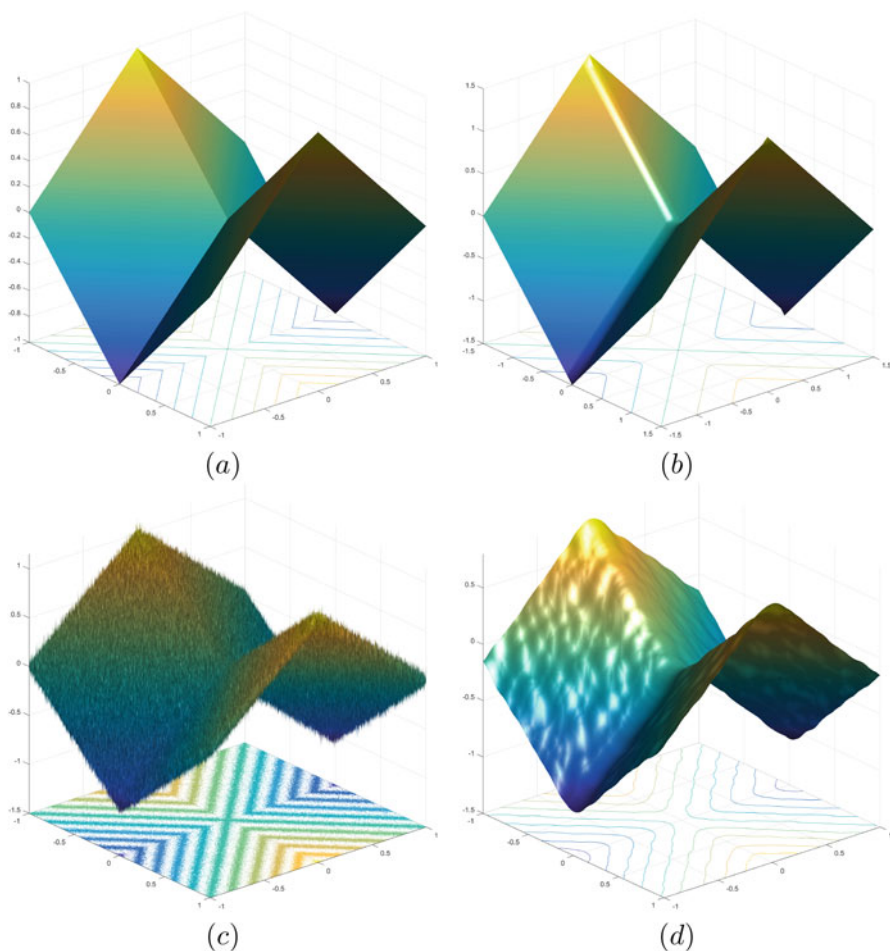


Fig. 5 (a) Input function $f(x, y) = |x| - |y|$; (b) graph of $C_{\lambda, \tau}^{l, u}(f)$ for $\lambda = 5$ and $\tau = 5$; (c) input function $f(x, y) + n(x, y)$ with $n(x, y)$ a bivariate normal distribution with mean value equal to 0.05; (d) graph of $C_{\lambda, \tau}^{l, u}(f + n)$ for $\lambda = 5$ and $\tau = 5$

$$\begin{aligned}
 R_{\lambda}(f) &= f - C_{\lambda}^l(f); & V_{\lambda}(f) &= f - C_{\lambda}^u(f); \\
 E_{\lambda}(f) &= R_{\lambda}(f) - V_{\lambda}(f) = C_{\lambda}^u(f) - C_{\lambda}^l(f).
 \end{aligned}
 \tag{38}$$

If f is of sub-quadratic growth, that is, $|f(x)| \leq A(1 + |x|^{\alpha})$ with $0 \leq \alpha < 2$, in particular f can be a bounded function, the requirement for λ in (38) is simply $\lambda > 0$.

The ridge transform $R_{\lambda}(f) = f - C_{\lambda}^l(f)$ and the valley transform $V_{\lambda}(f) = f - C_{\lambda}^u(f)$ are nonnegative and nonpositive, respectively, because of the ordering property of the compensated convex transforms and their support set is disjoint to

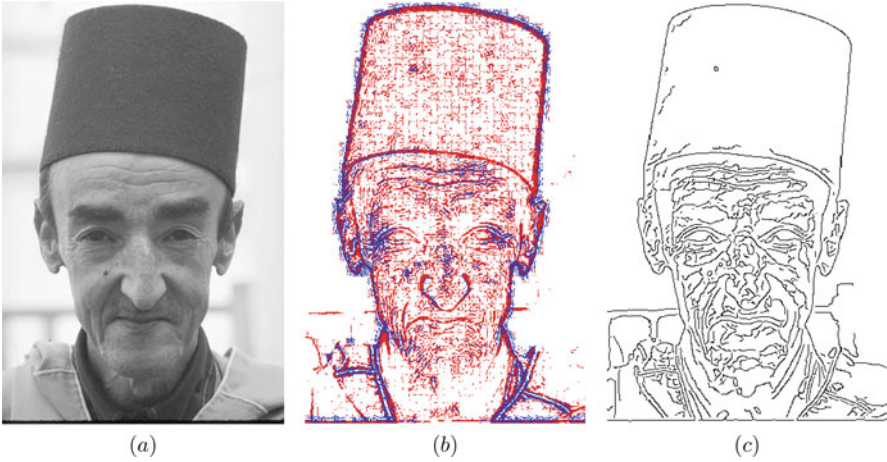


Fig. 6 (a) Input image; (b) suplevel set of the ridge and valley transform with $\lambda = 2.5$ and for the level equal to $0.005 \cdot \max [R_\lambda(f)]$ and $0.005 \cdot \max [-V_\lambda(f)]$, respectively; (c) Canny edges

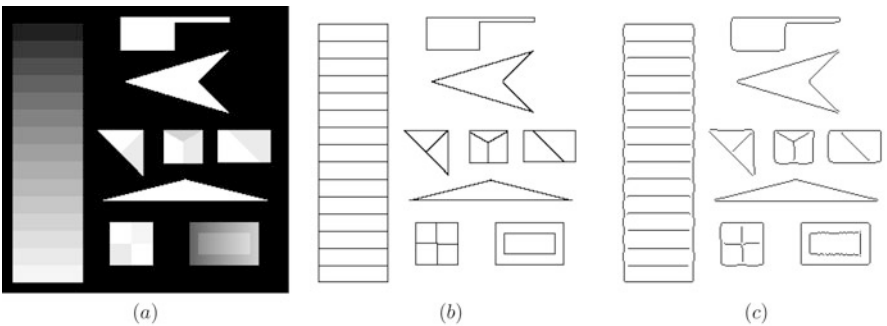


Fig. 7 (a) Input test image from Smith and Brady (1997); (b) suplevel set of the ridge transform with $\lambda = 0.1$ and for the level equal to $0.004 \cdot \max [R_\lambda(f)]$; (c) Canny edges

each other. In the applications, we usually consider $-V_\lambda(f)$ to make the resulting function nonnegative. Figure 6 displays the suplevel set of $R_\lambda(f)$ and $-V_\lambda(f)$ of the same level for a gray scale image f compared to the Canny edge filter, whereas Fig. 7 demonstrates on the test image used in Smith and Brady (1997) the ability of $R_\lambda(f)$ to detect edges between different gray levels.

The transforms $R_\lambda(f)$ and $V_\lambda(f)$ satisfy the following properties:

- (i) The transforms $R_\lambda(f)$ and $V_\lambda(f)$ are invariant with respect to translation, in the sense that

$$R_\lambda(f + \ell) = R_\lambda(f) \quad \text{and} \quad V_\lambda(f + \ell) = V_\lambda(f) \quad (39)$$

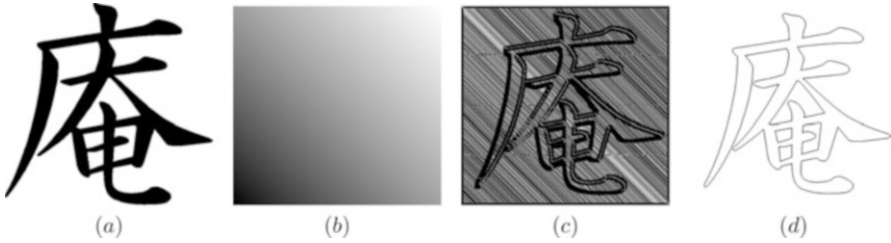


Fig. 8 (a) A binary image χ of a Chinese character; (b) image $255\chi + \ell$ with $\ell = 70(i - j)$ for $1 \leq i \leq 546, 1 \leq j \leq 571$, i.e., the scaled characteristic function of the character plus an affine function; (c) edges extracted by Canny edge detector; (d) edges extracted by the edge transform $E_\lambda(f)$ with $\lambda = 0.1$ after thresholding

for all affine functions $\ell \in \text{Aff}(\mathbb{R}^n)$. Consequently, the edge transform $E_\lambda(f)$ is also invariant with respect to translation.

(ii) The transforms $R_\lambda(f)$ and $V_\lambda(f)$ are scale covariant in the sense that

$$R_\lambda(\alpha f) = \alpha R_{\lambda/\alpha}(f) \quad \text{and} \quad V_\lambda(\alpha f) = \alpha V_{\lambda/\alpha}(f) \tag{40}$$

for all $\alpha > 0$. Consequently, the edge transform $E_\lambda(f)$ is also scale covariant.

(iii) The transforms $R_\lambda(f)$, $V_\lambda(f)$, and $E_\lambda(f)$ are all stable under curvature perturbations in the sense that for any $g \in C^{1,1}(\mathbb{R}^n)$ satisfying $|Dg(x) - Dg(y)| \leq \epsilon|x - y|$, if $\lambda > \epsilon$ then

$$\begin{aligned} R_{\lambda+\epsilon}(f) &\leq R_\lambda(f + g) \leq R_{\lambda-\epsilon}(f); & V_{\lambda-\epsilon}(f) &\leq V_\lambda(f + g) \leq V_{\lambda+\epsilon}(f); \\ E_{\lambda+\epsilon}(f) &\leq E_\lambda(f + g) \leq E_{\lambda-\epsilon}(f). \end{aligned} \tag{41}$$

The numerical experiments depicted in Fig. 8 illustrate the affine invariance of the edge transform expressed by (39), whereas Fig. 9 shows implications of the stability of the edge transform under curvature perturbations according to (41).

To get an insight on the geometric structure of the edge transform, it is informative to consider the case where f is the characteristic function of a set. Let $\Omega \subset \mathbb{R}^n$ be a non-empty open regular set such that $\bar{\Omega} \neq \mathbb{R}^n$ and $\Gamma \subset \partial\Omega$, then for $\lambda > 0$, we have that (Zhang et al. 2015b)

$$E_\lambda(\chi_{\Omega \cup \Gamma})(x) \begin{cases} = 0 & x \in (\Omega^{1/\sqrt{\lambda}})^c \cup \Omega \setminus (\Omega^c)^{1/\sqrt{\lambda}} \\ \in (0, 1) & x \in \Omega^{1/\sqrt{\lambda}} \setminus \bar{\Omega} \cup (\Omega^c)^{1/\sqrt{\lambda}} \setminus \Omega^c \\ = 1 & x \in \partial\Omega. \end{cases} \tag{42}$$

Furthermore, $E_\lambda(\chi_{\Omega \cup \Gamma})$ is continuous in \mathbb{R}^n , and, for $x \in \mathbb{R}^n$, there holds

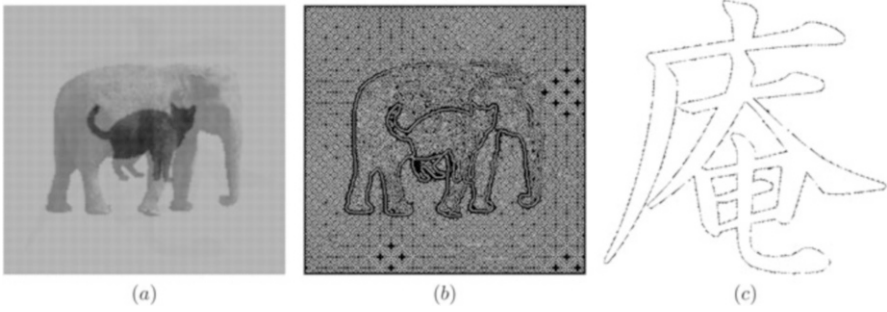


Fig. 9 (a) A scaled binary image of a Chinese character perturbed by a smooth image; (b) edges extracted by Canny edge detector; (c) edges extracted by the edge transform $E_\lambda(f)$ after thresholding

$$\lim_{\lambda \rightarrow +\infty} E_\lambda(\chi_{\Omega \cup \Gamma})(x) = \chi_{\partial\Omega}(x), \tag{43}$$

that is, λ controls the width of the neighborhood of $\chi_{\partial\Omega}$. As $\lambda \rightarrow \infty$, the support of $E_\lambda(\chi_{\overline{\Omega}})$ shrinks to the support of $\chi_{\partial\Omega}$.

Figure 10 illustrates the behavior of $E_\lambda(\chi_{\overline{\Omega}})$ by displaying the support of $E_\lambda(\chi_{\overline{\Omega}})$ for different values of λ .

Since the original function f is directly involved in the definitions of the ridge, valley, and edge transforms, the transforms (38) are not Hausdorff stable if we consider a dense sampling of the original function. It is possible nevertheless to establish stable versions of ridge and valley transforms in the case that f is the characteristic function χ_E of a non-empty compact set $E \subset \mathbb{R}^n$. For this result, it is fundamental the observation on the Hausdorff stability of the upper transform of the characteristic function χ_E of non-empty compact subsets of \mathbb{R}^n (see Zhang et al. 2015b, Theorem 5.5) which motivates the definition of stable ridge transform of E as

$$SR_{\tau,\lambda}(\chi_E) = C_\lambda^u(\chi_E) - C_\tau^l(C_\lambda^u(\chi_E)). \tag{44}$$

For the ridge defined by (44), we have that if E, F are non-empty compact subsets of \mathbb{R}^n , for $\lambda > 0$ and $\tau > 0$, then there holds

$$|SR_{\lambda,\tau}(\chi_E)(x) - SR_{\lambda,\tau}(\chi_F)(x)| \leq 4\sqrt{\lambda} \text{dist}_{\mathcal{H}}(E, F) \quad (\text{for } x \in \mathbb{R}^n). \tag{45}$$

Figure 11 illustrates the meaning of (45). Figure 11a displays a domain E represented by a binary image of an elephant, and (c) shows a domain F obtained by randomly sampling E , whereas (b) and (d) picture a suplevel set of the stable ridge transforms of the respective characteristic functions. Similarly to the stable ridge transform of a non-empty compact subset E of \mathbb{R}^n , we can then define the stable valley transform of E for $\lambda > \tau$ as

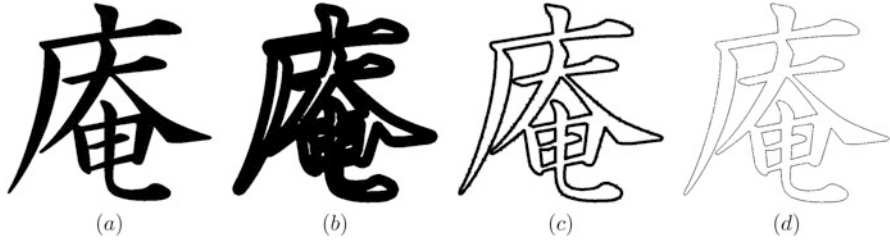


Fig. 10 Scale effect associated with λ on the support of the edge transform of the (a) image $f = 255 \cdot \chi$ of a Chinese character for different values of λ : (b) $\lambda = 1$; (c) $\lambda = 10$; (d) $\lambda = 100$

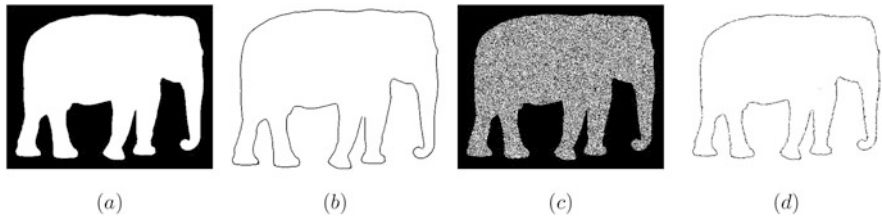


Fig. 11 (a) Domain E given by the image of an elephant displayed here as $1 - \chi_E$; (b) boundary extraction using the stable ridge transform, $SR_{\lambda,\tau}(\chi_E)$, for $\lambda = 0.1$ and $\tau = \lambda/8$; (c) domain F obtained by randomly sampling E ; (d) boundary extraction of the data sample after thresholding the stable ridge transform, $SR_{\lambda,\tau}(\chi_F)$, computed for $\lambda = 0.1$ and $\tau = \lambda/8$

$$SV_{\lambda,\tau}(\chi_E)(x) = V_\tau(C_\lambda^u(\chi_E))(x) \quad x \in \mathbb{R}^n, \quad \lambda > \tau > 0,$$

and the stable edge transform of E for $\lambda > \tau$ as

$$SE_{\lambda,\tau}(\chi_E)(x) = E_\tau(C_\lambda^u(\chi_E))(x) \quad x \in \mathbb{R}^n, \quad \lambda > \tau > 0.$$

The condition $\lambda > \tau$ is invoked because it is not difficult to see that

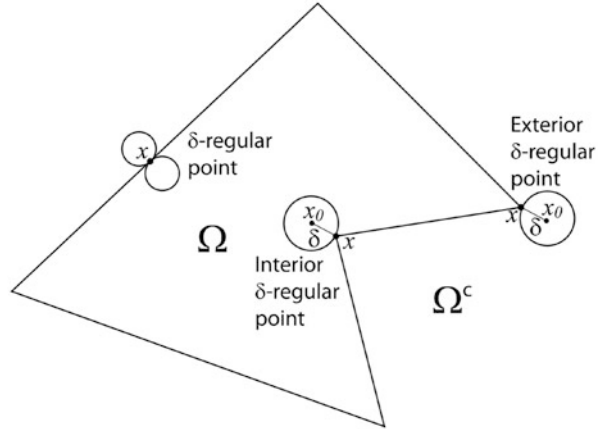
$$C_\tau^u(C_\lambda^u(f)) = \begin{cases} C_\lambda^u(f), & \text{for } \lambda \leq \tau \\ C_\tau^u(f), & \text{for } \lambda \geq \tau. \end{cases}$$

Hence, if $\lambda \leq \tau$, we would get $SV_{\lambda,\tau}(\chi_E)(x) = 0$, and $SE_{\lambda,\tau}(\chi_E)(x)$ would simply be equal to $SR_{\lambda,\tau}(\chi_E)(x)$.

Extractable Corner Points

Let $\Omega \subset \mathbb{R}^n$ be a bounded open set with $|\partial\Omega| = 0$ (i.e., $\partial\Omega$ has zero n -dimensional measure) and $x \in \partial\Omega$. We say that the point $x \in \partial\Omega$ is a δ -regular point of $\partial\Omega$ if there is an open ball $B(x_0; \delta) \subset \bar{\Omega}^c$, $x_0 \in \Omega^c$, $\delta > 0$, such that $x \in \partial B(x_0; \delta)$ and if there is an open ball $B(x_0; \delta) \subset \Omega$, $x_0 \in \Omega$, $\delta > 0$, such that $x \in \partial B(x_0; \delta)$. If the point $x \in \partial\Omega$ meets only the first condition, we refer to it as exterior δ -regular

Fig. 12 Exterior and interior δ -regular point of $\partial\Omega$



point, whereas if it meets only the second condition, it is called interior δ -regular point. Figure 12 displays the different types of points of $\partial\Omega$.

The stable ridge transform allows the characterization of such points given that if $x \in \partial\Omega$ is a δ -regular point of Ω with $\delta > 0$ sufficiently small, in Zhang et al. (2015b) it is shown that there holds

$$SR_{\lambda,\tau}(\chi_{\bar{\Omega}})(x_0) \leq \frac{(\sqrt{\lambda + \tau} - \sqrt{\tau})^2}{\lambda}. \tag{46}$$

As a result, we define an extractable corner point of Ω if for at least sufficiently large $\lambda > 0$ and $\tau > 0$,

$$SR_{\lambda,\tau}(\chi_{\Omega})(x_0) > \mu_1(\lambda, \tau), \tag{47}$$

where

$$\mu_1(\lambda, \tau) := \frac{(\sqrt{\lambda + \tau} - \sqrt{\tau})^2}{\lambda} \tag{48}$$

is called the standard height for codimension-1 regular boundary points. The analysis of the behavior of $SR_{\lambda,\tau}(\chi_{K_a})$ in the case of the prototype exterior corner defined by the set $K_a = \{(x, y) \in \mathbb{R}^2 : |y| \leq ax, a, x \geq 0\}$, with angle θ satisfying $a = \tan(\theta/2)$, shows that the value of $SR_{\lambda,\tau}(\chi_{K_a})$ at the corner tip $(0, 0)$ of K_a is given by

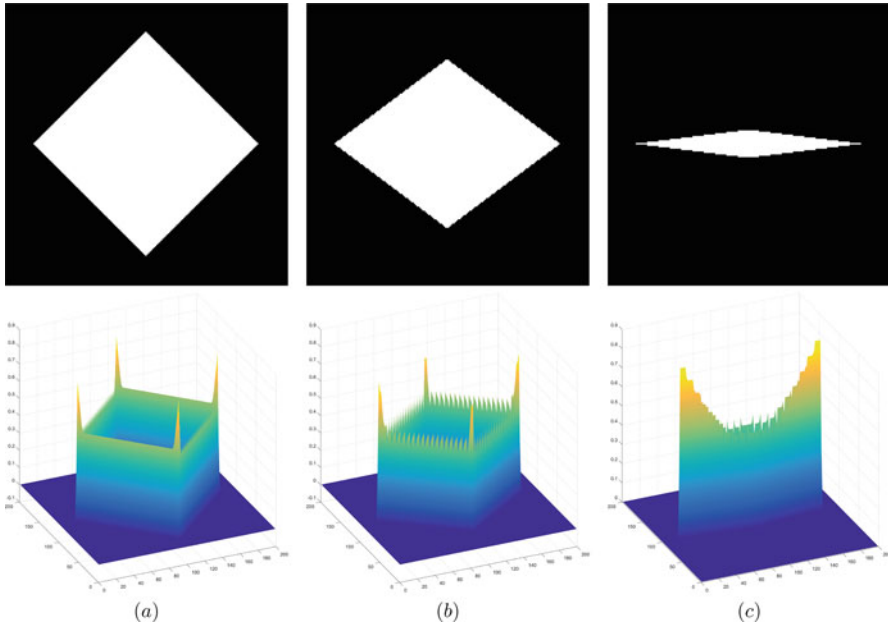


Fig. 13 Graph of $SR_{\lambda, \tau}(\chi_{K_a})$ for different pairs of opening angle θ (a) $\pi/2$ & $\pi/2$; (b) $5\pi/12$ & $7\pi/12$; (c) $\pi/12$ & $11\pi/12$

$$SR_{\lambda, \tau}(\chi_{K_a})(0, 0) := \mu_2(a, \lambda, \tau) = \begin{cases} \frac{\lambda}{\lambda + (1 + a^2)\tau} & \text{if } a^2 \leq \sqrt{\frac{\lambda + \tau}{\tau}} \\ \frac{1 + a^2}{a^2} \frac{(\sqrt{\lambda + \tau} - \sqrt{\tau})^2}{\lambda} & \text{otherwise.} \end{cases} \tag{49}$$

One can then verify that for $a > 0$, and for any $\lambda, \tau > 0$,

$$\mu_2(a, \lambda, \sigma) > \mu_1(\lambda, \tau) \quad \text{and} \quad \lim_{a \rightarrow \infty} \mu_2(a, \lambda, \sigma) = \mu_1(\lambda, \sigma).$$

This result means that when the angle θ approaches π , the singularity at $(0, 0)$ disappears. Figure 13 illustrates the behavior of $SR_{\lambda, \tau}(\chi_{K_a})$ for different values of the opening angle θ and for $\tau = \sigma\lambda$ with $\sigma = 1/8$, for which the value of a that separates the two conditions in (49) corresponds to $\theta = 2\pi/3$.

Based on this prototype example (Zhang et al. 2015b, Example 6.11), one can therefore conclude that $R_\tau(C_\lambda^u(\chi_{\tilde{\Omega}}))$ can actually detect exterior corners, whereas it might happen that at some δ -singular points of $\partial\Omega$, $R_\tau(C_\lambda^u(\chi_{\tilde{\Omega}}))$ takes on values lower than at the regular points of $\partial\Omega$. As a result, a different Hausdorff stable method will be therefore needed to detect interior corners and boundary intersections of domains.

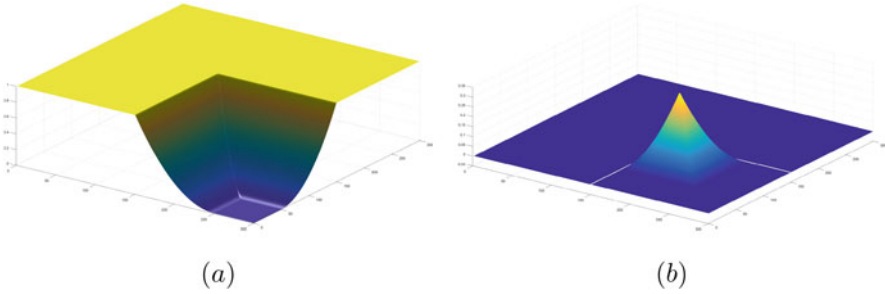


Fig. 14 Prototype of internal corner in an L -shape domain. (a) Graph of $D_\lambda^2(\cdot, K)$ for $\lambda = 0.0001$; (b) graph of $V_\lambda^d(\cdot, K)$

Interior Corners

Since a prototype interior corner is defined as the complement of an exterior corner, one could think of detecting interior corners of Ω by looking at the stable ridge transform of the complement of Ω in \mathbb{R}^n . But this would not provide useful information for geometric objects subject to finite sampling. On the other hand, traditional methods, such as the Harris and the Susan (Smith and Brady 1997) corner detector, as well as other local mask-based corner detection methods, would also not apply directly to such a situation. In this case, therefore we adopt an indirect approach. This consists of constructing an ad hoc geometric design-based function that is robust under sampling and is such that its singularities can be identified with the geometric singularities we want to extract: (i) interior corners of a domain and (ii) intersections of smooth manifolds. By applying one of the transforms introduced in section “Basic Transforms” according to the type of singularity, we can detect the singularity of interest. Given a non-empty closed set $K \subset \mathbb{R}^2$ with $K \neq \mathbb{R}^n$, an instance of function whose singularities capture the type of geometric feature of K which we are interested of is the distance-based function (26) for $\lambda > 0$, which we rewrite next for ease of reference

$$D_\lambda^2(x, K) = \left(\max\{0, 1 - \sqrt{\lambda} \text{dist}(x, K)\} \right)^2, \quad x \in \mathbb{R}^n. \tag{50}$$

Figure 14a displays the graph of $D_\lambda^2(x, K)$ for a prototype of interior corner in an L -shape domain and shows that such singularity is of the valley type. By applying then to $D_\lambda^2(\cdot, K)$ the valley transform (38) with the same parameter λ as used to compute $D_\lambda^2(\cdot, K)$ itself, we obtain

$$\begin{aligned} V_\lambda^d(x, K) &= -V_\lambda(D_\lambda^2(\cdot, K))(x) \\ &= C_\lambda^u(D_\lambda^2(\cdot, K))(x) - D_\lambda^2(x, K), \quad x \in \mathbb{R}^n, \end{aligned} \tag{51}$$

whose graph is displayed in Fig. 14b. We observe therefore that this transform allows the definition of the set of interior corner points and intersection points of

scale $1/\sqrt{\lambda}$ as the support of $V_\lambda^d(\cdot, K)$, that is

$$I_\lambda(K) = \{x \in \mathbb{R}^n, V_\lambda^d(x, K) > 0\}. \quad (52)$$

In this manner, we obtain a marker which is localized in the neighborhood of the feature. Figure 15 displays, for $\lambda = 0.0001$, the behavior of $D_\lambda^2(\cdot, K)$, of $V_\lambda^d(\cdot, K)$, and of the suplevel set of $V_\lambda^d(\cdot, K)$ for a level equal to $0.8 \max_{x \in \mathbb{R}^2} \{V_\lambda^d(x, K)\}$ as approximation of $I_\lambda(K)$, considering different opening angles of the interior corner prototype K . We observe that the marker reduces and the maximum of $V_\lambda^d(x, K)$ depends on the opening angle of the corner. The larger is the angle, the smaller is the value of $\max V_\lambda^d(x, K)$ which agrees with what we expect given that the interior angle disappears and the marker vanishes. Finally, since $D_\lambda^2(\cdot, K)$ is Hausdorff-Lipschitz continuous, it is easy to see that so is $V_\lambda^d(x, K)$.

Stable Multiscale Intersection Transform of Smooth Manifolds

Rather than devising an ad hoc function that embeds the geometric features as its singularities, one can suitably modify the landscape of the characteristic function of the object and generate singularities which are localized in a neighborhood of the geometric feature of interest. This is, for instance, the rationale behind the transformation introduced in Zhang et al. (2015c). The objective is to obtain a Hausdorff stable multiscale method that is robust with respect to sampling, so that it can be applied to geometric objects represented by point clouds, and that is able to describe possible hierarchy of features as defined in terms of some characteristic geometric property. If we denote by $K \subset \mathbb{R}^n$ the union of finitely many smooth compact manifolds M_k , for $k = 1, \dots, m$, in this section we are interested to extract two types of geometric singularities:

- (i) Transversal surface-to-surface intersections.
- (ii) Boundary points shared by two smooth manifolds.

These problems are studied extensively in computer-aided geometric design under the general terminology of shape interrogation (Patrikalakis and Maekawa 2002). The traditional approach to surface-to-surface intersection problems is to consider parameterized polynomial surfaces and to solve systems of algebraic equations numerically based on real algebraic geometry (Patrikalakis and Maekawa 2002). The application of these methods typically requires some topological information such as triangle mesh connectivity or a parameterization of the geometrical objects; hence, they are difficult to implement in the cases of free-form surfaces and of manifolds represented, for instance, by point clouds. For the latter case, other types of approaches are usually used which aim at identifying, according to some criteria, the points that are likely to belong to a neighborhood of the sharp feature. State-of-art methods currently in use are mostly justified by numerical experiments, and

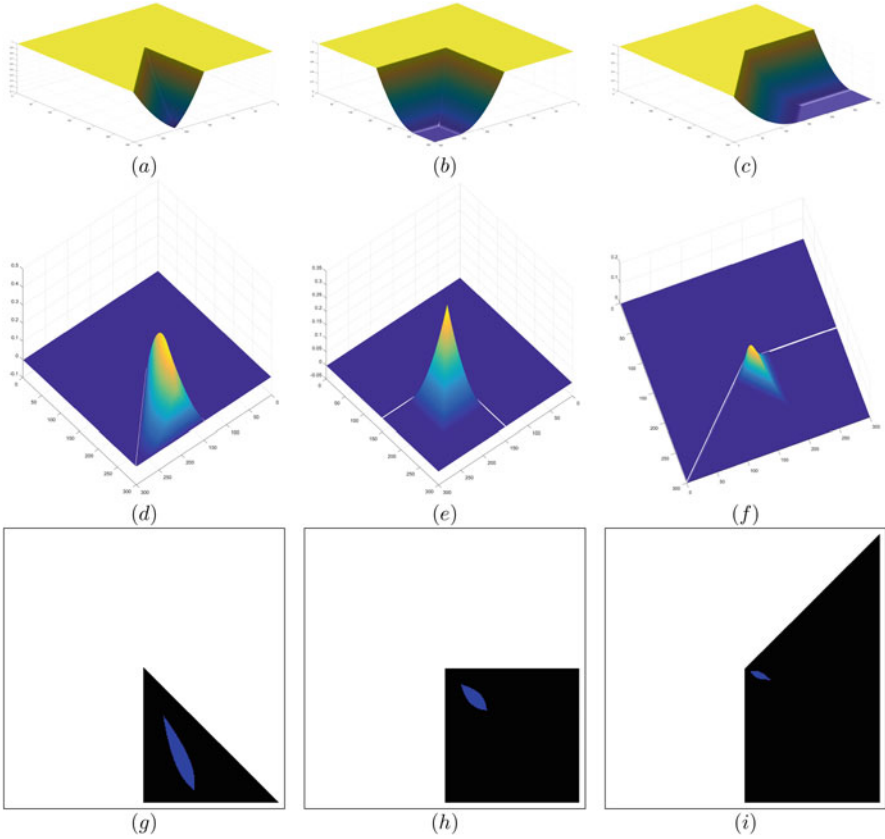


Fig. 15 Graph of $D_\lambda^2(\cdot, K)$, $\lambda = 0.0001$, for the three prototypes of interior angle: (a) acute angle, (b) rectangular angle, and (c) obtuse angle. Graph of $V_\lambda^d(\cdot, K)$, $\lambda = 0.0001$, for the three prototypes of interior angle: (d) acute angle, (e) rectangular angle, and (f) obtuse angle. Suplevel set of $V_\lambda^d(\cdot, K)$, $\lambda = 0.0001$ and level equal to $0.8 \max_{x \in \mathbb{R}^2} \{V_\lambda^d(x, K)\}$ for different values of the opening angle of the interior corner prototype: (g) acute angle, $\max_{x \in \mathbb{R}^2} \{V_\lambda^d(x, K)\} = 0.4137$; (h) rectangular angle, $\max_{x \in \mathbb{R}^2} \{V_\lambda^d(x, K)\} = 0.3323$; (i) obtuse angle, $\max_{x \in \mathbb{R}^2} \{V_\lambda^d(x, K)\} = 0.1053$

their stability properties, under dense sampling of the set M , are not known. Let $K \subset \mathbb{R}^n$ be a non-empty compact set. By using compensated convex transforms, we introduced the intersection extraction transform of scale $\lambda > 0$ (Zhang et al. 2015c) by

$$I_\lambda(x; K) = \left| C_{4\lambda}^u(\chi_K)(x) - 2\left(C_\lambda^u(\chi_K)(x) - C_\lambda^l(C_\lambda^u(\chi_K))(x) \right) \right|, \quad x \in \mathbb{R}^n. \tag{53}$$

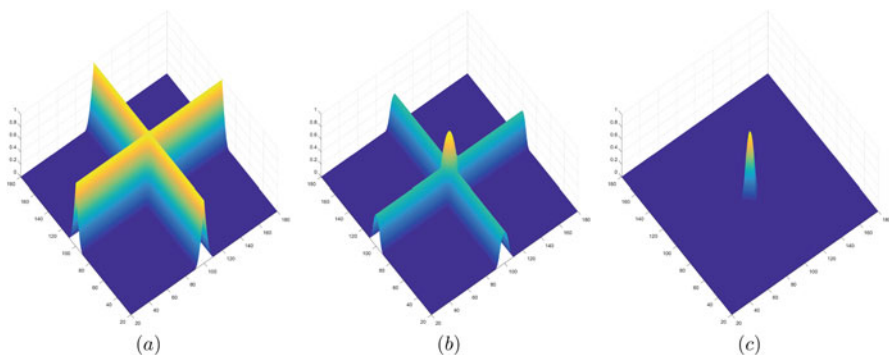


Fig. 16 Graph of: (a) The upper transform $C_\lambda^u(\chi_{K_{\alpha=1}})(x)$ of the characteristic function of two crossing lines with right angle; (b) The mixed transform $C_\lambda^l(C_\lambda^u(\chi_{K_{\alpha=1}}))(x)$; (c) The intersection filter $I_\lambda(\cdot; K_{\alpha=1})$ together with the graph of the characteristic function of $K_{\alpha=1}$ displayed as reference

By recalling the definition of the stable ridge transform (45) of scale λ and τ for the characteristic function χ_K , $I_\lambda(x; K)$ can be expressed in terms of $SR_{\lambda,\tau}(\chi_K)(x)$ for $\tau = \lambda$ as

$$I_\lambda(x; K) = \left| C_{4\lambda}^u(\chi_K)(x) - 2SR_{\lambda,\lambda}(\chi_K)(x) \right|, \quad x \in \mathbb{R}^n. \tag{54}$$

As instance of how $I_\lambda(\cdot; K)$ is used to remove or filter regular points, Fig. 16 illustrates the graphs of $C_\lambda^u(\chi_{K_{\alpha=1}})(x)$, $C_\lambda^l(C_\lambda^u(\chi_{K_{\alpha=1}}))(x)$ and of the filter $I_\lambda(\cdot; K_{\alpha=1})$ in the case of the intersection of two lines perpendicular to each other. This example can be generalized to “regular directions” and “regular points” on manifolds K and verify that $I_\lambda(x, K) = 0$ at these points. Let $K \subset \mathbb{R}^n$ be a non-empty compact set and e a δ -regular direction of $x \in K$, then $I_\lambda(y; K) = 0$ for $y \in [x - \delta e, x + \delta e] := \{x + t\delta e, -1 \leq t \leq 1\}$ when $\lambda \geq 1/\delta^2$. In particular, we have that at the point x ,

$$C_\lambda^l(C_\lambda^u(\chi_K))(x) = 1/2. \tag{55}$$

If K is a C^1 manifold in a neighborhood of $x \in K$ and x is a δ -regular point of K , then $I_\lambda(y; K) = 0$ if $y - x \in N_x$ and $|y - x| \leq \delta$. Since $C_\lambda^u(\chi_K)(x) = 1$ for $x \in K$, by using $I_\lambda(\cdot; K)$, we have that the regular points will be removed by the transform itself, leaving only points near the singular ones. In this context, for compact C^2 m -dimensional manifolds with $1 \leq m \leq n - 1$, since $I_\lambda(y; K) = 0$ for all δ -regular points $y \in K$ when $\lambda > 0$ is sufficiently large, the condition $I_\lambda(y; K) = 0$ can thus be used to define singular points which can be extracted by $I_\lambda(\cdot; K)$ if there exists a constant $c_x > 0$, depending at most only on x , such that $I_\lambda(x; K) \geq c_x > 0$ for sufficiently large $\lambda > 0$.

From the definition (54) of $I_\lambda(\cdot; K)$ in terms of the stable ridge transform and of the upper transform of the characteristic function of the manifold K , since such

transforms are Hausdorff stable, it follows that $I_\lambda(\cdot; K)$ is also Hausdorff stable, that is, for E, F non-empty compact subsets of \mathbb{R}^n and $\lambda > 0$, then there holds

$$|I_\lambda(x; E) - I_\lambda(x; F)| \leq 12\sqrt{\lambda} \operatorname{dist}_{\mathcal{H}^\ell}(E, F), \quad x \in \mathbb{R}^n. \tag{56}$$

Stable Multiscale Medial Axis Map

The medial axis of an object is a geometric structure introduced by Blum (1967) as a means of providing a compact representation of a shape which was initially defined as the set of the shock points of a grass fire lit on the boundary that propagate uniformly inside the object. Closely related definitions of skeleton and cut-locus (Siddiqi and Pizer 2008) have since been proposed and have served for the study of its topological properties (Albano 2014; Albano et al. 2013; Lieutier 2004; Matheron 1988) and its stability (Chazal and Soufflet 2004) and for the development of fast and efficient algorithms for its computation (Aichholzer et al. 2009; Attali and Montanvert 1997; Kimmel et al. 1995). Hereafter we refer to the definition given in Lieutier (2004). For a given non-empty closed set $K \subset \mathbb{R}^n$, with $K \neq \mathbb{R}^n$, we define the medial axis M_K of K as the set of points $x \in \mathbb{R}^n \setminus K$ such that $x \in M_K$ if and only if there are at least two different points $y_1, y_2 \in K$, satisfying $\operatorname{dist}(x, K) = |x - y_1| = |x - y_2|$, whereas for a non-empty bounded open set $\Omega \subset \mathbb{R}^n$, the medial axis of Ω is defined by $M_\Omega := \Omega \cap M_{\partial\Omega}$.

The application of the lower transform to study the medial axis M_K of a set K is motivated by the identification of the medial axis with the singularity set of the Euclidean distance function and by the geometric structure of this set (Albano et al. 2013). However, for our setting, it is more convenient to consider the squared-distance function and to use the identification of the singular set of the squared-distance function with the set of points where the squared-distance function fails to be locally $C^{1,1}$. Since the lower compensated convex transform to the Euclidean squared-distance function gives a smooth ($C^{1,1}$) tight approximation outside a neighborhood of the closure of the medial axis, in Zhang et al. (2015a) the quadratic multiscale medial axis map with scale $\lambda > 0$ is defined as a scaled difference between the squared-distance function and its lower transform, that is,

$$\begin{aligned} M_\lambda(x; K) &:= (1 + \lambda)R_\lambda(\operatorname{dist}^2(\cdot, K))(x) \\ &= (1 + \lambda)\left(\operatorname{dist}^2(x, K) - C_\lambda^l(\operatorname{dist}^2(\cdot, K))(x)\right), \end{aligned} \tag{57}$$

whereas for a bounded open set $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega$, the quadratic multiscale medial axis map of Ω with scale $\lambda > 0$ is defined by

$$M_\lambda(x; \Omega) := M_\lambda(x; \partial\Omega) \quad x \in \Omega.$$

A direct consequence of the definition of $M_\lambda(x; K)$ is that for $x \in \mathbb{R}^n \setminus M_K$ we have

$$\lim_{\lambda \rightarrow \infty} M_\lambda(x; K) = 0, \quad (58)$$

and the limit map $M_\infty(x; K)$ presents well separated values, in the sense that they are zero outside the medial axis and remain strictly positive on it. To gain an insight of the geometric structure of $M_\lambda(x; K)$, for $x \in M_K$, Zhang et al. (2015a) makes use of the separation angle θ_x introduced in Lieutier (2004). Let $K(x)$ denote the set of points of ∂K that realize the distance of x to K and by $\angle[y_1 - x, y_2 - x]$ the angle between the two nonzero vectors $y_1 - x$ and $y_2 - x$ for $y_1, y_2 \in K(x)$, then

$$\theta_x = \max\{\angle[y_1 - x, y_2 - x], \quad y_1, y_2 \in K(x)\}. \quad (59)$$

By means of this geometric parameter, it was shown in Zhang et al. (2015a) that for every $\lambda > 0$ and $x \in M_K$ that

$$\sin^2(\theta_x/2) \operatorname{dist}^2(x, K) \leq M_\lambda(x; K) \leq \operatorname{dist}^2(x, K). \quad (60)$$

This result along with the examination of prototype examples ensures that the multiscale medial axis map of scale λ keeps a constant height along the part of the medial axis generated by a two-point subset, with the value of the height depending on the distance between the two generating points. Such values can, therefore, be used to define a hierarchy between different parts of the medial axis, and one can thus select the relevant parts through simple thresholding, that is, by taking suplevel sets of the multiscale medial axis map, justifying the word ‘‘multiscale’’ in its definition. For each branch of the medial axis, the multiscale medial axis map automatically defines a scale associated with it. In other words, a given branch has a strength which depends on some geometric features of the part of the set that generates that branch.

An inherent drawback of the medial axis M_K is in fact its sensitivity to boundary details, in the sense that small perturbations of the object (with respect to the Hausdorff distance) can produce huge variations of the corresponding medial axis. This does not occur in the case of the quadratic multiscale medial axis map, given that Zhang et al. (2015a) shifts the focus from the support of $M_\lambda(\cdot; K)$ to the whole map. Let $K, L \subset \mathbb{R}^n$ denote non-empty compact sets and $\mu := \operatorname{dist}_{\mathcal{H}}(K, L)$, it was shown in Zhang et al. (2015a) that for $x \in \mathbb{R}^n$, we have

$$\left| M_\lambda(x; K) - M_\lambda(x; L) \right| \leq \mu(1 + \lambda) \left((\operatorname{dist}(x, K) + \mu)^2 + 2\operatorname{dist}(x, K) + 2\mu + 1 \right). \quad (61)$$

While the medial axis of K is not a stable structure with respect to the Hausdorff distance, its medial axis map $M_\lambda(x; K)$ is by contrast a stable structure. This result complies with (61) which shows that as λ becomes large, the bound in (61) becomes large.

With the aim of giving insights into the implications of the Hausdorff stability of $M_\lambda(x; \partial\Omega)$, we display in Fig. 17 the graph of the multiscale medial axis map of a nonconvex domain Ω and of an ϵ -sample K_ϵ of its boundary. An inspection of

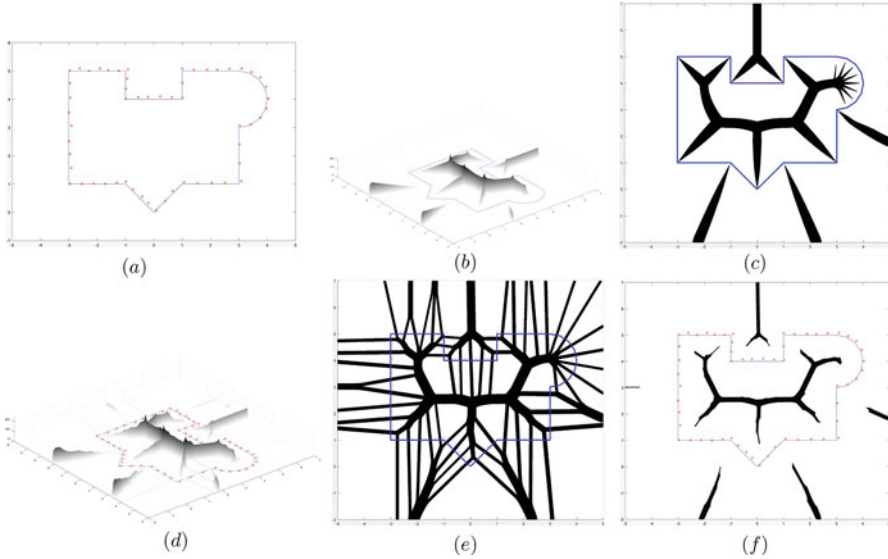


Fig. 17 Multiscale medial axis map of a nonconvex domain Ω and of an ϵ -sample K_ϵ of its boundary. (a) Nonconvex domain Ω (—) and an ϵ -sample K_ϵ (\times) of $\partial\Omega$; (b) graph of $M_\lambda(\cdot; \partial\Omega)$ for $\lambda = 2.5$; (c) support of $M_\lambda(\cdot; \partial\Omega)$; (d) graph of $M_\lambda(\cdot; K_\epsilon)$; (e) support of $M_\lambda(\cdot; \Omega)$; (f) suplevel set of $M_\lambda(x; K_\epsilon)$ for a threshold equal to $0.15 \max_{x \in \mathbb{R}^2} \{M_\lambda(x; K_\epsilon)\}$

the graph of $M_\lambda(x; \partial\Omega)$ and $M_\lambda(x; K_\epsilon)$, displayed in Fig. 17b and d, respectively, reveals that both functions take comparable values along the main branches of M_Ω . Also, $M_\lambda(x; K_\epsilon)$ takes small values along the secondary branches, generated by the sampling of the boundary of Ω . These values can therefore be filtered out by a simple thresholding so that a stable approximation of the medial axis of Ω can be computed. This can be appreciated by looking at Fig. 17f, which displays a suplevel set of $M_\lambda(x; K_\epsilon)$ that appears to be a reasonable approximation of the support of $M_\lambda(x; \partial\Omega)$ shown in Fig. 17c whereas Fig. 17e depicts the support of $M_\lambda(\cdot; K_\epsilon)$.

A relevant implication of (61) concerns with the continuous approximation of the medial axis of a shape starting from subsets of the Voronoi diagram of a sample of the shape boundary which is pertinent for shape reconstruction from point clouds. Let us consider an ϵ -sample K_ϵ of $\partial\Omega$, that is, a discrete set of points such that $\text{dist}_{\mathcal{H}}(\partial\Omega, K_\epsilon) \leq \epsilon$. Since the medial axis of K_ϵ is the Voronoi diagram of K_ϵ , if we denote by V_ϵ the set of all the vertices of the Voronoi diagram $\mathcal{V}or(K_\epsilon)$ of K_ϵ and denote by P_ϵ the subset of V_ϵ formed by the “poles” of $\mathcal{V}or(K_\epsilon)$ introduced in Amenta and Bern (1999) (i.e., those vertices of $\mathcal{V}or(K_\epsilon)$ that converge to the medial axis of Ω as the sample density approaches infinity), then, for $\lambda > 0$, it was established in Zhang et al. (2015a) that

$$\lim_{\epsilon \rightarrow 0^+} M_\lambda(x_\epsilon; K_\epsilon) = 0 \quad \text{for } x_\epsilon \in V_\epsilon \setminus P_\epsilon.$$

Since as $\epsilon \rightarrow 0+$, $K_\epsilon \rightarrow \partial\Omega$, and knowing that $P_\epsilon \rightarrow M_\Omega$ (Amenta et al. 2001), then on the vertices of $Vor(K_\epsilon)$ that do not tend to M_Ω , $M_\lambda(x_\epsilon; K_\epsilon)$ must approach zero in the limit because of (58). As a result, in the context of the methods of approximating the medial axis starting from the Voronoi diagram of a sample set (such as those described in Amenta et al. 2001, Dey 2006, and Siddiqi and Pizer 2008), the use of the multiscale medial axis map offers an alternative and much easier tool to construct continuous approximations to the medial axis with guaranteed convergence as $\epsilon \rightarrow 0+$.

We conclude this topic by showing how compensated convex transform is used to obtain a fine result of geometric measure theory. Let us introduce the set $V_{\lambda,K}$ defined as

$$V_{\lambda,K} = \{x \in \mathbb{R}^n : \lambda \text{dist}(x, M_K) \leq \text{dist}(x, K)\}, \quad (62)$$

which represents a neighborhood of \overline{M}_K . From the property of the tight approximation of the lower transform of the squared-distance function, it was shown in Zhang et al. (2015a) that

$$\text{dist}^2(\cdot, K) \in C^{1,1}(\mathbb{R}^n \setminus V_{\lambda,K}), \quad (63)$$

and a sharp estimate for the Lipschitz constant of $D\text{dist}^2(\cdot, K)$ was also obtained. This result can be viewed as a weak Lusin-type theorem for the squared-distance function which extends regularity results of the squared-distance function to any closed non-empty subset of \mathbb{R}^n .

Approximation Transform

The theory of compensated convex transforms can also be applied to define Lipschitz continuous and smooth geometric approximations and interpolations for bounded real-valued functions sampled from either a compact set K in \mathbb{R}^n or the complement of a bounded open set Ω , i.e., $K = \mathbb{R}^n \setminus \Omega$. The former is motivated by approximating or interpolating sparse data and/or contour lines whereas the latter by the so-called inpainting problem in image processing (Chan and Shen 2005), where some parts of the image content are missing. The aim of ‘‘inpainting’’ is to use other information from parts of the image to repair or reconstruct the missing parts.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ denote the underlying function to be approximated, $f_K : K \subset \mathbb{R}^n \rightarrow \mathbb{R}$ the sampled function defined by $f_K(x) = f(x)$ for $x \in K$, and $\Gamma_{f_K} := \{(x, f_K(x)), x \in K\}$ its graph, the setting for the application of the compensated convex transforms to obtain an approximation transform is the following. Given $M > 0$, we define first two functions extending f_K to $\mathbb{R}^n \setminus K$, namely,

$$\begin{aligned}
 f_K^{-M}(x) &= f(x)\chi_K(x) - M\chi_{\mathbb{R}^n \setminus K} = \begin{cases} f_K(x), & x \in K, \\ -M, & x \in \mathbb{R}^n \setminus K; \end{cases} \\
 f_K^M(x) &= f(x)\chi_K(x) + M\chi_{\mathbb{R}^n \setminus K} = \begin{cases} f_K(x), & x \in K, \\ M, & x \in \mathbb{R}^n \setminus K, \end{cases}
 \end{aligned}
 \tag{64}$$

where χ_G denotes the characteristic function of a set G . We then compute the arithmetic average of the proximal hull of $f_K^M(x)$ and the upper proximal hull of f_K^{-M} as follows:

$$A_\lambda^M(f_K)(x) = \frac{1}{2} \left(C_\lambda^l(f_K^M)(x) + C_\lambda^u(f_K^{-M})(x) \right), \quad x \in \mathbb{R}^n, \tag{65}$$

which we refer to as the average compensated convex approximation transform of f_K of scale λ and level M (Zhang et al. 2016a).

In the case that $K \subset \mathbb{R}^n$ is a compact set and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is bounded and uniformly continuous, error estimates are available for $M \rightarrow \infty$ and for $x \in \text{co}[K]$. If for $x \in \text{co}[K] \setminus K$ we denote by $r_c(x)$ the convex density radius as the smallest radius of a closed ball $\bar{B}(x; r_c(x))$ such that x is in the convex hull of $K \cap \bar{B}(x; r_c(x))$, then for $\lambda > 0$ and all $x \in \text{co}[K]$ there holds

$$|A_\lambda^\infty(f_K)(x) - f(x)| \leq \omega \left(r_c(x) + \frac{a}{\lambda} + \sqrt{\frac{2b}{\lambda}} \right), \tag{66}$$

where $\omega = \omega(t)$ is the least concave majorant of the modulus of continuity ω_f of f and $a \geq 0, b \geq 0$ are such that $\omega(t) \leq at + b$ for $t \geq 0$. Error estimates are also available for a finite $M > 0$ under the extra restriction that $f(x) = c_0$ for $|x| \geq r$ where $c_0 \in \mathbb{R}$ and $r > 0$ are constants. In this case, for $R > r$, we extend f_K to be equal to c_0 outside a large ball $B(0; R)$ containing K and define $K_R = K \cup B^c(0; R)$. Thus we obtain similar error estimate to (66) for $A_\lambda^M(f_{K_R})(x)$. Furthermore, we have that when $M > 0$ is sufficiently large, $A_\lambda^M(f_K)$ approaches f_K in K as $\lambda \rightarrow \infty$, whereas if f is a $C^{1,1}$ function and $\lambda > 0$ is large enough, $A_\lambda^M(f_K)$ is an interpolation of f in the convex hull $\text{co}[K]$ of K . In the special case of a finite set K , the average approximation $A_\lambda^M(f_K)$ defines an approximation for the scattered data $\Gamma_{f_K} = \{(x, f_K(x)), x \in K\}$.

If the closed set K is the complement of a non-empty bounded open set $\Omega \subset \mathbb{R}^n$, we can also obtain estimates that are similar to (66). Clearly, $\text{co}[K] = \mathbb{R}^n$ for such a K , thus if $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is bounded and uniformly continuous, satisfying $|f(x)| \leq A_0$ for some constant $A_0 > 0$ and for all $x \in \mathbb{R}^n$, and d_Ω denotes the diameter of Ω , then for $\lambda > 0, M > A_0 + \lambda d_\Omega^2$ and all $x \in \mathbb{R}^n$, we have

$$|A_\lambda^M(f_K)(x) - f(x)| \leq \omega \left(r_c(x) + \frac{a}{\lambda} + \sqrt{\frac{2b}{\lambda}} \right), \tag{67}$$

where, as for (66), the constants $a \geq 0$ and $b \geq 0$ are such that $\omega(t) \leq at + b$ for $t \geq 0$ with $\omega = \omega(t)$ the least concave majorant of the modulus of continuity ω_f of f .

Both the estimates (66) and (67) can be improved for Lipschitz functions and for $C^{1,1}$ functions.

Another natural and practical question in data approximation and interpolation is the stability of a given method. For approximations and interpolations of sampled functions, we would like to know, for two sample sets which are “close” to each other under the Hausdorff distance (Ambrosio and Tilli 2004), for instance, whether the corresponding approximations are also close to each other. It is easy to see that differentiation- and integration-based approximation methods are not Hausdorff stable because continuous functions can be sampled over a finite dense set. One of the advantages of the compensated convex approximation is that for a bounded uniformly continuous function f , and for fixed $M > 0$ and $\lambda > 0$, the mapping $K \rightarrow A_\lambda^M(f_K)$ is continuous with respect to the Hausdorff distance for compact sets K , and the continuity is uniform with respect to $x \in \mathbb{R}^n$. This means that if another sampled subset $E \subset \mathbb{R}^n$ (finite or compact) is close to K , then the output $A_\lambda^M(f_E)(x)$ is close to $A_\lambda^M(f_K)(x)$ uniformly with respect to $x \in \mathbb{R}^n$. As far as we know, not many known interpolation/approximation methods share such a property.

By using the mixed compensated convex transforms (Zhang 2008a), it is possible to define a mixed average compensated convex approximation with scales $\lambda > 0$ and $\tau > 0$ for the sampled function $f_K : K \rightarrow \mathbb{R}$ by

$$(SA)_{\tau,\lambda}^M(f_K)(x) = \frac{1}{2}(C_\tau^u(C_\lambda^l(f_K^M))(x) + C_\tau^l(C_\lambda^u(f_K^{-M}))(x)), \quad x \in \mathbb{R}^n. \quad (68)$$

Since the mixed compensated convex transforms are $C^{1,1}$ functions (Zhang 2008a, Theorem 2.1(iv) and Theorem 4.1(ii)), the mixed average approximation $(SA)_{\tau,\lambda}^M$ is a smooth version of our average approximation. Also, for a bounded function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, satisfying $|f(x)| \leq M, x \in \mathbb{R}^n$ for some constant $M > 0$, we have the following estimates (Zhang et al. 2015b, Theorem 3.13):

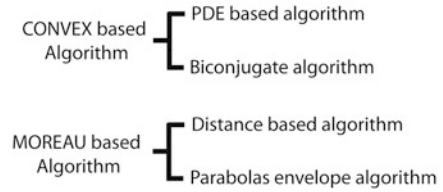
$$0 \leq C_\tau^u(C_\lambda^l(f))(x) - C_\lambda^l(f)(x) \leq \frac{16M\lambda}{\tau}, \quad 0 \leq C_\tau^u(f)(x) - C_\tau^l(C_\lambda^u(f))(x) \leq \frac{16M\lambda}{\tau}$$

for all $x \in \mathbb{R}^n, \lambda > 0$, and $\tau > 0$ and hence we can easily show that for any closed set $K \subset \mathbb{R}^n$,

$$|(SA)_{\tau,\lambda}^M(f_K)(x) - A_\lambda^M(f_K)(x)| \leq \frac{16M\lambda}{\tau}, \quad x \in \mathbb{R}^n.$$

This implies that for given $\lambda > 0$ and $M > 0$, the mixed approximation $(SA)_{\tau,\lambda}^M(f_K)$ converges to the basic average approximation $A_\lambda^M(f_K)$ uniformly in \mathbb{R}^n as $\tau \rightarrow \infty$, with rate of convergence $16M\lambda/\tau$.

Fig. 18 Different approaches for computing the lower compensated convex transform $C_\lambda^l(f)$



Numerical Algorithms

The numerical realization of the convex transforms introduced in section “**Compensated Convexity-Based Transforms**” relies on the availability of numerical schemes for computing the upper and lower transforms of a given function. Because of the relation (4) between the upper and lower transform, the computation of the above transforms ultimately boils down to the evaluation of the lower compensated convex transform. As a result, without loss of generality, in the following, we refer just to the actual implementation of $C_\lambda^l(f)$. With this respect, we can proceed in two different ways according to whether we use definition (2) in terms of the convex envelope or the characterization (5) as proximity hull of the function and use its definition in terms of the Moreau envelopes. In the following, we describe some algorithms that can be used successfully for the computation of $C_\lambda^l(f)$ and discuss their relative merits. Figure 18 summarizes the different approaches considered in this paper.

Convex-Based Algorithms

Algorithms to compute convex hull such as the ones given in Barber et al. (1996) are more suitable for discrete set of points, and their complexity is related to the cardinality of the set. An adaptation of these methods to our case, with the set to convexify given by the epigraph of $f + \lambda|\cdot|^2$, does not appear to be very effective, especially for functions defined in subsets of \mathbb{R}^n for $n \geq 2$, compared to the methods that (directly) compute the convex envelope of a function (Vese 1999; Oberman 2008; Contento et al. 2015).

PDE-Based Algorithm

Of particular interest for applications to image processing, where functions involved are defined on grid of pixels, is the characterization of the convex envelope as the viscosity solution of a nonlinear obstacle problem (Oberman 2008). An approximated solution is then obtained by using centered finite differences along directions defined by an associated stencil to approximate the first eigenvalue of the Hessian matrix at the grid point. A generalization of the scheme introduced in Oberman (2008) in terms of the number of convex combinations of the function values at the grid points of the stencil is briefly summarized in Algorithm 1 and described below. Given a uniform grid of points $x_k \in \mathbb{R}^n$, equally spaced with grid size h , let us denote by S_{x_k} the d -point stencil of \mathbb{R}^n with center at x_k . The stencil S_{x_k} is defined as $S_{x_k} = \{x_k + hr, |r|_\infty \leq 1, r \in \mathbb{Z}^n\}$ where $|r|_\infty$ is the ℓ^∞ -norm of

$r \in \mathbb{Z}^n$ and $d = \#(S)$ is the cardinality of the finite set S . At each grid point x_k , we compute an approximation of the convex envelope of f at x_k by an iterative scheme where each iteration step m is given by

$$(\text{co } f)_m(x_k) = \min \left\{ f(x_k), \sum \lambda_i (\text{co } f)_{m-1}(x_i) : \sum \lambda_i = 1, \lambda_i \geq 0, x_i \in S_{x_k} \right\}$$

with the minimum taken between $f(x_k)$ and only some convex combinations of $(\text{co } f)_{m-1}$ at the stencil grid points x_i of S_{x_k} . It is then not difficult to show that the scheme is monotone, thus convergent. However, there is no estimate of the rate of convergence which, in actual applications, appears to be quite slow. Furthermore, results are biased by the type of underlying stencil.

Algorithm 1 Computation of the convex envelope of f according to Oberman (2008)

- 1: Set $m = 1, (\text{co } f)_0 = f, \text{tol}$
 - 2: $\epsilon = \|f\|_{L^2}$
 - 3: **while** $\epsilon > \text{tol}$ **do**
 - 4: $\forall x_k, (\text{co } f)_m(x_k) = \min \left\{ f(x_k), \sum \lambda_i (\text{co } f)_{m-1}(x_i) : \sum \lambda_i = 1, \lambda_i \geq 0, x_i \in S_{x_k} \right\}$
 - 5: $\epsilon = \|(\text{co } f)_m - (\text{co } f)_{m-1}\|_{L^2}$
 - 6: $m \leftarrow m + 1$
 - 7: **end while**
-

Biconjugate Algorithm

Based on the characterization of the convex envelope of f in terms of the biconjugate $(f^*)^*$ of f (Hiriart-Urruty and Lemaréchal 2001; Rockafellar 1970), where f^* is the Legendre-Fenchel transform of f , we can approximate the convex envelope by computing twice the discrete Legendre-Fenchel transform. We can thus improve speed efficiency with respect to a brute force algorithm, which computes $(f^*)^*$ with complexity $O(N^2)$ with N the number of grid points, if we have an efficient scheme to compute the discrete Legendre-Fenchel transform of a function. For functions $f : X \rightarrow \mathbb{R}$ defined on Cartesian sets of the type $X = \prod_{i=1}^n X_i$ with X_i intervals of $\mathbb{R}, i = 1, \dots, n$, the Legendre-Fenchel transform of f can be reduced to the iterate evaluation of the Legendre-Fenchel transform of functions dependent only on one variable as follows:

$$\begin{aligned}
 (\xi_1, \xi_2, \dots, \xi_n) \in \mathbb{R}^n &\rightarrow f^*(\xi_1, \dots, \xi_n) = \sup_{x \in X} \{ \xi \cdot x - f(x) \} \\
 &= \sup_{x_1, \dots, x_{n-1} \in \prod_{i=1}^{n-1} X_i} \left\{ x_1 \xi_1 + \dots + x_{n-1} \xi_{n-1} \right. \\
 &\quad \left. - \sup_{x_n \in X_n} \{ x_n \xi_n - f(x_1, \dots, x_{n-1}, x_n) \} \right\}. \tag{69}
 \end{aligned}$$

As a result, one can improve the complexity of the computation of f^* if one has an efficient scheme to compute the Legendre-Fenchel transform of functions of only one variable. For instance, the algorithm described in Lucet (1997) and Helluy and Mathis (2011), which exploits an idea of Brenier (1989) and improves the implementation of Corrias (1996), computes the discrete Legendre-Fenchel transform in linear time, that is, with complexity $O(N)$. If g_h denote the grid values of a function of one variable, the key idea of Brenier (1989) and Corrias (1996) is to compute $(g_h)^*$ as approximation of g^* using the following result:

$$(g_h)^*(\xi) = (\text{co}[\Pi f_h])^*(\xi), \quad \xi \in \mathbb{R} \tag{70}$$

where Πg_h denotes the continuous piecewise affine interpolation of the grid values g_h . Therefore, applying an algorithm with linear complexity, for instance, the beneath-beyond algorithm (Preparata and Shamos 1985), to compute the convex envelope $\text{co}[\Pi g_h]$, followed by the use of analytical expressions for the Legendre-Fenchel transform of a convex piecewise affine function yields an efficient method to compute $(g_h)^*$ (Lucet 1997). For functions defined in a bounded domain, in Lucet (1997), it was recommended to increment the size of the domain for a better precision of the computation of the Legendre-Fenchel transform. The work Helluy and Mathis (2011) avoids this by elaborating the exact expression of the Legendre-Fenchel transform of a convex piecewise affine function defined in a bounded domain X which is equal to infinity in $\mathbb{R} \setminus X$, or it has therein an affine variation. In this manner, they can avoid boundary effects. For ease of reference, we report next the analytical expression of g^* in the case where $g : \mathbb{R} \rightarrow \overline{\mathbb{R}}$ is convex piecewise affine. Without loss of generality, let $x_1 < \dots < x_N$ be a grid of points of \mathbb{R} , $c_1 < \dots < c_N$, and assume $g : \mathbb{R} \rightarrow \overline{\mathbb{R}}$ to be defined as follows:

$$g : x \in \mathbb{R} \rightarrow \begin{cases} +\infty & \text{if } x \leq x_1 \\ g_i + c_i(x_i - x) & \text{if } x_i \leq x \leq x_{i+1}, \quad i = 1, \dots, N - 1 \\ g_N + c_N(x_N - x) & \text{if } x \geq x_N \end{cases} \tag{71}$$

where $g_i = g(x_i)$ and c_i , for $i = 1, \dots, N$, represents the slope of each affine piece of g . It is not difficult to verify that the analytical expression of g^* is given by (Helluy and Mathis 2011)

$$g^* : \xi \in \mathbb{R} \rightarrow \begin{cases} x_1 \xi - g_1 & \text{if } \xi \leq c_1 \\ x_{i+1} \xi - g_{i+1} & \text{if } c_i \leq \xi \leq c_{i+1}, \quad i = 1, \dots, N - 2 \\ +\infty & \text{if } \xi \geq c_N. \end{cases} \tag{72}$$

Once we know g^* , using the decomposition (69), we can compute f^* and thus the biconjugate f^{**} .

Moreau Envelope-Based Algorithms

The computation of the Moreau envelope is an established task in the field of computational convex analysis (Lucet 2006) that has been tackled by various different approaches aimed at reducing the quadratic complexity of a direct brute force implementation of the transform. Such reduction is achieved, one way or another, by a dimensional reduction.

Distance-Based Algorithm

The fundamental idea of the scheme presented in Zhang et al. (2021) is the generalization of the Euclidean distance transform of binary images, by replacing the binary image by an arbitrary function on a grid. The decomposition of the structuring element which yields the exact Euclidean distance transform (Shih and Mitchell 1992) into basic ones leads to a simple and fast algorithm where the discrete lower Moreau envelope can be computed by a sequence of local operations, using one-dimensional neighborhoods. Unless otherwise stated, in the following, $i, j, k, r, s, p, q \in \mathbb{Z}$ denote integer numbers, whereas $m, n \in \mathbb{N}$ are nonnegative integers. Given $n \geq 1$, we introduce a grid of points of the space \mathbb{R}^n with regular spacing $h > 0$ denoted by $x_k \in \mathbb{R}^n$, $k \in \mathbb{Z}$ and define the discrete lower Moreau envelope at $x_k \in \mathbb{R}^n$ as

$$M_\lambda^h(f)(x_k) = \inf\{f(x_k + rh) + \lambda h^2|r|^2, r \in \mathbb{Z}^n\}. \quad (73)$$

By taking the infimum in (73) over a finite number $m \geq 1$ of directions, we obtain the m -th approximation of the discrete Moreau lower envelope $M_\lambda^h(f)(x_k)$ which can be evaluated by taking the values $f_m(x_k)$ given by Algorithm 2. For the convergence analysis and convergence rate, we refer to Zhang et al. (2021) where it is shown that the scheme has a linear convergence rate with respect to h .

Algorithm 2 Computation of $f_m(x_k)$ at the points x_k of the grid of \mathbb{R}^n of size h for given $m \geq 1$

```

1: Set  $i = 1, m \in \mathbb{N}$ 
2:  $\forall x_k, f_0(x_k) = f(x_k)$ 
3: while  $i < m$  do
4:    $\tau_i = 2i - 1$ 
5:    $f_i(x_k) = \min\{f_{i-1}(x_k + rh) + \lambda h^2|r|^2\tau_i : r \in \mathbb{Z}^n, |r|_\infty \leq 1\}$ 
6:    $i \leftarrow i + 1$ 
7: end while

```

Parabola Envelope-Based Algorithm

Similar to the computation of the Legendre-Fenchel transform, in the scheme proposed by Felzenszwalb and Huttenlocher (2012), the authors apply the dimensional reduction directly to the computation of the Moreau envelope which is factored by n one-dimensional Moreau envelope. For instance, in the case of $n = 2$, let $\Omega = X \times Y$, with $X, Y \subset \mathbb{R}$, and $(\xi_1, \xi_2) \in \Omega = X \times Y$, for any $x = (x_1, x_2) \in \mathbb{R}^2$, we have

$$\begin{aligned}
 M_\lambda(f)(x_1, x_2) &= \inf_{(\xi_1, \xi_2) \in \Omega} \{ \lambda |(x_1, x_2) - (\xi_1, \xi_2)|^2 + f(\xi_1, \xi_2) \} \\
 &= \inf_{\xi_1 \in X} \left\{ \lambda |x_1 - \xi_1|^2 + \inf_{\xi_2 \in Y} \{ \lambda |x_2 - \xi_2|^2 + f(\xi_1, \xi_2) \} \right\}. \tag{74}
 \end{aligned}$$

For the computation of $M_\lambda(f)$ with f function of one variable, if we denote by \mathcal{F} the family of parabolas with given curvature λ of the following type

$$\mathbf{p}_q : x \in \mathbb{R} \rightarrow \mathbf{p}_q(x) = \lambda |x - q|^2 + f(q),$$

parameterized by $q \in \Omega \subset \mathbb{R}$, we have that

$$M_\lambda(f)(x) = \inf_{\mathbf{p}_q \in \mathcal{F}} \{ \mathbf{p}_q(x) : \mathbf{p}_q(y) \leq f(y) \text{ for any } y \in \mathbb{R}^n \} \tag{75}$$

that is, the Moreau envelope of a function of one variable is reduced to the computation of the lower envelope of parabolas of given curvature λ . The computation of such envelope is realized by Felzenszwalb and Huttenlocher (2012) in two steps. In the first one, they compute the envelope by adding the parabolas one at time which is done in linear time and comparing each parabola to the parabolas that realize the envelope, which is done in constant time, whereas in the second step, they compute the value of the envelope at the given point $x \in \mathbb{R}$. The key points of the scheme result from two observations. The first one is that given any two parabolas of \mathcal{F} parameterized by $q, r \in \Omega$, their intersection occurs only at one point with coordinate

$$x_s = \frac{(f(q) - f(r)) + \lambda(q^2 - r^2)}{2\lambda(q - r)},$$

whereas the second one regards the relation between the parabolas so that if $q < r$, then $\mathbf{p}_q(x) \leq \mathbf{p}_r(x)$ for $x < x_s$ and $\mathbf{p}_q(x) \geq \mathbf{p}_r(x)$ for $x > x_s$. This scheme allows the evaluation of $M_\lambda(f)(x)$ for any $x \in \mathbb{R}^n$ even if f is defined only on a bounded open set Ω , without any consideration on how to extend f on $\mathbb{R}^n \setminus \Omega$.

The Moreau Transform as Legendre–Fenchel Transform

By using the link between the Moreau envelope and the Legendre-Fenchel transform given by Rockafellar and Wets (1998) and Lucet (2006)

$$M_\lambda(f)(x) = \lambda |x|^2 - 2\lambda \left(\frac{f}{2\lambda} + \frac{|\cdot|^2}{2} \right)^* (x), \tag{76}$$

it is possible to design another scheme to calculate the Moreau envelope by computing the Legendre-Fenchel transform of the augmented function that appears in (76) (Lucet 2006). In this case, however, special considerations must be taken

about the primary domain, where the Moreau envelope is defined, and the dual domain, which is the one where the Legendre-Fenchel transform is defined.

Numerical Examples

In this section, we present some illustrative numerical examples of implementation of the transforms introduced in section “[Compensated Convexity-Based Transforms](#)”. We precede this discussion by the computation of a two-dimensional prototype example with analytical expression of $C_\lambda^u(\chi_K)$ which we use to select the most suitable numerical scheme out of those described in section “[Numerical Algorithms](#)” for the computation of the compensated convex transforms.

Prototype Example: Upper Transform of a Singleton Set of \mathbb{R}^2

Given the singleton set $K = \{0\} \subset \mathbb{R}^2$, the analytical expression of $C_\lambda^u(\chi_K)$ established in Zhang et al. (2015c, Example 1.2) is given by

$$C_\lambda^u(\chi_K)(x) = \begin{cases} 0, & \text{if } |x| > 1/\sqrt{\lambda}, \\ \lambda(1/\sqrt{\lambda} - |x|)^2, & \text{if } |x| \leq 1/\sqrt{\lambda}. \end{cases} \quad (77)$$

We compute then $C_\lambda^u(\chi_K)$ by applying the convex-based algorithms, i.e., Algorithm 1 (Oberman 2008) and the biconjugate-based scheme (shorted as *BS* hereafter) (Lucet 1997; Helluy and Mathis 2011), and the Moreau-based algorithms, i.e., Algorithm 2 and the parabola envelope scheme (shorted as *PES* hereafter) (Felzenszwalb and Huttenlocher 2012). To compare the accuracy of the schemes, we will consider (i) the Hausdorff distance between the support of the exact and the computed upper transform,

$$e_{\mathcal{H}} = \text{dist}_{\mathcal{H}} \left(\overline{B}(0; 1/\sqrt{\lambda}), \text{sprt} \left(C_\lambda^{u,h}(\chi_K) \right) \right)$$

with $C_\lambda^{u,h}(\chi_K)$ the computed upper compensated transform; (ii) the relative L^∞ error norm given by

$$e_{L^\infty} = \frac{\max_{x \in \mathbb{R}^2} |C_\lambda^{u,h}(\chi_K)(x) - C_\lambda^u(\chi_K)(x)|}{\max_{x \in \mathbb{R}^2} |C_\lambda^u(\chi_K)(x)|},$$

and (iii) the execution time t_c in seconds by a PC with processor Intel® Core™ i7-4510U CPU@2.00 GHz and 8 GB of memory RAM.

Figure 19 displays the support of $C_\lambda^u(\chi_K)$ given by $\overline{B}(0; 1/\sqrt{\lambda})$ and of $C_\lambda^{u,h}(\chi_K)$ computed by the numerical schemes mentioned above. Algorithm 2 and the parabola envelope algorithm yield the same results; thus, Fig. 19 displays the support as computed by only one of the two schemes. In this case, we observe that the support

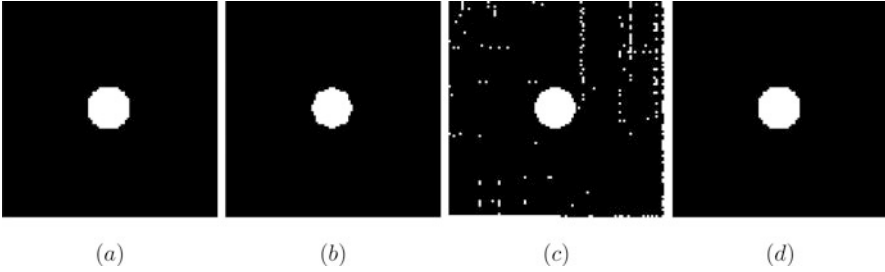


Fig. 19 Supports of the exact and computed upper compensated transform of the characteristic function of a singleton set of \mathbb{R}^2 by the different numerical schemes. **(a)** Exact support given by $\overline{B}(0; 1/\sqrt{\lambda})$ for $\lambda = 0.01$; **(b)** support of $C_\lambda^{u,h}(\chi_K)$ computed by Algorithm 1 (Oberman 2008); **(c)** support of $C_\lambda^{u,h}(\chi_K)$ computed by the biconjugate-based scheme (Lucet 1997; Helluy and Mathis 2011) for $h_d = 0.001$; **(d)** support of $C_\lambda^{u,h}(\chi_K)$ computed by Algorithm 2 (Zhang et al. (2021)) which coincides with the one computed using the parabola envelope scheme (Felzenszwalb and Huttenlocher 2012)

coincides with the exact one. This does not happen for the support computed by the other two schemes. The application of Algorithm 1 evidences the bias of the scheme with the underlying stencil, whereas by applying the biconjugate-based scheme, we note some small error all over the domain. The spread of this error depends on the dual mesh grid size h_d . Table 1 reports the values of t_c , e_{L^∞} and $d_{\mathcal{H}}$ for the different schemes. For the biconjugate-based scheme, we have different results according to the parameter h_d that controls the uniform discretization of the dual mesh. The value $h_d = 1$ means that we are considering the same grid size as the grid of the input function χ_K , whereas lower values for h_d means that we are computing on a finer dual mesh compared to the primal one. The results given in Table 1 show that in terms of the values of $C_\lambda^u(\chi_K)$, the biconjugate-based scheme is the one that produces the best results (compare the values of e_{L^∞}), but this occurs at the fraction of cost of reducing h_d which means to increase the number of the dual grid nodes and consequently the computational time. The issue of the choice of the dual grid on the accuracy of the computation of the convex envelope by the conjugate has been also tackled and recognized in Contento et al. (2015). However, as already pointed out in the analysis of Fig. 19, the support of $C_\lambda^{u,h}(\chi_K)$ computed by the biconjugate scheme is the one to yield the worst value for $e_{\mathcal{H}}$.

Intersection of Sampled Smooth Manifolds

In the following numerical experiments, we verify the effectiveness of the filter $I_\lambda(\cdot; K)$ introduced in section “Stable Multiscale Intersection Transform of Smooth Manifolds” and its Hausdorff stability property. We will consider both $2d$ - and $3d$ -geometries. The geometry is digitized and input as an image, but also other computer representations of the geometry can clearly be handled. This depends finally on the representation of the input geometry for the numerical scheme that

Table 1 Comparison between the different numerical schemes for the computation of $C_\lambda^u(\chi_K)$ for $\lambda = 0.01$. The symbol h_d refers to the dual mesh size of the scheme that computes the convex envelope via the biconjugate

		I_c	e_{L^∞}	$e_{\mathcal{H}}$	
Convex based schemes	Algorithm 1	1.9791	0.0390	1.7321	
	Biconjugate scheme	$h_d = 1$	0.1575	48	9.4999
		$h_d = 0.1$	0.2157	0.2400	9
		$h_d = 0.01$	0.5935	0.0142	7.6158
		$h_d = 0.001$	16.6603	0.0032	7.5498
Moreau based schemes	Algorithm 2	0.1246	0.0249	0	
	PE scheme	0.2553	0.0249	0	

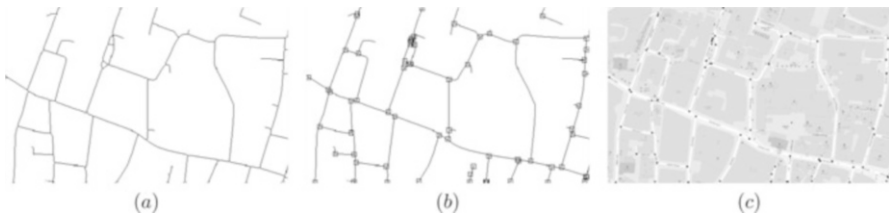


Fig. 20 (a) Medial axis of the road network; (b) location of the intersection points; (c) map of the road network and location of the intersection points shown in (b)

is used to compute the compensated transforms. Figure 20 displays a road network extract from a map of the city of London and represents a set of $2d$ curves which intersect to each other in different manner. The figure shows the position of the local maxima of $I_\lambda(\cdot; K)$ which are seen to coincide with all the crossing and turning points of the given curves. We also have some false positive due to the digitization of the road network.

Figure 21 displays the results of the application of the filter $I_\lambda(\cdot; K)$ to $3d$ geometries represented by point clouds. Figure 21a displays the Plücker’s conoid of parametric equation

$$x = v \cos u, \quad y = v \sin u, \quad z = \sin 4u \quad \text{for } u \in [0, 2\pi[, \quad v \in [-1, 1],$$

with the location of its singular lines and the parts of surface with higher curvature. Figure 21b depicts the intersections between manifolds of different dimensions, namely, in the figure, we have the Whitney umbrella of the implicit equation $x^2 = y^2z$, a cylinder, and a helix, with the location of their mutual intersections and also of where the Whitney surface intersects itself; finally, Fig. 21c displays the intersection between a cylinder, planes, and a helix.

The intersection of the line with the plane for the geometry shown in Fig. 21 is weaker than the geometric singularities of the surfaces. With this meaning, the values of the local maxima of $I_\lambda(\cdot; K)$ determine a scale between the different types

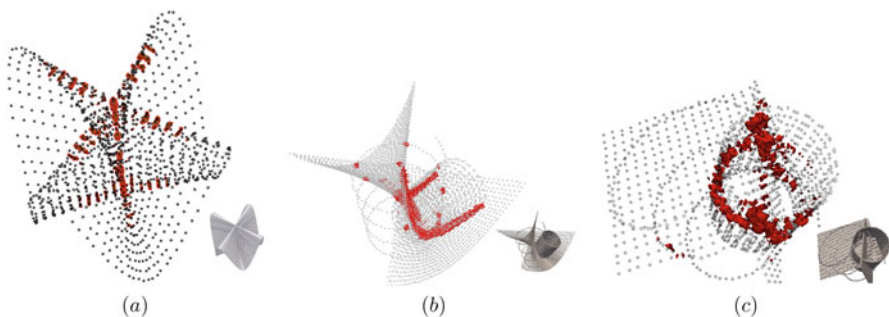


Fig. 21 (a) Plücker surface with identification of its singular lines and surface parts of higher curvatures; (b) intersections of the Whitney surface of equation $x^2 = y^2z$ with a helix and a cylinder; (c) intersections of planes with a cylinder and an helix

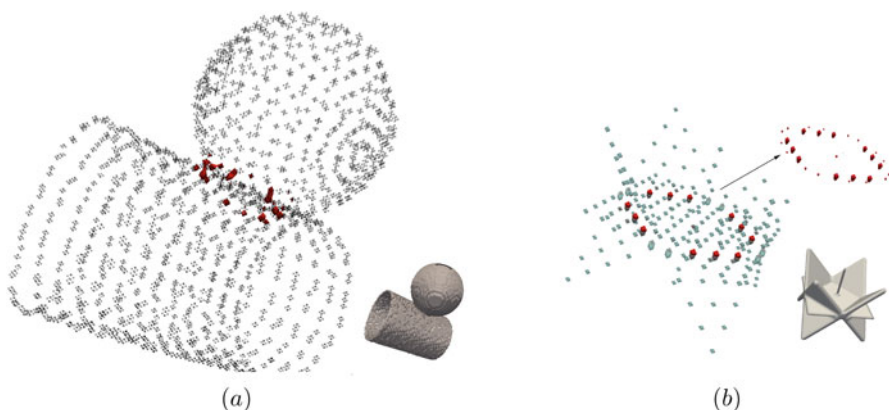


Fig. 22 (a) Tangential intersection of a sampled sphere and cylinder which are “almost” tangentially intersected and indication of the intersection marker; (b) intersection markers for the intersection among loosely sampled piecewise affine surfaces of equation $||10x - 75| - |10y - 75| + |10z - 75| - 45|=0$, the circle of equation $(10x - 75)^2 + (10z - 75)^2 \leq 45^2$ on the plane of equation $y = 75$ and the line of equation $x = 75, z = 75$

of intersections present in the manifold K and represent the multiscale nature of the filter $I_\lambda(\cdot; K)$.

Finally, the numerical experiments displayed in Fig. 22 refer to critical conditions that are not directly covered by the theoretical results we have obtained. Figure 22a shows the result of the application of $I_\lambda(\cdot; K)$ to a sphere and a cylinder that are “almost” tangentially intersecting each other, whereas Fig. 22b illustrates the results of the application of the filter to detect the intersection between loosely sampled piecewise affine functions, a plane and a line.

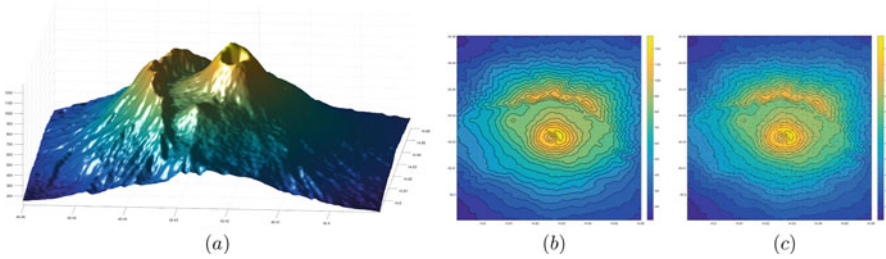


Fig. 23 Reconstruction of real-world digital elevation maps. (a) Ground truth model from USGS-STRM1 data relative to the area with geographical coordinates $[N 40^{\circ}23'25'', N 40^{\circ}27'37''] \times [E 14^{\circ}47'25'', E 14^{\circ}51'37'']$; (b) sample set K_1 formed by only level lines at regular height interval of 58.35 m. The set K_1 contains 14% of the ground truth points; (c) sample set K_2 formed by taking randomly 30% of the points belonging to the level lines of the set K_1 and scattered points corresponding to 5% density. The sample set K_2 contains 7% of the ground truth points

Approximation Transform

We report here on applications of the average approximation compensated convex transform developed in Zhang et al. (2016a, 2018) to three classes of problems. These include (i) surface reconstruction from real-world data using level lines and single points; (ii) salt & pepper noise restoration, and (iii) image inpainting.

Level Set Reconstruction

We consider here the problem of producing a digital elevation map from a sample of the the NASA SRTM global digital elevation model of Earth land. The data provided by the National Elevation Dataset (Gesch et al. 2009) contain geographical coordinates (latitude, longitude, and elevation) of points sampled at one arc-second intervals in latitude and longitude. For our experiments, we choose the region defined by the coordinates $[N 40^{\circ}23'25'', N 40^{\circ}27'37''] \times [E 14^{\circ}47'25'', E 14^{\circ}51'37'']$ extracted from the SRTM1 cell $N40E014.hgt$ (SRTMLandcover Download site). Such region consists of an area with extension $7.413 \text{ km} \times 5.844 \text{ km}$ and height varying between 115 m and 1282 m, with variegated topography features. In the digitization by the US Geological Survey, each pixel represents a $30 \text{ m} \times 30 \text{ m}$ patch. Figure 23a displays the elevation model from the SRTM1 data which we refer in the following as the ground truth model. We will take a sample f_K of such data; make the reconstruction using the $A_{\lambda}^M(f_K)$ computed with Algorithm 2 and the AMLE interpolant (Almansa et al. 2002; Caselles et al. 1998) using the MatLab® code described in Parisotto and Schönlieb (2016); and compare them with the ground truth model.

In the numerical experiments, we consider two sample data, characterized by different data density and typo of information. The first, which we refer to as sample set K_1 , consists only of level lines at regular height interval of 658.35 m and contains

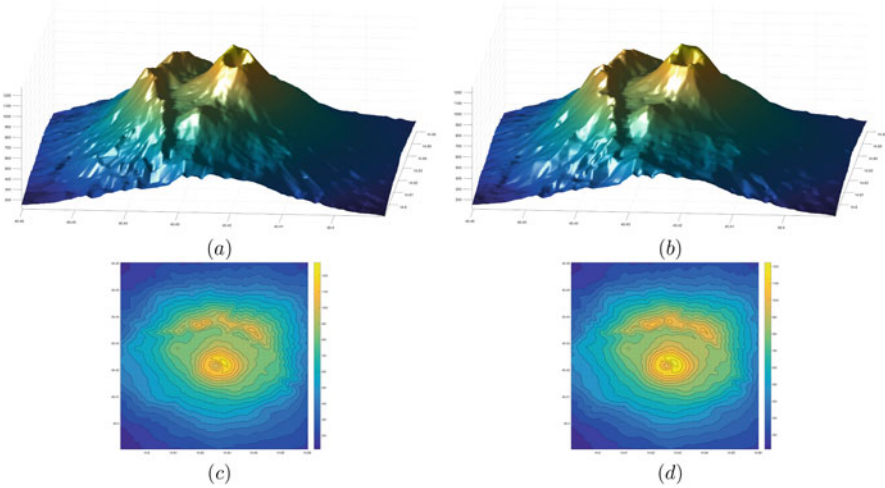


Fig. 24 Reconstruction of real-world digital elevation maps. **(a)** Graph of $A_\lambda^M(f_K)$ for sample set K_1 . Relative L^2 -Errors: $\epsilon = 0.0118, \epsilon_K = 0$. Parameters: $\lambda = 2 \cdot 10^3, M = 1 \cdot 10^6$. Total number of iterations: 3818; **(b)** graph of $A_\lambda^M(f_K)$ for sample set K_2 . Relative L^2 -Errors: $\epsilon = 0.0109, \epsilon_K = 0$. Parameters: $\lambda = 2 \cdot 10^3, M = 1 \cdot 10^6$. Total number of iterations: 1662; **(c)** isolines of $A_\lambda^M(f_K)$ from sample set K_1 at regular heights of 58.35 m; **(d)** isolines of $A_\lambda^M(f_K)$ from sample set K_2 at regular heights of 58.35 m

the 14% of the ground truth real digital data. The second sample set, denoted by K_2 , has been formed by taking randomly the 30% of the points belonging to the level lines of the set K_1 and scattered points corresponding to 5% density so that the sample set K_2 amounts to about 7% of the ground truth points. The two sample sets K_1 and K_2 are shown in Fig. 23b and c, respectively.

The graphs of the $A_\lambda^M(f_K)$ interpolant and of the AMLE interpolant for the two sample sets along with the respective isolines at equally spaced heights equal to 58.35 m are displayed in Figs. 24 and 25, respectively, whereas Table 2 contains the values of the relative L^2 -error ϵ on Ω and ϵ_K on the sample set K between such interpolants and the ground truth model, given by, respectively,

$$\epsilon = \frac{\|f - A_\lambda^M(f_K)\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}} \quad \text{and} \quad \epsilon_K = \frac{\|f_K - A_\lambda^M(f_K)\|_{L^2(K)}}{\|f_K\|_{L^2(K)}}, \quad (78)$$

where f is the ground truth model and $A_\lambda^M(f_K)$ is the average approximation of the sample f_K of f over K . We observe that while $A_\lambda^M(f_K)$ yields an exact interpolation of f_K over Ω , this is not the case for the AMLE approximation.

Though both reconstructions are comparable visually to the ground truth model, a closer inspection of the pictures shows that in the reconstruction from the synthetic data, the AMLE interpolant does not reconstruct correctly the mountains peaks, which appear to be smoothed and introduce artificial ridges along the slopes of the

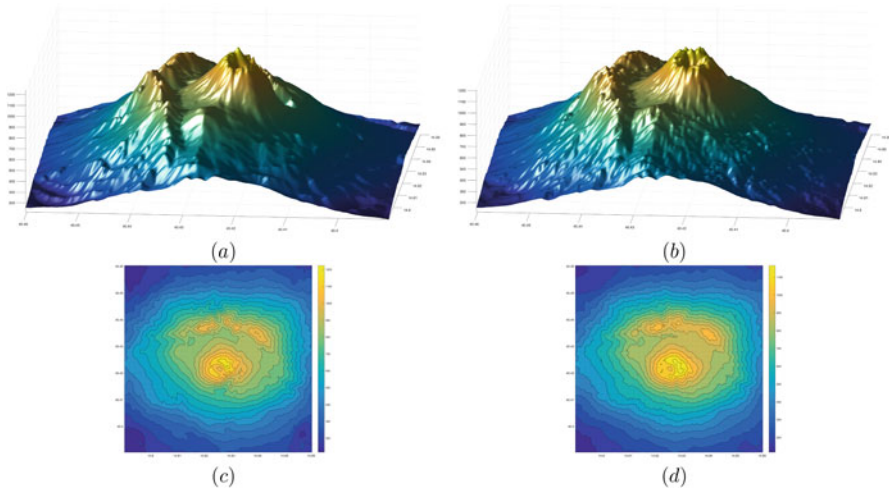


Fig. 25 Reconstruction of real-world digital elevation maps. (a) Graph of the AMLE interpolant from set K_1 . Relative L^2 -Error: $\epsilon = 0.0410$, $\epsilon_K = 0.0110$. Total number of iterations: 11542; (b) graph of the AMLE interpolant from set K_2 . Relative L^2 -Error: $\epsilon = 0.02863$, $\epsilon_K = 0.0109$. Total number of iterations: 12457; (c) isolines of the AMLE interpolant from sample set K_1 at regular heights of 58.35 m; (d) isolines of the AMLE interpolant from sample set K_2 at regular heights of 58.35 m

Table 2 Relative L^2 -error for the DEM reconstruction from the two sample sets using the $A_\lambda^M(f_K)$ and the AMLE interpolant. The realization of $\epsilon_K = 0$ for $A_\lambda^M(f_K)$ says that $A_\lambda^M(f_K)$ yields an exact interpolation of f_K over Ω , unlike the AMLE approximation

Sample set	ϵ		ϵ_K	
	$A_\lambda^M(f_K)$	AMLE	$A_\lambda^M(f_K)$	AMLE
K_1	0.0118	0.0410	0	0.0110
K_2	0.0109	0.0286	0	0.0109

mountains. In contrast, the $A_\lambda^M(f_K)$ interpolant appears to be better for capturing features of the ground truth model. Finally, we also note that though the sample set K_1 contains a number of ground truth points higher than the sample set K_2 , the reconstruction from K_2 appears to be better than the one obtained from K_1 . This behavior was found for both interpolations, though it is more notable in the case of the $A_\lambda^M(f_K)$ interpolant. By taking scattered data, we are able to get a better characterization of irregular surfaces, compared to the one obtained from a structured representation such as provided by the level lines.

Salt and Pepper Noise Removal

As an application of scattered data approximation to image processing, we consider here the restoration of an image corrupted by salt & pepper noise. This is an impulse-type noise that is caused, for instance, by malfunctioning pixels in camera sensors or

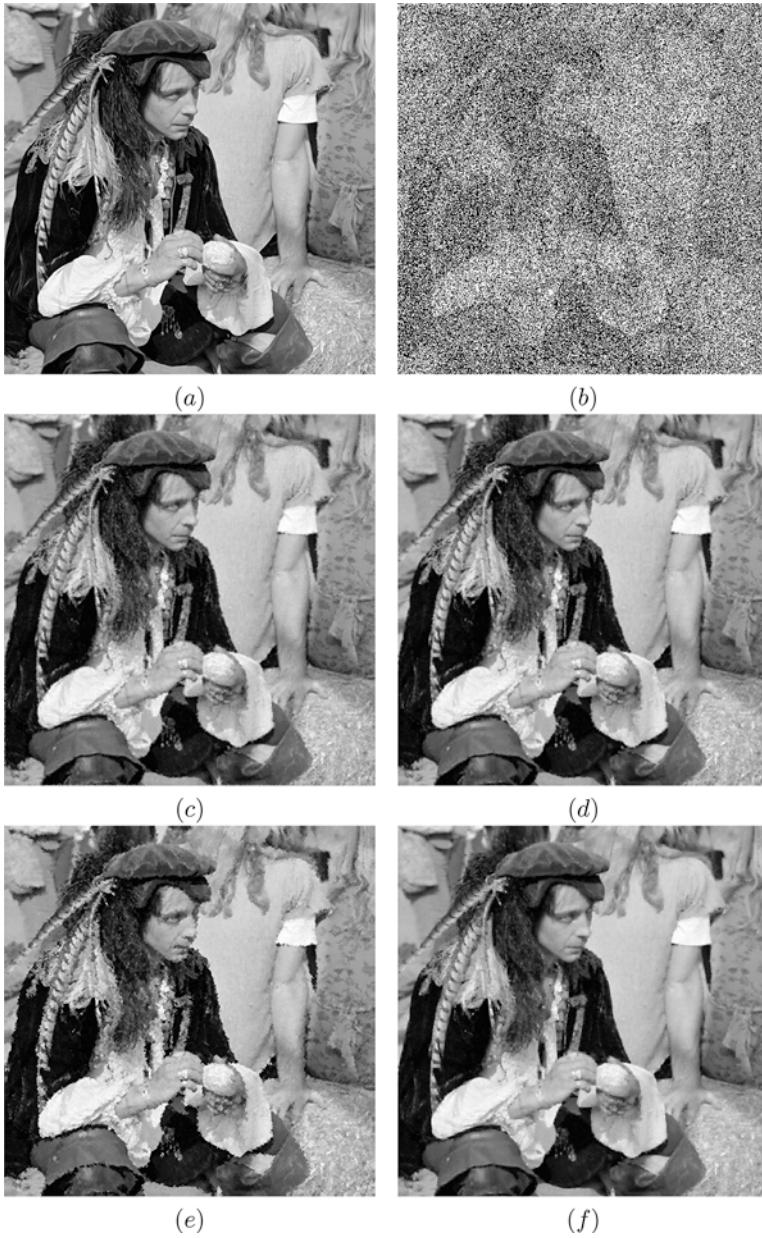


Fig. 26 (continued)

faulty memory locations in hardware, so that information is lost at the faulty pixels and the corrupted pixels are set alternatively to the minimum or to the maximum value of the range of the image values. When the noise density is low, about less than 40%, the median filter (Astola et al. 1997) or its improved adaptive median filter (Hwang and Haddad 1995) is quite effective for restoring the image. However, this filter loses its denoising power for higher noise density given that details and features of the original image are smeared out. In those cases, other techniques must be applied; one possibility is the two-stage TV-based method proposed in Chan et al. (2005) which consists of applying first an adaptive median filter to identify the pixels that are likely to contain noise and construct, thus a starting guess which is used in the second stage for the minimization of a functional of the form

$$F(u, y) = \Psi(u, y) + \alpha\Phi(u)$$

where y denotes the noisy image, Ψ is a data-fidelity term, and Φ is a regularization term, with $\alpha > 0$ a parameter. In the following numerical experiments, we consider the image displayed in Fig. 26a with size 512×512 pixels, damaged by 70% salt & pepper noise. The resulting corrupted image is displayed in Fig. 26b where on average only 78,643 pixels out of the total 262,144 pixels carry true information. The true image values represent our sample function f_K , whereas the set of the true pixels forms our sample set K . To assess the restoration performance, we use the peak signal-to-noise ratio (PSNR) which is expressed in the units of dB and, for an 8-bit image, i.e., with values in the range $[0, 255]$, is defined by

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{mn} \sum_{i,j} |f_{i,j} - r_{i,j}|^2} \quad (79)$$

where $f_{i,j}$ and $r_{i,j}$ denote the pixel values of the original and restored image, respectively, and m, n denotes the size of the image f . In our numerical experiments, we have considered the following cases. The first one assumes the set K to be given by the noise-free interior pixels of the corrupted image together with

←
Fig. 26 (a) Original image; (b) original image covered by a salt & pepper noise density of 70%. PSNR = 6.426 dB; (c) restored image $A_\lambda^M(f_K)$ by Moreau-based scheme (Algorithm 2) with the set K padded by two pixels. PSNR = 26.020 dB. $\lambda = 20$, $M = 1E13$. Total number of iterations: 21; (d) restored image $A_\lambda^M(f_K)$ by convex-based scheme (Algorithm 1) with the set K padded by two pixels. PSNR = 26.642 dB. $\lambda = 20$, $M = 1E13$. Total number of iterations: 1865; (e) restored image by the adaptive median filter (Hwang and Haddad 1995) used as starting guess for the two-stage TV-based method described in Cai et al. (2007) and Chan et al. (2005). Window size $w = 33$ pixels. PSNR = 22.519 dB; (f) restored image by the two-stage TV-based method described in Cai et al. (2007) and Chan et al. (2005) with the set K padded by two pixels. PSNR = 26.475 dB. Total number of iterations: 3853

the boundary pixels of the original image. In the second case, K is just the set of the noise-free pixels of the corrupted image, without any special consideration on the image boundary pixels. In analyzing this second case, to reduce the boundary effects produced by the application of Algorithms 1 and 2, we have applied our method to an enlarged image and then restricted the resulting restored image to the original domain. The enlarged image has been obtained by padding a fixed number of pixels before the first image element and after the last image element along each dimension, making mirror reflections with respect to the boundary. The values used for padding are all from the corrupted image. In our examples, we have considered two versions of enlarged images, obtained by padding the corrupted image with two pixels and ten pixels, respectively. Tables 3, 4, and 5 compare the values of the PSNR of the restored images by our method and the TV-based method applied to the corrupted image with noise-free boundary and to the two versions of the enlarged images with the boundary values of the enlarged images given by the padded noisy image data. We observe that there are no important variations in the denoising result between the different methods of treating the image boundary. This is also reflected by the close value of the PSNR of the resulting restored images. For 70% salt & pepper noise, Fig. 26c and d display the restored image $A_\lambda^M(f_K)$ by Algorithms 1 and 2, respectively, with K equal to the true set that has been enlarged by two pixels, whereas Fig. 26e and f show the restored image by the adaptive median filter and the TV-based method (Cai et al. 2007; Chan et al. 2005) using the same set K . Although the visual quality of the images restored from 70% noise corruption is comparable between our method and the TV-based method, the PSNR using our method with Algorithm 2 is higher than that for the TV-based method in all of the experiments reported in Tables 3, 4, and 5. An additional advantage of our method is its speed. Our method does not require initialization which is in contrast with the two-stage TV-based method, for which the initialization, for instance, is given by the restored image using an adaptive median filter.

Table 3 Comparison of PSNR of the restored images by the compensated convexity-based method ($A_\lambda^M(f_K)$) by applying the convex-based scheme (Algorithm 1) and the Moreau-based scheme (Algorithm 2), and by the two-stage TV-based method (TV), with the set K with noise-free boundary

	PSNR		
	K with noise-free boundary		
	$A_\lambda^M(f)$		TV
Algorithm 1	Algorithm 2		
Noise Density			
70% (6.426 dB)	26.634 dB	26.674 dB	26.506 dB
90% (5.371 dB)	22.968 dB	23.117 dB	22.521 dB
99% (4.938 dB)	18.357 dB	18.424 dB	17.420 dB

Table 4 Comparison of PSNR of the restored images by the compensated convexity-based method ($A_\lambda^M(f_K)$) by applying the convex-based scheme (Algorithm 1) and the Moreau-based scheme (Algorithm 2), and by the two-stage TV-based method (TV), with the set K padded by two pixels

	PSNR		
	K padded by two pixels		
	$A_\lambda^M(f)$		TV
Noise Density	Algorithm 1	Algorithm 2	
70% (6.426 dB)	26.020 dB	26.642 dB	26.475 dB
90% (5.371 dB)	22.654 dB	23.078 dB	22.459 dB
99% (4.938 dB)	18.026 dB	18.240 dB	17.314 dB

Table 5 Comparison of PSNR of the restored images by the compensated convexity-based method ($A_\lambda^M(f_K)$) by applying the convex-based scheme (Algorithm 1) and the Moreau-based scheme (Algorithm 2), and by the two-stage TV-based method (TV), with the set K padded by ten pixels

	PSNR		
	K padded by ten pixels		
	$A_\lambda^M(f)$		TV
Noise Density	Algorithm 1	Algorithm 2	
70% (6.426 dB)	26.020 dB	26.640 dB	26.468 dB
90% (5.371 dB)	22.654 dB	23.068 dB	22.446 dB
99% (4.938 dB)	18.026 dB	18.342 dB	17.330 dB

Finally, to demonstrate the performance of our method in some extreme cases of very sparse data, we consider cases of noise density equal to 90% and 99%. Figure 27 displays the restored image by the compensated convexity-based method and by the TV-based method for the case where K is padded by two pixels and ten pixels for 90% and 99% noise level, respectively. As far as the visual quality of the restored images is concerned, and to the extent that such judgment can make sense given the high level of noise density, the inspection of Fig. 27 seems to indicate that $A_\lambda^M(f_K)$ gives a better approximation of details than the TV-based restored image. This is also reflected by the values of the PSNR index in Tables 3, 4, and 5.

Inpainting

Inpainting is the problem where we are given an image that is damaged in some parts and we want to reconstruct the values in the damaged part on the basis of the known values of the image. This topic has attracted lot of interest especially as an application of TV-related models (Chan and Shen 2005; Schönlieb 2015). The main motivation is that functions of bounded variations provide the appropriate functional setting given that such functions are allowed to have jump discontinuities

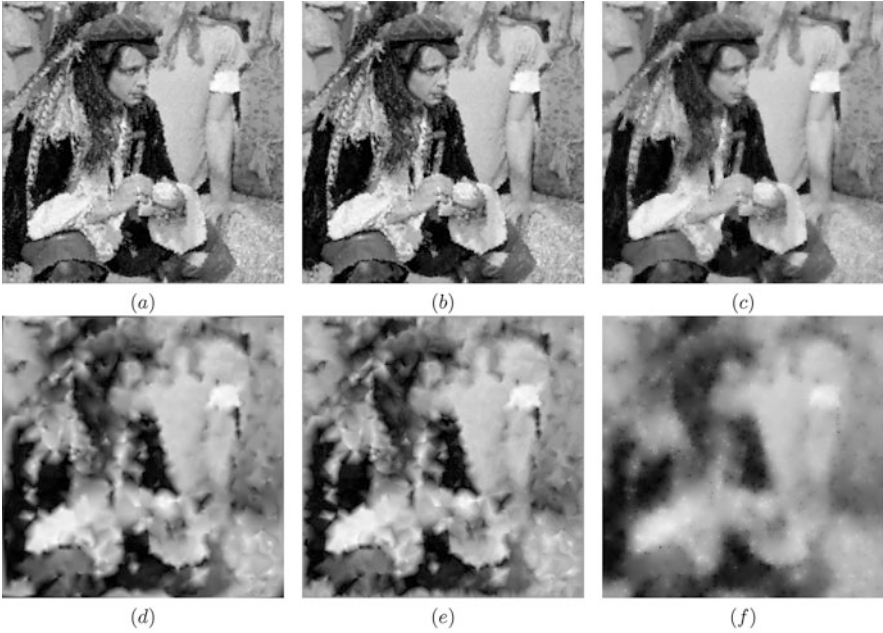


Fig. 27 Restoration of 90% corrupted image (PSNR = 5.372 dB) with the set K padded by two pixels. (a) Restored image $A_\lambda^M(f_K)$ by Moreau-based scheme (Algorithm 2). PSNR = 22.654 dB. $\lambda = 10$, $M = 1e13$. Total number of iterations: 32; (b) restored image $A_\lambda^M(f_K)$ by convex-based scheme (Algorithm 1). PSNR = 23.078 dB. $\lambda = 10$, $M = 1e13$. Total number of iterations: 10445; (c) restored image by the two-stage TV-based method described in Cai et al. (2007) and Chan et al. (2005). PSNR = 22.459 dB. Total number of iterations: 2679. Restoration of 99% corrupted image (PSNR = 4.938 dB), with the set K padded by ten pixels. (d) restored image $A_\lambda^M(f_K)$ by Moreau-based scheme (Algorithm 2). PSNR = 18.026 dB. $\lambda = 2$, $M = 1e13$. Total number of iterations: 78; (e) restored image $A_\lambda^M(f_K)$ by convex-based scheme (Algorithm 1). PSNR = 18.342 dB. $\lambda = 2$, $M = 1e13$. Total number of iterations: 54823; (f) restored image by the two-stage TV-based method described in Cai et al. (2007) and Chan et al. (2005). PSNR = 17.330 dB. Total number of iterations: 13125

(Ambrosio et al. 2000). These authors usually argue that continuous functions cannot be used to model digital image-related functions as functions representing images may have jumps (Chan and Shen 2005), which are associated with the image features. However, from the human vision perspective, it is hard to distinguish between a jump discontinuity, where values change abruptly, and a continuous function with sharp changes within a very small transition layer. By the application of our compensated convex-based average transforms, we are adopting the latter point of view. A comprehensive study of this theory applied to image inpainting can be found in Zhang et al. (2016a, 2018) where we also establish error estimates for our inpainting method and compare with the error analysis for image inpainting discussed in Chan and Kang (2006). We note that for the relaxed Dirichlet problem of the minimal graph (Chan and Kang 2006) or of the TV model used in Chan and



Fig. 28 Inpainting of a text overprinted on an image. **(a)** Input image; **(b)** restored image $A_{\lambda}^M(f_K)$ using Algorithm 2. PSNR = 39.122 dB. Parameters: $\lambda = 18$ and $M = 1 \cdot 10^5$. Total number of iterations: 19; **(c)** restored image by the AMLE method described in Schönlieb (2015) and Parisotto and Schönlieb (2016). PSNR = 36.406 dB. Total number of iterations: 5247; **(d)** restored image by the split Bregman inpainting method described in Getreuer (2012). PSNR = 39.0712 dB. Total number of iterations: 19

Kang (2006), as the boundary value of the solution does not have to agree with the original boundary value, extra jumps can be introduced along the boundary. By comparison, since our average approximation is continuous, it will not introduce such a jump discontinuity at the boundary.

To assess the performance of our reconstruction compared to state-of-art inpainting methods, we consider synthetic example where we are given an image f and we overprint some text on it. The problem is then removing the text overprinted on the image displayed in Fig. 28a and how close we can get to the original

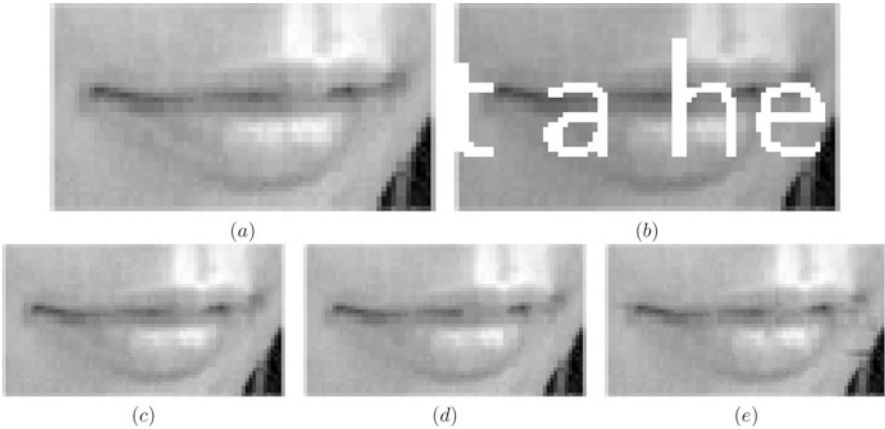


Fig. 29 Comparison of a detail of the original image with the corresponding detail of the restored images according to the compensated convexity method and the TV-based method. Lips detail of the: original image (a) without and (b) with overprinted text. Lips detail of the: (c) restored image $A_{\lambda}^M(f_K)$ using Algorithm 2; (d) AMLE-based restored image; (e) TV-based restored image

image f . If we denote by P the set of pixels containing the overprinted text, and by Ω the domain of the whole image, then $K = \Omega \setminus P$ is the set of the true pixels, and the inpainting problem is in fact the problem of reconstructing the image over P from knowing f_K , if we denote by f the original image values. We compare our method with the total variation-based image inpainting method solved by the split Bregman method described in Getreuer (2012) and with the AMLE inpainting reported in Schönlieb (2015). The restored image $A_{\lambda}^M(f_K)$ obtained by our compensated convexity method is displayed in Fig. 28b, and the restored image by the AMLE method is shown in Fig. 28d, whereas (c) presents the restored image by the split Bregman inpainting method. All the restored images look visually quite good. However, if we use the PSNR as a measure of the quality of the restoration, we find that $A_{\lambda}^M(f_K)$ has a value of PSNR equal to 39.122 dB, and the split Bregman inpainting restored image gives a value for PSNR = 39.071 dB, whereas the AMLE restored image has PSNR equal to 36.406 dB. To assess how well $A_{\lambda}^M(f_K)$ is able to preserve image details and not to introduce unintended effects such as image blurring and staircase effects, Fig. 29 displays details of the original image and of the restored images by the three methods. Once again, the good performance of $A_{\lambda}^M(f_K)$ can be appreciated visually.

We conclude this section with two real-world applications, where we actually do not know the true background picture f ; thus, the assessment of the inpainting must simply rely on the visual quality of the approximation. Figure 30 compares the results of the average compensated approximation and of the TV-based approximation in the case of the restoration of an image containing a scratch, whereas Fig. 31 refers to the removal of an unwanted thin object, the walking stick, from the picture. For both the examples, the two approximations yield qualitatively good results.

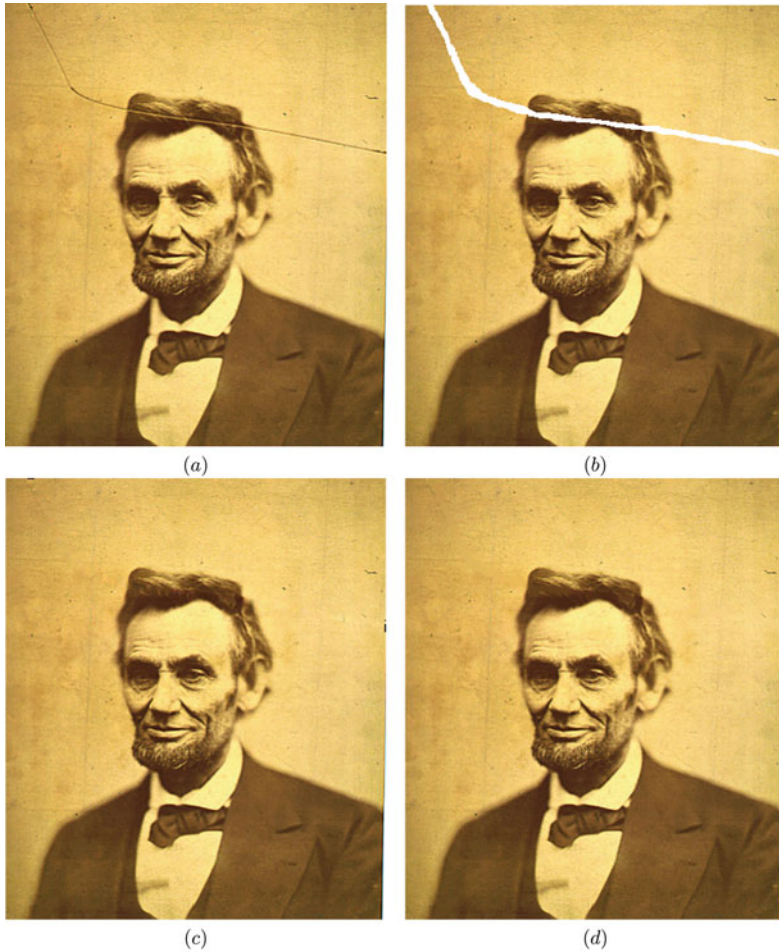


Fig. 30 Restoration of an old image. (a) Input image with the scratch; (b) input image with manual definition of the mask, given by the domain to repair; (c) restored image $A_{\lambda}^M(f_K)$ with $\lambda = 15$, $M = 10^6$; (d) TV-based restored image

Conclusions

Compensated convex transforms, or also known as proximity hull in the case of the lower transform, or gray scale opening and closing morphological operators with quadratic structuring elements in mathematical morphology, provide a geometric tight-approximation method for general functions that yields novel ways to smooth functions, to identify singularities in functions, and to interpolate and approximate data. Many of the compensated convex-based methods we have discussed in this paper have important Hausdorff stability properties that are especially significant



Fig. 31 Removal of a thin object from a picture. **(a)** Input image; **(b)** input image with manual definition of the mask, given by the domain to be inpainted; **(c)** restored image $A_{\lambda}^M(f_K)$ with $\lambda = 15$, $M = 10^6$; **(d)** TV-based restored image

for the extraction of information when data is presented in point-cloud form. The methods are also intrinsically multiscale, given that the parameters λ and/or τ that enter their definitions, provide scale for features that can thus be selected by the user. We have illustrated applications to image processing such as image inpainting

and restoration of image with high density of salt & pepper noise, to surface reconstruction from real-world data using level lines and isolated points, and to shape interrogation such as detection of intersection of sampled geometries and of line network such as in a city map. The performance of the methods and the accuracy of the results show that, when coupled with efficient numerical schemes, such as the linear-time numerical scheme that we have developed to compute the discrete Moreau envelope, the theory of compensated convex transforms provides a valid and feasible alternative to state-of-art methods especially for processing data without any a prior information or that are represented by point clouds.

Acknowledgments AO acknowledges the partial financial support of the Argentinian Research Council (CONICET) through the project PIP 11220170100100CO, the National University of Tucumán through the project PIUNT CX-E625, and the FonCyT through the project PICT 2016 201-0105 Prestamo Bid. EC is grateful for the financial support of the College of Science, Swansea University, and KZ wishes to thank the University of Nottingham for its support.

References

- Aichholzer O., Aigner W., Aurenhammer F., Hackl T., Jüttler B., Rabl M.: Medial Axis Computation for Planar Free-Form Shapes. *Comput. Aided Design* **41**, 339–349 (2009)
- Albano, P.: The regularity of the distance function propagates along minimizing geodesics. *Nonlinear Anal.* **95**, 308–312 (2014)
- Albano, P., Cannarsa, P., Nguyen, K.T., Sinestrari, C.: Singular gradient flow of the distance function and homotopy equivalence. *Math. Ann.* **356**, 23–43 (2013)
- Alberti, G., Ambrosio, L., Cannarsa, P.: On the singularities of convex functions. *Manuscr. Math.* **76**, 421–435 (1992)
- Almansa, A., Cao, F., Gousseau, Y., Rougé, B.: Interpolation of digital elevation models using AMLE and related methods. *IEEE Trans. Geosci. Remote Sens.* **40**, 314–325 (2002)
- Ambrosio, L., Tilli, P.: *Topics on Analysis in Metric Spaces*. Oxford University Press, New York (2004)
- Ambrosio, L., Fusco, N., Pallara D.: *Functions of Bounded Variation and Free Discontinuity Problems*. Clarendon Press, New York (2000)
- Amenta, N., Bern, M.: Surface reconstruction by Voronoi filtering. *Discret. Comput. Geom.* **22**, 481–504 (1999)
- Amenta, N., Choi, S., Kolluri, R.: The power crust, unions of balls, and the medial axis transform. *Comput. Geom-Theor. Appl.* **19**, 127–153 (2001)
- Astola, J., Kuosmanen, P.: *Fundamentals of Nonlinear Digital Filtering*. CRC Press, Boca Raton (1997)
- Attali, D., Montanvert, A.: Computing and simplifying 2D and 3D semicontinuous skeletons of 2D and 3D shapes. *Comput. Vis. Image Underst.* **67**, 261–273 (1997)
- Attouch, H., Aze, D.: Approximations and regularizations of arbitrary functions in Hilbert spaces by the Lasry-Lions methods. *Anal. Non-Lin. H. Poincaré Inst.* **10**, 289–312 (1993)
- Ball, J.M., James, R.D.: Fine phase mixtures as minimizers of energy. *Arch. Ration. Mech. Anal.* **100**, 13–52 (1987)
- Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**, 469–483 (1996)
- Blum, H.: A transformation for extracting new descriptors of shape. In: Dunn, W.W. (ed.) *Proc. Symposium on Models for the Perception of Speech and Visual Form*, pp. 362–380. MIT Press, Cambridge (1967)

- Brenier, Y.: Un algorithme rapide pour le calcul de transformées de Legendre-Fenchel discrètes. *C.R. Acad. Sci. Paris Sér. I Math.* **308**, 587–589 (1989)
- Cai, J.-F., Chan, R., Morini, B.: Minimization of an edge-preserving regularization functional by conjugate gradient type methods. In: Tai, X.-C., Lie, K.-A., Chan, T.F., Osher, S. (eds.) *Image Processing Based on Partial Differential Equations*, pp. 109–122. Springer, Heidelberg (2007)
- Cannarsa, P., Sinestrari, C.: *Semiconcave Functions, Hamilton-Jacobi Equations and Optimal Control*. Birkhäuser, Boston (2004)
- Carlsson, M.: On convex envelopes and regularization of non-convex functionals without moving global minima. *J. Optim. Theory Appl.* **183**, 66–84 (2019)
- Caselles, V., Morel, J.-M., Sbert, C.: An axiomatic approach to image interpolation. *IEEE Trans. Image Process.* **7**, 376–386 (1998)
- Chan, T.F., Kang, S.H.: Error analysis for image inpainting. *J. Math. Imag. Vis.* **26**, 85–103 (2006)
- Chan, T., Shen, J.: *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia (2005)
- Chan, R.H., Ho, C.-W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Trans. Image Process.* **14**, 1479–1485 (2005)
- Chazal, F., Soufflet, R.: Stability and finiteness properties of medial axis and skeleton. *J. Control Dyn. Syst.* **10**, 149–170 (2004)
- Contento, L., Ern, A., Vermiglio, R.: A linear-time approximate convex envelope algorithm using the double Legendre-Fenchel transform with application to phase separation. *Comput. Optim. Appl.* **60**, 231–261 (2015)
- Corrias, L.: Fast Legendre-Fenchel transform and applications to Hamilton-Jacobi equations and conservation laws. *SIAM J. Numer. Anal.* **33**, 1534–1558 (1996)
- Crandall, M.G., Ishii, H., Lions, P.-L.: User's guide to viscosity solutions of second order partial differential equations. *Bull. Am. Math. Soc.* **27**, 1–67 (1992)
- Dacorogna, B.: *Direct Methods in the Calculus of Variations*, 2nd edn. Springer, New York (2008)
- Dey, T.K.: *Curve and Surface Reconstruction*. Cambridge University Press, New York (2006)
- Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. *Theory Comput.* **8**, 415–428 (2012)
- Gesch, D., Evans, G., Mauck, J., Hutchinson, J., Carswell, W.J. Jr.: *The national map elevation*. U.S. Geological Survey Fact Sheet 3053 (2009)
- Getreuer, P.: Total variation inpainting using split Bregman. *Image Process. Line* **2**, 147–157 (2012)
- Hare, W.L.: A proximal average for nonconvex functions: a proximal stability perspective. *SIAM J. Optim.* **20**, 650–666 (2009)
- Hartman, P.: On functions representable as a difference of convex functions. *Pac. J. Math.* **9**, 707–713 (1959)
- Helluy, P., Mathis, H.: Pressure laws and fast Legendre transform. *Math. Models Methods Appl. Sci.* **21**, 745–775 (2011)
- Hiriart-Urruty, J.-B., Lemaréchal, C.: *Fundamentals of Convex Analysis*. Springer, Heidelberg (2001)
- Hwang, H., Haddad, R.A.: Adaptive median filters: new algorithms and results. *IEEE Trans. Image Process.* **4**, 499–502 (1995)
- Jackway, P.T.: Morphological scale-space. In: *IAPR International Conference on Pattern Recognition*, pp. 252–255. IEEE Computer Society Press, Los Alamitos (1992)
- Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A.: Skeletonization via distance maps and level sets. *Comput. Vis. Image Underst.* **62**, 382–391 (1995)
- Lasry, J.M., Lions, P.L.: A remark on regularization in Hilbert Spaces. *Israel Math. J.* **55**, 257–266 (1986)
- Lieutier, A.: Any open bounded subset of \mathbb{R}^n has the same homotopic type as its medial axis. *Comput. Aided Des.* **36**, 1029–1046 (2004)

- Lindeberg, T.: Generalized Gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space. *J. Math. Imag. Vis.* **40**, 36–81 (2011)
- Lucet, Y.: Faster than the Fast Legendre-Transform, the linear-time Legendre Transform. *Numer. Algorithms* **16**, 171–185 (1997)
- Lucet, Y.: Fast Moreau envelope computation I: numerical algorithms. *Numer. Algorithms* **43**, 235–249 (2006)
- Maragos, P., Schafer, R.: Morphological filters-Part I: their set theoretic analysis and relations to linear shift-invariant filters. *IEEE Trans. Acoust. Speech Sig. Process.* **35**, 1153–1169 (1987)
- Matheron, G.: Examples of topological properties of skeletons. In: Serra, J. (ed.) *Image Analysis and Mathematical Morphology*, Part II. Academic Press, San Diego (1988)
- Moreau, J.-J.: Proximité dualité dans un espace Hilbertien. *Bull. Soc. Math. Fr.* **93**, 273–299 (1965)
- Oberman, A.M.: Computing the convex envelope using a nonlinear partial differential equation. *Math. Models Methods Appl. Sci.* **18**, 759–780 (2008)
- Parikh, N., Boyd, S.: Proximal algorithms. *Found. Trends Optim.* **1**, 123–231 (2013)
- Parisotto, S., Schönlieb, B.-C.: MATLAB Codes for the Image Inpainting Problem, GitHub repository, MATLAB Central File Exchange, Sept 2016
- Patrikalakis, N.M., Maekawa, T.: *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer, Heidelberg (2002)
- Preparata, F.P., Shamos, M.: *Computational Geometry. An Introduction*. Springer, Berlin (1985)
- Reshetnyak, Y.G.: On a generalization of convex surfaces. *Mat. Sbornik* **40**, 381–398 (1956)
- Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton, New Jersey (1970)
- Rockafellar, R.T., Wets, R.J.-B.: *Variational Analysis*. Springer, Berlin (1998)
- Schönlieb, C.-B.: *Partial Differential Equation Methods for Image Inpainting*. Cambridge University Press, New York (2015)
- Serra, J.: *Image Analysis and Mathematical Morphology*. Academic Press, London (1982)
- Shih, F.Y., Mitchell, O.: A mathematical morphology approach distance transformation. *IEEE Trans. Image Process.* **1**, 197–204 (1992)
- Siddiqi, K., Pizer, S.M. (eds.): *Medial Representations*. Springer, New York (2008)
- Smith, S., Brady, J.: SUSAN—a new approach to low-level image processing. *Int. J. Comput. Vis.* **23**, 45–78 (1997)
- Soille, P.: *Morphological Image Analysis*, 2nd edn. Springer, Berlin (2004)
- SRTM and Landcover Download site. <http://ve2dbe.com/geodata/>. Accessed: 30 Sept 2020
- Tartar, L.: Estimations fines de coefficients homogénéisés. In: Krée, P. (ed.) *Ennio De Giorgi Colloquium. Research Notes in Mathematics*, vol. 125, pp. 168–187. Pitman, London (1985)
- van den Boomgaard, R.: The morphological equivalent of the Gauss convolution. *Nieuw Archief Voor Wiskunde* **10**, 219–236 (1992)
- Vese, L.: A method to convexify functions via curve evolution. *Commun. Partial Diff. Equ.* **24**, 1573–1591 (1999)
- Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart (1998)
- Zhang, K.: Compensated convexity and its applications. *Anal. Non-Lin. H. Poincaré Inst.* **25**, 743–771 (2008a)
- Zhang, K.: Convex analysis based smooth approximations of maximum functions and squared-distance functions. *J. Nonlinear Convex Anal.* **9**, 379–406 (2008b)
- Zhang, K., Crooks, E., Orlando, A.: Compensated convexity, multiscale medial axis maps and sharp regularity of the squared distance function. *SIAM J. Math. Anal.* **47**, 4289–4331 (2015a)
- Zhang, K., Orlando, A., Crooks, E.: Compensated convexity and Hausdorff stable geometric singularity extractions. *Math. Models Methods Appl. Sci.* **25**, 747–801 (2015b).
- Zhang, K., Orlando, A., Crooks, E.: Compensated convexity and Hausdorff stable extraction of intersections for smooth manifolds. *Math. Models Methods Appl. Sci.* **25**, 839–873 (2015c).
- Zhang, K., Crooks, E., Orlando, A.: Compensated convexity methods for approximations and interpolations of sampled functions in Euclidean spaces: theoretical foundations. *SIAM J. Math. Anal.* **48**, 4126–4154 (2016a)

- Zhang, K., Crooks, E., Orlando, A.: Compensated convex transforms and geometric singularity extraction from semiconvex functions (in Chinese). *Sci. Sin. Math.* **46**, 1–22 (2016b). (revised English version available at <https://arxiv.org/abs/1610.01451>)
- Zhang, K., Crooks, E., Orlando, A.: Compensated convexity methods for approximations and interpolations of sampled functions in Euclidean Spaces: applications to contour lines, sparse data and inpainting. *SIAM J. Imaging Sci.* **11**, 2368–2428 (2018)
- Zhang, K., Orlando, A., Crooks, E.: Compensated Convexity on Bounded Domains, Mixed Moreau Envelopes and Computational Methods. *Appl. Math. Model.* **94**, 688–720 (2021)



The Potts Model with Different Piecewise Constant Representations and Fast Algorithms: A Survey

52

Xuecheng Tai, Lingfeng Li, and Egil Bae

Contents

Introduction	1888
Representation by Integer-Valued Labeling Function	1893
Potts Model for Integer-Valued Functions	1893
Graph Cuts for the Integer-Labeled Potts Model	1895
Continuous Max-Flow Formulation for Integer-Valued Potts Model	1900
Numerical Algorithms for the Integer-Valued Continuous Max-Flow Problems	1903
Representation by Simplex-Constrained Vector Functions	1905
Primal-Dual Formulation for Simplex-Constrained Potts Model	1906
Dual Formulation for Simplex-Constrained Potts Model	1907
Continuous Max-Flow Formulation for Simplex-Constrained Potts Model	1908
Representation by Overlapping Functions	1911
Potts Model with Overlapping Binary Functions Representation	1911
Extension to More General Cases	1911
Convex Relaxation via Convex Envelope for Overlapping Representation	1913
Numerical Algorithms for Relaxed Potts Model via Overlapping Representation	1915
A Continuous Max-Flow Approach for 4-Phase Overlapping Binary Representation	1915
Extension to the High-Dimensional Graphical Models	1918
Constructing a Graph for a Given Data Set	1919

X. Tai (✉)

Hong Kong Center for Cerebro-cardiovascular Health Engineering (COCHE), Shatin, Hong Kong
e-mail: xtai@hkcoche.org

L. Li

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong, China

Department of Mathematics, Southern University of Science and Technology, Shenzhen, China
e-mail: lingfengli@life.hkbu.edu.hk

E. Bae

Norwegian Defence Research Establishment (FFI), Kjeller, Norway
e-mail: Egil.Bae@ffi.no

Graphical Potts Model with Simplex-Constrained Representation	1920
Efficient Inference in CRF Model	1922
Conclusion	1924
References	1924

Abstract

Markov random fields (MRF) and the Potts model have many applications in different areas. Especially, conditional random fields (CRF) and Potts model have been used in connection with classifiers. In this work, we focus on the Potts model and use image segmentation and data classification as examples to show some new techniques and fast algorithms for this model. We survey different piecewise constant representation techniques. Many of these representations can be interpreted as min-cut and max-flow problems on some special graphs. We will concentrate especially on the continuous setting and formulate continuous min-cut and max-flow models. When the min-cut/max-flow models are discretized, they give corresponding discrete min-cut/max-flow models on grids. Using these connections, we are able to turn the non-convex Potts model into some simple convex minimization problems with solutions that can be obtained by properly designed fast algorithms. In this survey, we will start by introducing some widely studied variational segmentation models and the classical level-set approaches to solve them. Then, we will describe three different piecewise constant representations for the general Potts model and their corresponding convex relaxations and fast algorithms. In the end, we will also generalize the method to a graph setting for high-dimensional data classifications. This survey presents the different techniques and algorithms in an integrated and self-contained manner.

Keywords

Image processing · Variational method · Graph theory · Potts model · Segmentation · Classification

Introduction

This work intends to give a survey on classification methods using conditional random field (CRF) and Potts models, especially with some new piecewise constant representations. These methods can be interpreted as solving continuous min-cut and max-flow problems and result in globally optimal solutions. We will start with image segmentation as an example to show these techniques.

Image segmentation or labeling is one of the most fundamental tasks in computer vision. Given an input image $I(x)$ defined on an open rectangular domain $\Omega \subset \mathbb{R}^2$, the goal of segmentation is to partition the image into different phases Ω_k , $k = 1, \dots, n$. Among all the image segmentation models, one of the most commonly studied models is the Potts model (Potts 1952; Geman and Geman 1984). The Potts model was first derived from statistical mechanics for modeling interacting spins on

crystalline lattice, and later people found it useful in computer vision and signal processing (Geman and Geman 1984; Boykov et al. 1998, 2001) using discrete optimization. More recently, the following continuous variational extension of the Potts model has become particularly popular:

$$\min_{\{\Omega_k\}_{k=1}^n} E_{Potts}(\{\Omega_k\}) = \sum_{k=1}^n \int_{\Omega_k} f_k(x) dx + \alpha \sum_{k=1}^n |\partial\Omega_k|, \quad (1)$$

$$\text{s.t. } \cup_{k=1}^n \overline{\Omega_k} = \Omega, \quad (2)$$

$$\Omega_k \cap \Omega_l = \emptyset, \quad \forall k \neq l, \quad (3)$$

where $f_k(x)$ is the data fidelity term for each phase and depends on the input image $I(x)$. The second term $|\partial\Omega_k|$, named edge force term, measures the length of the boundary of Ω_k . This term serves as a regularization which helps the model to generate segments with smooth and tight boundaries.

Another important model for image segmentation is the Mumford-Shah model (Mumford and Shah 1989) which is closely related to the Potts model (1). The Mumford-Shah model aims to find an optimal piecewise smooth function g to approximate the input image, while minimizing the Hausdorff measure of the discontinuity set of g . We focus on the case where the discontinuity set of g consists of closed curves $\partial\Omega_k$, $k = 1, \dots, n$, each encompassing a subregion Ω_k of Ω . The Mumford-Shah model can then be written as the partition problem:

$$\min_{g, \{\Omega_k\}_{k=1}^n} E_{MS}(g, \{\Omega_k\}) = \sum_{k=1}^n \int_{\Omega_k} (|g(x) - I(x)|^2 + |\nabla g(x)|^2) dx + \alpha \sum_{k=1}^n |\partial\Omega_k|, \quad (4)$$

$$\text{s.t. } \cup_{k=1}^n \overline{\Omega_k} = \Omega \quad (5)$$

$$\Omega_k \cap \Omega_l = \emptyset, \quad k \neq l. \quad (6)$$

By restricting $g(x)$ to be a piecewise constant function, one can get a simplified version of (4):

$$\min_{\{g_k\}_{k=1}^n, \{\Omega_k\}_{k=1}^n} E_{MS}(g, \{\Omega_k\}) = \sum_{k=1}^n \int_{\Omega_k} |g_k - I(x)|^2 dx + \alpha \sum_{k=1}^n |\partial\Omega_k|, \quad (7)$$

$$\text{s.t. } \cup_{k=1}^n \overline{\Omega_k} = \Omega \quad (8)$$

$$\Omega_k \cap \Omega_l = \emptyset, \quad k \neq l, \quad (9)$$

where $g(x)$ takes the value g_k in Ω_k . By fixing the values of g_k in (7), we get a special case of the Potts model with the fidelity term defined as

$$f_k(x) = |g_k - I(x)|^2. \quad (10)$$

An efficient implementation of the Mumford-Shah model was given by Chan and Vese (2001) and Vese and Chan (2002) using the level-set framework (Osher and Fedkiw 2003). For a subregion Ω_0 in Ω , a corresponding level-set function $\psi(x)$ satisfies

$$\begin{cases} \psi(x) > 0, & x \notin \overline{\Omega_0} \\ \psi(x) < 0, & x \in \Omega_0 \\ \psi(x) = 0, & x \in \partial\Omega_0 \end{cases} . \tag{11}$$

Then, the characteristic function of Ω_0 can be obtained by $1 - H(\psi)$ where H is the Heaviside function

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} . \tag{12}$$

We further define the Dirac delta function using the distributional derivative:

$$\delta(z) = \frac{d}{dz} H(z). \tag{13}$$

The perimeter of $\partial\Omega_0$ can then be represented as (Chan and Vese 2001):

$$|\partial\Omega_0| = \int_{\Omega} |\nabla H(\psi(x))| dx = \int_{\Omega} \delta(\psi(x)) |\nabla \psi(x)| dx. \tag{14}$$

Therefore, for a simple two-phase segmentation problem, the Mumford-Shah model (7) via level-set representation can be rewritten as

$$\begin{aligned} \min_{\psi, c_1, c_2} \int_{\Omega} & |I(x) - c_1|^2 (1 - H(\psi(x))) + |I(x) - c_2|^2 H(\psi(x)) \\ & + \delta(\psi(x)) |\nabla \psi(x)| dx, \end{aligned} \tag{15}$$

which is also well-known as the Chan-Vese model (Chan and Vese 2001). For the multiphase problem, Vese and Chan (2002) suggests to use several overlapping level-set functions to encode the different regions. One commonly used level-set function for model (15) is the signed distance function (SDF) (Sussman et al. 1994) which satisfies the eikonal equation

$$|\nabla \psi| = 1. \tag{16}$$

When minimizing the functional (15) using gradient flow approach, we update ψ iteratively. During each iteration, we often need to compute $|\nabla \psi|$, so this property

can make the computation much easier and faster. However, a reinitialization step should be performed iteratively to make sure that $|\nabla\psi| = 1$ holds.

Though the level-set function is very useful in image segmentation, it still suffers from some disadvantages. First, a single level-set function can only represent two phases. To represent multiphases, one need to use several overlapping (Vese and Chan 2002) or non-overlapping (Zhao et al. 1996) level-set functions, which will increase the cost of storage. Second, a computationally expensive reinitialization step is needed during the iterations. Third, the non-differentiability of the Heaviside function and delta function may cause extra difficulty for the computation.

In Lie et al. (2005, 2006b), the authors proposed a new method for representing different phases. The new method, named piecewise constant level-set method (PCLSM), uses one piecewise constant functions to represent multiple phases and each phase corresponds to a unique constant value. Consequently, the PCLSM requires less storage compared to the classical level-set method. Other piecewise constant representations have also been studied in Chan et al. (2006), Lellmann et al. (2009), Lie et al. (2006a), and Zach et al. (2008). One of the most important advantages of PCLSM is that good convex relaxations can be obtained and many efficient algorithms for convex optimization can be applied. Usually, the convex relaxations are derived by relaxing the integrality constraints of the piecewise constant functions (Bae and Tai 2015; Bae et al. 2011; Chan et al. 2006; Lellmann et al. 2009; Pock et al. 2008, 2009; Zach et al. 2008), which is inspired by the seminal work Chan et al. (2006) and Strang (1983). Generally speaking, to represent n disjoint subregions $\{\Omega_k\}_{k=1}^n$ in terms of piecewise constant functions, there are in particular three classical ways:

1. Integer-valued labeling function (Lie et al. 2005, 2006b): We first assign a unique integer value l_k to each subregion Ω_k and define a labeling function $\phi : \Omega \rightarrow \{l_1, \dots, l_n\}$ such that $\phi(x) = l_k$ if and only if $x \in \Omega_k, k = 1, \dots, n$. Then, each Ω_k can be represented by

$$\Omega_k = \{x \in \Omega | \phi(x) = l_k\}. \quad (17)$$

This representation requires only one function to represent n subregions. However, different choices and ordering of the labels l_k may affect the partition result.

2. Simplex-constrained vector functions (Lellmann et al. 2009; Zach et al. 2008): Instead of using one labeling function, here we define a vector-valued labeling function $v : \Omega \rightarrow \Delta^n$ where the constraint set is defined as

$$\Delta^n = \left\{ v(x) = (v_1(x), \dots, v_n(x)) \in \{0, 1\}^n \mid \sum_{k=1}^n v_k(x) = 1, \forall x \in \Omega \right\}, \quad (18)$$

and

$$v_k(x) = \begin{cases} 1, & x \in \Omega_k \\ 0, & x \notin \Omega_k \end{cases}, \quad k = 1, \dots, n. \tag{19}$$

In this representation, each $v_k(x)$ is also served as an indicator function of Ω_k . A related variant with a different parameterization of the unit simplex was proposed in Pock et al. (2009) and Chambolle et al. (2012). Although this method requires more storage for the constraint set of the dual variables, it avoids the problem of choosing the integer labels and can better approximate the boundary regularization $|\partial\Omega_k|$ in (1) after relaxing the integrality constraints.

3. **Overlapping binary functions:** Define $m = \log_2(n)$ binary functions $(\phi^1, \dots, \phi^m) : \Omega \rightarrow \{0, 1\}^m$ such that $x \in \Omega_k$ if and only if $(\phi^1(x) \dots \phi^m(x))$ is the binary representation of the integer k . This representation was pioneered in a level-set framework in Vese and Chan (2002) and the resulting optimization problem is often called the Chan-Vese model. The use of binary functions for the multiphase Chan-Vese model in the continuous setting was done in Chan et al. (2006) and Lie et al. (2006a). It was observed in Chan et al. (2006) that it is possible to give a convex relaxation to these binary models. In fact, this idea can be easily generalized to overlapping integer-valued functions and vector-valued functions.

In this survey, we will further assume that all the labeling functions, i.e., ϕ , v_k , and ϕ^k , belong to the bounded variation space:

$$BV(\Omega) := \{\phi \in L^1(\Omega) | TV(\phi; \Omega) < \infty\}, \tag{20}$$

where $TV(\phi; \Omega)$ is the total variation (TV) of ϕ on Ω and is defined as

$$TV(\phi; \Omega) = \sup_{\psi(x) \in C_c^1(\Omega)} \left\{ \int_{\Omega} \phi(x) \operatorname{div}(\psi(x)) dx \mid \|\psi\|_{\infty} \leq 1 \right\}. \tag{21}$$

If $\phi \in C^1(\Omega)$, we can further derive that

$$TV(\phi; \Omega) = \sup_{\psi(x) \in C_c^1(\Omega, \mathbb{R}^2)} \left\{ - \int_{\Omega} (\nabla\phi(x))^T \psi(x) dx \mid \|\psi\|_{\infty} \leq 1 \right\} = \int_{\Omega} |\nabla\phi(x)| dx. \tag{22}$$

Sometimes people can also use the distributional derivative or weak derivative of ϕ to denote the TV as $\int_{\Omega} |\nabla\phi(x)|$. Total variation was first introduced to image processing in Rudin et al. (1992) for denoising problems. Because the $BV(\Omega)$ space allows every shape discontinuity in the functions, it is very suitable for image analysis. One can refer to Acar and Vogel (1994) and Chambolle et al. (2010) for more detailed analysis of total variation and its application in image processing problems.

In the domain of statistics and probability, the Potts model (1) with piecewise constant representation is often referred to as the Markov random field (MRF) (Geman and Geman 1984). A MRF is a set of random variables, which are defined on an undirected graph, satisfying the Markov property. In other words, the status of one random variable only depends on the status of its neighbors described by a graph. Suppose $I(x)$ is an input image and $\omega(x) \in \{l_1, \dots, l_n\}$ is a labeling function. Then, the MRF model defines the a posteriori probability distribution using the Gibbs distribution:

$$P(\omega|I) = \frac{1}{Z} \exp(-U(\omega)), \quad (23)$$

where Z is a normalization factor and U is a potential function. Based on the Hammersley-Clifford theorem, the potential U must have a specific form, and the Potts model (1) is actually a special case of the MRF model. Therefore, the MRF model can also be viewed as a generalization of the Potts model (Boykov et al. 1998). We can also easily observe that the solution of maximizing the a posteriori probability with respect to ω is equivalent to the solution of minimizing the potential energy $U(\omega)$. One important variant of the MRF model is the conditional random field (CRF) (Lafferty et al. 2001) which can incorporate more features of $I(x)$ into the spatial regularization.

In the rest of this survey, we will introduce the Potts model formulation, convex relaxation, and fast algorithm for 3 different piecewise constant representations, respectively, in sections “Representation by Integer-Valued Labeling Function”, “Representation by Simplex-Constrained Vector Functions”, and “Representation by Overlapping Functions”. Then, in section “Extension to the High-Dimensional Graphical Models”, we will discuss the possible extension of the Potts model to graphs.

Representation by Integer-Valued Labeling Function

Potts Model for Integer-Valued Functions

In Lie et al. (2005, 2006a,b), the piecewise constant representation was proposed and applied to the Mumford-Shah model. The main idea of this method is to seek a partition of the domain Ω into n subdomains $\Omega_k, k = 1, 2, \dots, n$. A piecewise constant function ϕ is used to identify the subdomains

$$\phi(x) = l_k \quad x \in \Omega_k, \quad (24)$$

where $l_k < l_{k+1}$ for $k = 1, 2, \dots, n - 1$. Once the function ϕ is identified, we can construct the corresponding characteristic functions for each subdomain Ω_k as

$$\psi_k(x) = \frac{1}{z_k} \prod_{\substack{j=1 \\ j \neq k}}^n (\phi(x) - l_j), \quad \text{with} \quad z_k = \prod_{\substack{j=1 \\ j \neq k}}^n (l_k - l_j). \tag{25}$$

If ϕ is defined as in (24), we have $\psi_k(x) = 1$ for $x \in \Omega_k$, otherwise we have $\psi_k(x) = 0$. Based on these characteristic functions, we can extract the geometrical information of the boundaries of the subdomains $\{\Omega_k\}_{k=1}^n$. For example, the length of the interfaces surrounding each subdomain $\Omega_k, k = 1, 2, \dots, n$, should be

$$|\partial\Omega_k| = \int_{\Omega} |\nabla\psi_k| dx. \tag{26}$$

Typically, $|\nabla\psi|$ can be defined using the L_1 norm or L_2 norm of $\nabla\phi$, i.e., as

$$\int_{\Omega} |\phi_x| + |\phi_y|, \quad \text{or} \quad \int_{\Omega} \sqrt{|\phi_x|^2 + |\phi_y|^2}.$$

The corresponding total variation is called the isotropic TV and anisotropic TV, respectively. The anisotropic TV can be handled by both graph cuts and continuous minimization algorithms but leads to a grid bias that favors boundary curves with tangential lines in the vertical and horizontal directions. The isotropic TV leads to smoother boundary curves with no bias in the orientation of the tangential lines, but due to the coupling between the directional derivatives ϕ_x and ϕ_y it is not graph representable and can thus not be solved by graph cuts. A major advantage of the continuous minimization algorithm is their ability to handle the isotropic TV.

The Potts model can now be equivalently written as the following minimization problem:

$$\min_{\phi \in \{l_1, \dots, l_n\}} E(\phi) = \int_{\Omega} f(x, \phi(x)) dx + \alpha \sum_{k=1}^n \int_{\Omega} |\nabla\psi_k| dx, \tag{27}$$

where $f(x, \phi(x))$ is the data term and is defined as

$$f(x, \phi(x)) = \begin{cases} f_k(x) & \text{if } \phi(x) = l_k \\ +\infty & \text{else} \end{cases}. \tag{28}$$

In (28), $f_k(x)$ is a given region force function for the k -th region. In the Mumford-Shah model, $f_k(x) = |I(x) - c_k|^2$, where c_k is the average intensity in the k -th region, and it is assumed to be a constant. It is easy to see that

$$\phi(x) = \sum_{k=1}^n l_k \psi_k(\phi(x)), \quad \text{and} \quad \nabla\psi_k(x) = \psi'_k(\phi(x)) \nabla\phi(x).$$

Thus, there exist two constants $\rho_1(n) > 0, \rho_2(n) > 0$, such that

$$\rho_1(n) \int_{\Omega} |\nabla\phi|dx \leq \sum_{k=1}^n \int_{\Omega} |\psi_k(\phi)|dx \leq \rho_2(n) \int_{\Omega} |\nabla\phi|dx. \tag{29}$$

Unless ‘‘symmetry’’ is a crucial issue for the segmentation problem, we replace the regularization term in (27) and solve the following minimization problem

$$\min_{\phi \in \{l_1, \dots, l_n\}} E(\phi) = \int_{\Omega} f(x, \phi)dx + \alpha \int_{\Omega} |\nabla\phi|dx. \tag{30}$$

Notice that the regularization term $\int_{\Omega} |\nabla\phi|dx$ does not perfectly resemble the total boundary length $\sum_{k=1}^n \int_{\Omega} |\nabla\psi_k|$ and some edges will be counted multiple times.

Graph Cuts for the Integer-Labeled Potts Model

Instead of solving the Euler-Lagrange equation, graph cuts algorithms have been proposed to solve the minimization problem (30). Graph cut is a well-established technique in computer vision that has been used for solving discrete optimization problems (Boykov and Kolmogorov 2001; Boykov et al. 1998, 2001; Szeliski et al. 2006; Darbon and Sigelle 2006a,b) arising in image segmentation, stereo reconstruction, and image restoration, among others. We will give a brief introduction of this algorithm in the following.

We first consider an image $I : \hat{\Omega} \rightarrow \mathbb{R}$ defined on a discrete image domain $\hat{\Omega}$ of size $M \times N$ with symmetric boundary conditions. Let \mathcal{V} be the set of all grid points in $\hat{\Omega}$ and $\phi : \mathcal{V} \rightarrow \{l_1, \dots, l_n\}$ be a labeling function. For the sake of simplicity, here we assume $l_1 = 1, l_2 = 2, \dots, l_n = n$. We then define a neighborhood system $\mathcal{N} \subseteq \mathcal{V} \times \mathcal{V}$ over \mathcal{V} such that each pair of adjoining grid points belongs to \mathcal{N} . For example, let $x_i = (i_1, i_2) \in \mathcal{V}, x_j = (i_1 + 1, i_2) \in \mathcal{V}$, then (x_i, x_j) and (x_j, x_i) belongs to \mathcal{N} . If $x_j = (i_1 + 1, i_2 + 1)$, then (x_i, x_j) and (x_j, x_i) are not in \mathcal{N} . We also set the edge force term in (30) as the anisotropic TV, i.e.,

$$|\nabla\phi| = |\partial_1\phi| + |\partial_2\phi|, \tag{31}$$

where ∂_1 and ∂_2 are the partial derivatives with respect to the two independent variables. $|\nabla\phi(x_i)|$ can then be approximated by

$$|\nabla\phi(x_i)| \approx \frac{1}{2} \sum_{x_j \in \mathcal{N}_i} |\phi(x_j) - \phi(x_i)|, \tag{32}$$

where $\mathcal{N}_i = \{x_j \in \mathcal{V} | (x_i, x_j) \in \mathcal{N}, (x_j, x_i) \in \mathcal{N}\}$ is the set of all neighbors of x_i . Therefore, we write the discretized version of (30) as follows:

$$\min_{\phi \in \{1, 2, \dots, n\}} E_d(\phi) = \sum_{x_i \in \mathcal{V}} f(x_i, \phi(x_i)) + \frac{\alpha}{2} \sum_{(x_i, x_j) \in \mathcal{N}} |\phi(x_i) - \phi(x_j)|, \quad (33)$$

In case $n = 2$, one can solve (33) by the graph cut technique (Boykov and Kolmogorov 2001). In Ishikawa (2003), the author constructs a special graph to simulate the discrete Potts energy (33) for any natural $n \geq 2$ and converts this problem into a min-cut problem, which can be solved in polynomial time. A slightly improved version of the graph, which involved one less layer of vertex duplication, was proposed in Bae and Tai (2009b). A graph model usually contains a set of vertices and a set of edges which connect some pairs of vertices. Each edge is also assigned a positive number as the cost or weight of this edge. A similar graph cut approach has also been used to solve the image restoration problem (Darbon and Sigelle 2006b). In case the data term $f(x, \phi(x))$ is a convex function in $\phi(x)$, it is also possible to convert (30) to an independent sequence of binary TV minimization problems, which each can be solved using graph cuts (Darbon and Sigelle 2006a).

To use the graph cut algorithm for the multiphase segmentation problem (33) for general data terms $f(x, \phi(x))$, we will construct a graph such that there is a one-to-one correspondence between cuts in the graphs and labeling functions ϕ . Assuming ϕ is constrained to take n discrete values, we need to duplicate the vertices in \mathcal{V} by $n - 2$ times. The constructed graph is denoted as $\mathcal{G} = (\mathcal{V}^*, \mathcal{E})$ where \mathcal{V}^* is the vertices set and \mathcal{E} is the edges set. The vertices set is defined as:

$$\mathcal{V}^* = \left\{ x_i^{(k)} \mid x_i \in \mathcal{V}, k \in \{1, \dots, n - 1\} \right\} \cup \{s, t\}, \quad (34)$$

where s and t are two vertices named source and sink. For ease of notation, we will also let $x_i^{(0)}$ and $x_i^{(n)}$ denote s and t , respectively. The edge set \mathcal{E} is divided into three groups: \mathcal{E}_D corresponds to the data fidelity term in (33), \mathcal{E}_R corresponds to the TV regularization term in (33), and \mathcal{E}_C are edges that constrains the labeling functions. They are respectively defined as

$$\mathcal{E}_D = \left\{ (x_i^{(k-1)}, x_i^{(k)}) \mid x_i \in \mathcal{V}, k = 1, \dots, n \right\}, \quad (35)$$

$$\mathcal{E}_R = \left\{ (x_i^{(k)}, x_j^{(k)}) \mid (x_i, x_j) \in \mathcal{N}, k \in \{1, \dots, n - 1\} \right\}, \quad (36)$$

$$\mathcal{E}_C = \left\{ (x_i^{(k)}, x_j^{(k-1)}) \mid x_i \in \mathcal{V}, k = 1, \dots, n \right\}. \quad (37)$$

An illustration of the graph for segmenting a 1D signal of six grid points into four regions is shown in Fig. 1a. The corresponding segmentation of the 1D signal is shown in (b), where numbers indicate the integer labels assigned to each grid point.

The costs of these edges in the graph will be constructed such that the cost of each feasible cut on the graph is equal to the energy value (33) of the corresponding labeling function. The costs are set to be:

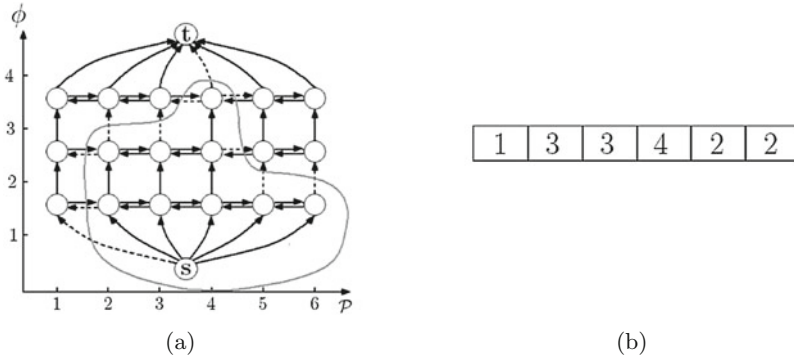


Fig. 1 Example of a graph for segmenting a 1D signal of 6 grid points into 4 regions. (a) The graph corresponding to a 1D signal of 6 grid points. Marked in gray is an example of a cut. (b) The values of the level-set function ϕ at each point of the 1D signal, corresponding to the example cut show in (a)

$$c(x_i^{(k-1)}, x_i^{(k)}) = f(x_i, \phi(x_i) = k) \quad x_i \in \mathcal{V}, k \in \{1, 2, \dots, n\}, \tag{38}$$

$$c(x_i^{(k)}, x_j^{(k)}) = \alpha/2 \quad (x_i, x_j) \in \mathcal{N}, k \in \{1, 2, \dots, n-1\}, \tag{39}$$

$$c(x_i^{(k+1)}, x_i^{(k)}) = +\infty \quad x_i \in \mathcal{V}, k \in \{0, 2, \dots, n-1\}, \tag{40}$$

Above we have used the convention that $x_i^n = t$ and $x_i^0 = s$. A cut on the graph $\mathcal{G} = (\mathcal{V}^*, \mathcal{E})$ is a partition of points in \mathcal{V}^* into two sets \mathcal{S} and \mathcal{T} , such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. Moreover, for every node $v \in \mathcal{S}$, there should be a path of edges from the source s to v that only visits nodes in \mathcal{S} . Similarly, for every node $v \in \mathcal{T}$, there should be a path of edges from v to the sink t that only visits nodes in \mathcal{T} . The cost of the cut is defined as the accumulated cost of all edges with one end node in \mathcal{S} and the other end node in \mathcal{T} . The min-cut problem is to find the cut of minimum cost and is formulated mathematically as:

$$\min_{\mathcal{S}, \mathcal{T}} \left\{ C(\mathcal{S}, \mathcal{T}) = \sum_{\substack{x_i^* \in \mathcal{S}, x_j^* \in \mathcal{T} \\ (x_i^*, x_j^*) \in \mathcal{E}}} c(x_i^*, x_j^*) \right\}. \tag{41}$$

For $x_i \in \mathcal{V}$, let's consider a subset of \mathcal{V}^* : $\mathcal{V}_i^* = \{x_i^{(k)}\}_{k=0}^n$, and a subset of \mathcal{E}_D : $\mathcal{E}_i = \{(x_i^{(k-1)}, x_i^{(k)})\}_{k=1}^n$. Because the costs of all constraint edges are infinite, a feasible cut, i.e., a cut with finite cost, on this graph must partition $\{x_i^{(k)}, \dots, x_i^{(n-1)}, x_i^{(n)}\}$ into \mathcal{T} and the rest into \mathcal{S} for some $k = 1, \dots, n$. In other words, there is one and only one edge in \mathcal{E}_i is cut for each $x_i \in \mathcal{V}$. Otherwise a constraint edge will be cut and the cost will be infinity. An example of feasible cuts is shown in Fig. 1. Now we introduce a set of characteristic functions $\lambda(x_i) = (\lambda_0(x_i), \dots, \lambda_n(x_i))$:

$$\lambda_k(x_i) = \begin{cases} 1 & x_i^{(k)} \in \mathcal{S} \\ 0 & x_i^{(k)} \in \mathcal{T} \end{cases}, \forall x_i \in \mathcal{V}, \quad k = 0, \dots, n, \tag{42}$$

where $\lambda_0 = 1$ and $\lambda_n = 0$ are constant functions. In addition, we require λ to belong to the constraint set:

$$C_\lambda = \{(\lambda_0, \dots, \lambda_n) : \mathcal{V} \rightarrow \{0, 1\}^{n+1} | 1 = \lambda_0(x_i) \geq \lambda_1(x_i) \geq \dots \geq \lambda_n(x_i) = 0, \forall x_i \in \mathcal{V}\}. \tag{43}$$

One can observe that each feasible cut on \mathcal{G} can be uniquely represented by a λ in C_λ . Then, given a cut $\lambda \in C_\lambda$, the total cost of all cut edges in \mathcal{E}_D can be computed by

$$\sum_{x_i \in \mathcal{V}} \sum_{k=1}^n f(x_i, k)(\lambda_{k-1}(x_i) - \lambda_k(x_i)) = \sum_{x_i \in \mathcal{V}} f(x_i, \phi(x_i)), \tag{44}$$

where $\phi : \mathcal{V} \rightarrow \{1, 2, \dots, n\}$ is a labeling function:

$$\phi(x_i) = \sum_{k=0}^{n-1} \lambda_k(x_i). \tag{45}$$

We see that (44) exactly resembles the data term in the discrete Potts model (33). We can also compute the cost of all cut edges in \mathcal{E}_R by:

$$\sum_{k=1}^{n-1} \sum_{(x_i, x_j) \in \mathcal{N}} \frac{\alpha}{2} |\lambda_k(x_j) - \lambda_k(x_i)| \tag{46}$$

$$= \sum_{(x_i, x_j) \in \mathcal{N}} \frac{\alpha}{2} \left(\sum_{k=1}^{n-1} |\lambda_k(x_j) - \lambda_k(x_i)| \right) \tag{47}$$

$$= \sum_{(x_i, x_j) \in \mathcal{N}} \frac{\alpha}{2} |\phi(x_j) - \phi(x_i)|. \tag{48}$$

Combining the cost of all cut edges in \mathcal{E}_D and \mathcal{E}_R , we can see that the total cost of a cut $\lambda \in C_\lambda$ exactly equals the Potts energy (33) of a labeling function ϕ . Therefore, the energy minimization problem (33) is now converted to the min-cut problem (41).

Based on the max-flow and min-cut theorem (Papadimitriou and Steiglitz 1998, p. 117), we also know that the discrete min-cut problem (41) is equivalent to a discrete max-flow problem. A flow on a graph \mathcal{G} is a mapping $p : \mathcal{E} \rightarrow \mathbb{R}^+$ satisfying a capacity constraint:

$$p(x_i^*, x_j^*) \leq c(x_i^*, x_j^*), \quad \forall (x_i^*, x_j^*) \in \mathcal{E}, \quad (49)$$

and a flow conservation constraint:

$$\sum_{x_j^*: (x_i^*, x_j^*) \in \mathcal{E}} p(x_i^*, x_j^*) = \sum_{x_j^*: (x_j^*, x_i^*) \in \mathcal{E}} p(x_j^*, x_i^*), \quad \forall x_i^* \in \mathcal{V}^* \setminus \{s, t\}, \quad (50)$$

which says there should be a balance between the incoming and outgoing flow at each vertex. The max-flow problem was originally proposed to model the traffic flow problem (Schrijver 2002). We can view the graph \mathcal{G} as a traffic system between two cities s and t . Each vertex except s and t is an intermediate city and each edge is a railway connecting two cities. The cost assigned to each edge represents the maximum capacity of transportation. Moreover, we assume that the traffic flow in and out is equal for each intermediate city. Then, the max-flow problem aims to find the maximum amount of traffic that can be transported from s to t under the given conditions. Mathematically, we formulate the problem as follows:

$$\max_p \sum_{x_i \in V} p(s, x_i^0), \quad (51)$$

$$\text{s.t. } p(x_i^{(k-1)}, x_i^{(k)}) \leq c(x_i^{(k-1)}, x_i^{(k)}) = f(x_i, k), \quad k = 1, \dots, n, \quad (52)$$

$$p(x_i^{(k)}, x_i^{(k-1)}) < +\infty, \quad k = 1, \dots, n \quad (53)$$

$$p(x_j^{(k)}, x_i^{(k)}) \leq c(x_j^{(k)}, x_i^{(k)}) = \alpha/2, \quad (54)$$

$$\sum_{x_j^*: (x_i^*, x_j^*) \in \mathcal{E}} p(x_i^*, x_j^*) = \sum_{x_j^*: (x_j^*, x_i^*) \in \mathcal{E}} p(x_j^*, x_i^*), \quad \forall x_i^* \in \mathcal{V}^* \setminus \{s, t\}. \quad (55)$$

There is also a primal-dual relation between the min-cut and max-flow problem, which says that the maximum flow on a graph is exactly equal to the minimal cut on the graph. Furthermore, the min-cut can be obtained by first solving the max-flow and then obtaining the severed edges from the flow saturated edges. The same primal-dual relationship will be discussed later in the continuous case. Although the min-cut and max-flow problem can be solved very efficiently, the graph cut approach suffers from the drawback that the anisotropic TV regularizer does not exactly equal the boundary length exactly and is not rotation invariant. To avoid these issues, one can use continuous extensions of max-flow models and algorithms (Strang 1983; Appleton and Talbot 2003; Yuan et al. 2014; Couprie et al. 2011), which have been adopted for integer constrained variational problems in Bae et al. (2014), Pock et al. (2008), and Liu et al. (2014).

Continuous Max-Flow Formulation for Integer-Valued Potts Model

Let's first consider the continuous labeling problem, $\phi(x) : \Omega \rightarrow [l_1, l_n]$, then we can define a binary function $\lambda(x, t) : \Omega \times [l_1, l_n] \rightarrow \{0, 1\}$ using super level-set representation as

$$\lambda(x, t) = \begin{cases} 1 & \text{if } t \leq \phi(x) \text{ and } \phi(x) \neq l_n \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

For example, let $\phi(x) = \exp(-x)$, $x \in [0, 1]$ be a continuous labeling function. Then, the domain of λ is a square $[0, 1] \times [0, 1]$ and λ is the indicator function of the region $\{(x, t) | x \in [0, 1], t \leq \exp(-x)\} \setminus \{(1, 0)\}$, as shown in Fig. 2. From the definition above, we can see $\lambda(x, l_n) = 0$ and $\lambda(x, l_1) = 1$ for any $x \in \Omega$. Moreover, $\lambda(x, t_1) \geq \lambda(x, t_2)$ for any $t_1 \leq t_2$. The labeling function can also be reconstructed by

$$\phi(x) = l_1 + \int_{l_1}^{l_n} \lambda(x, t) dt. \tag{57}$$

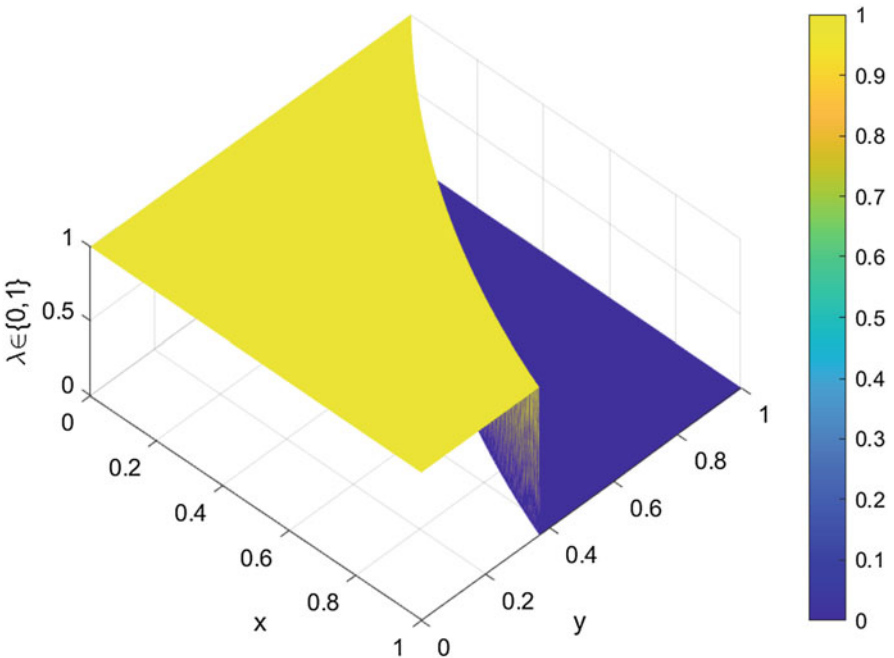


Fig. 2 Plot of $\lambda(x, t)$

We assume that for each x , $\lambda(x, t)$ is in the space of bounded variations, i.e., $\int_{l_1}^{l_n} |\partial_t \lambda(\cdot, t)| < \infty$ and $\|\lambda(\cdot, t)\|_2 < \infty$, where the partial derivative $\partial_t \lambda(x, t)$ is defined in the distributional sense. Then, the data term in (30) can be written as

$$\int_{\Omega} f(x, \phi) dx = - \int_{\Omega} \int_{l_1}^{l_n} f(x, t) \partial_t \lambda(x, t) dt dx = \int_{\Omega} \int_{l_1}^{l_n} f(x, t) |\partial_t \lambda(x, t)| dt dx, \quad (58)$$

where $\partial_t \lambda(x, t)$ satisfies

$$\int_{l_1}^{l_n} \partial_t \lambda(x, t) f(x, t) dt = -f(x, \phi(x)). \quad (59)$$

Using the generalized co-area formula (Fleming and Rishel 1960):

$$\int_{\Omega} g(x) |\nabla u(x)| dx = \int_{\mathbb{R}} \int_{u^{-1}(t)} g(x) d\mathcal{H}^1(x) dt, \quad (60)$$

where \mathcal{H}^1 denotes the one-dimensional Hausdorff measure, the regularization term in (30) is equivalent to

$$\int_{\Omega} \alpha |\nabla \phi(x)| dx = \int_{l_1}^{l_n} \alpha \mathcal{H}^1(\phi^{-1}(t)) dt = \int_{\Omega} \int_{l_1}^{l_n} \alpha |\nabla_x \lambda(x, t)| dt dx. \quad (61)$$

With (58) and (61), the following convex relaxation of (30) is considered in (Bae et al. 2010, 2014; Pock et al. 2008; Yuan et al. 2014):

$$\min_{\lambda \in D} \int_{\Omega} \int_{l_1}^{l_n} f(x, t) |\partial_t \lambda(x, t)| + \alpha |\nabla_x \lambda(x, t)| dt dx, \quad (62)$$

where $D = \{\lambda : \Omega \times [l_1, l_n] \rightarrow [0, 1] | \lambda(x, l_1) = 1, \lambda(x, l_n) = 0, \partial_t \lambda(x, t) \leq 0\}$. It is shown in Pock et al. (2008, Theorem 2) that the minimizer of (30) can be achieved by any threshold of the solution of (62). Since $\partial_t \lambda(x, t)$ is always non-positive, we can rewrite (62) as

$$\min_{\lambda \in D} \int_{\Omega} \int_{l_1}^{l_n} -f(x, t) \partial_t \lambda(x, t) + \alpha |\nabla_x \lambda(x, t)| dt dx. \quad (63)$$

If we discretize the domain along the t dimension using $t = l_1, \dots, l_n$, (63) can be approximated by

$$\min_{\lambda \in D'} \int_{\Omega} \sum_{k=1}^n f(x, l_k) (\lambda_{k-1}(x) - \lambda_k(x)) + \alpha |\nabla \lambda_k(x)| dx, \quad (64)$$

where $D' = \{\lambda_k = \lambda(x, l_k) \in L^2(\Omega) \cap BV(\Omega), k = 0, 1, \dots, n | 1 = \lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_n = 0\}$. One can see that (64) is a continuous version of the min-cut model (41). Similarly, we can also generalize the discrete max-flow problem (51) to a continuous version (Bae et al. 2010, 2014; Yuan et al. 2014):

$$\max_{p,q} \int_{\Omega} p_1(x) dx, \tag{65}$$

$$\text{s.t. } q_k \in C_{\alpha}, k = 1, \dots, n - 1, \tag{66}$$

$$p_k(x) \leq f(x, l_k), k = 1, \dots, n, \tag{67}$$

$$\text{div}(q_k) - p_k + p_{k+1} = 0, k = 1, \dots, n - 1. \tag{68}$$

where $C_{\alpha} = \{q : \Omega \rightarrow \mathbb{R}^2 | |q(x)| \leq \alpha, q \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\}$. In this setting, $p_k(x)$ simulates the flow from $x_i^{(k-1)}$ to $x_i^{(k)}$. $q_k(x) = (q_1^k(x), q_2^k(x))$ simulates the flow from x_i^k to its neighbors $\mathcal{N}_i^{(k)} = \{x_j^{(k)} | x_j \in \mathcal{N}_i\}$ along the first and second dimension. It can be shown that the dual problem of (65) is the continuous min-cut problem (64) by introducing the Lagrangian multipliers λ_k :

$$\begin{aligned} & \min_{\lambda_k} \max_{\substack{p_k(x) \leq f(x, l_k) \\ q_k \in C_{\alpha}}} \int_{\Omega} p_1(x) + \sum_{k=1}^{n-1} \lambda_k (\text{div} q_k - p_k + p_{k+1}) dx \\ &= \min_{\lambda_i} \max_{\substack{p_k(x) \leq f(x, l_k) \\ q_k \in C_{\alpha}}} \int_{\Omega} \sum_{k=1}^n p_k (\lambda_{k-1} - \lambda_k) + \sum_{k=1}^n \lambda_k \text{div}(q_k) dx. \end{aligned} \tag{69}$$

Since p_k is only bounded above, if $(\lambda_{k-1} - \lambda_k) < 0$, the energy can go to infinity. Therefore, the optimal solution must satisfy $\lambda \in D'$. Using the fact that

$$\int_{\Omega} \alpha |\nabla \lambda_k| dx = \max_{q_k \in C_{\alpha}} \int_{\Omega} \lambda_k \text{div}(q_k) dx, \tag{70}$$

we have (69) is equivalent to

$$\min_{\lambda \in D'} \int_{\Omega} \sum_{k=1}^n f(x, l_k) (\lambda_{k-1}(x) - \lambda_k(x)) + \alpha \sum_{k=1}^n |\nabla \lambda_k| dx, \tag{71}$$

which is exactly the continuous min-cut problem. Another equivalent continuous min-cut formulation of (63) is:

$$\min_{\lambda \in D} \int_{\Omega} \int_{l_1}^{l_n} \partial_t f(x, t) \lambda(x, t) + \alpha |\nabla_x \lambda(x, t)| dt dx, \tag{72}$$

which can be derived by integration by part. After discretizing along t dimension, we can have

$$\min_{\lambda \in D} \sum_{i=1}^{n-1} \int_{\Omega} (f(x, l_{i+1}) - f(x, l_i)) \lambda_i(x) + \alpha |\nabla \lambda_i(x)| dx. \tag{73}$$

It is shown in Liu et al. (2014, Proposition 1) that (73) is the dual of the following continuous max-flow problem:

$$\max_{f_t^k, f_s^k, f^{k,k+1}, \mathbf{g}^k} \sum_{k=1}^{n-1} \int_{\Omega} f_t^k(x) dx, \tag{74}$$

$$\text{s.t. } f_s^k(x) \leq f(x, l_{k+1}), f_t^k(x) \leq f(x, l_k), f^{k,k+1}(x) \geq c, |\mathbf{g}^k(x)| \leq \alpha, \tag{75}$$

$$f_s^k(x) - f_t^k(x) + f^{k-1,k}(x) - f^{k,k+1}(x) - \text{div}(\mathbf{g}^k(x)) = 0, \tag{76}$$

where $k = 0, 1, \dots, n, f^{-1,0} = f^{n,n+1} = 0$, and $f(x, l_0), f(x, l_{n+1})$, and c are set to be very large numbers. By introducing $n - 1$ Lagrangian multipliers λ_k for the flow conservation constraints, we can obtain an equivalent dual problem:

$$\min_{\lambda_i} \int_{\Omega} f(x, l_1) dx + \sum_{k=1}^{n-1} \int_{\Omega} (\lambda_{k+1}(x) - \lambda_k(x)) f(x, l_k) + \alpha |\nabla \lambda_k(x)| dx, \tag{77}$$

$$\text{s.t. } 1 = \lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{n-1} \geq \lambda_n = 0. \tag{78}$$

Both continuous max-flow problem (65) and (74) can be solved by the augmented Lagrangian method (ALM) with alternating direction method of multipliers (ADMM).

Numerical Algorithms for the Integer-Valued Continuous Max-Flow Problems

The Augmented Lagrangian functional of (65) is

$$\begin{aligned} \min_{\lambda_k} \max_{\substack{p_k(x) \leq f(x, l_k) \\ |q_k| \leq \alpha}} L(\lambda, p, q) &= \int_{\Omega} p_1 + \sum_{k=1}^{n-1} \lambda_i (\text{div} q_k - p_k + p_{k+1}) \\ &\quad - \sum_{k=1}^{n-1} \frac{c}{2} |\text{div} q_k - p_k + p_{k+1}|^2 dx. \end{aligned} \tag{79}$$

Algorithm 1 ADMM for Continuous Max-flow (65)

```

1: Initialize  $p_i^0$  and  $q_i^0$ .
2: while stopping criterion is not satisfied do
3:   for  $k = 1, \dots, n - 1$  do
4:     if  $k == 1$  then
5:        $\tilde{p}_1^{\tau+1} = \arg \max_{p_1(x) \leq f(x, l_1)} -\lambda_1 p_1 + p_1 - \frac{c}{2} |\operatorname{div} q_1^\tau - p_1 + p_2^\tau|^2$ 
6:     else if  $k == n$  then
7:        $\tilde{p}_n^{\tau+1} = \arg \max_{p_n(x) \leq f(x, l_n)} \lambda_{n-1} p_n - \frac{c}{2} |\operatorname{div} q_{n-1}^{\tau+1} - p_{n-1}^{\tau+1} + p_n|^2$ 
8:     else
9:        $\tilde{p}_k^{\tau+1} = \arg \max_{p_k(x) \leq f(x, l_k)} -\lambda_k p_k + \lambda_{k-1} p_k - \frac{c}{2} |\operatorname{div} q_k^\tau - p_k + p_{k+1}^\tau|^2 - \frac{c}{2} |\operatorname{div} q_{k-1}^{\tau+1} - p_{k-1}^{\tau+1} + p_k|^2$ 
10:    end if
11:    if  $k \leq n - 1$  then
12:       $q_k^{\tau+1} = \arg \max_{|q_k| \leq \alpha} \lambda - \frac{c}{2} |\operatorname{div} q_k - \tilde{p}_k^{\tau+1} + p_{k+1}^\tau|^2$ 
13:    end if
14:    run step 5 or 7 or 9 again with  $q_k^\tau$  replaced by  $q_k^{\tau+1}$  to obtain  $p_k^{\tau+1}$ .
15:    if  $k \leq n - 1$  then
16:      update the multipliers  $\lambda_k$ :  $\lambda_k^{\tau+1} = \lambda_k^\tau - c(\operatorname{div} q_k^{\tau+1} - p_k^{\tau+1} + p_{k+1}^{\tau+1})$ 
17:    end if
18:  end for
19: end while

```

The authors of Bae et al. (2014, Algorithm 1) used ADMM to solve this model. We summarize their algorithm in Algorithm 1. In steps 5, 7, and 9, the objective functional is a quadratic polynomial with convex constraints, so the solution can be found explicitly. For step 12, an inexact solution can be obtained by doing gradient ascent with projection. One can refer to Bae et al. (2014, Algorithm 1) for more detail. A similar ADMM algorithm for (74) is proposed in Liu et al. (2014, Algorithm 1). The augmented Lagrangian functional of (74) can be written as

$$\min_{\lambda_k} \max_{f_t^k, f_s^k, \mathbf{g}} \sum_{k=1}^{n-1} \int_{\Omega} f_t^k(x) + \lambda_i(x)(f_s^k(x) - f_t^k(x) - \operatorname{div}(\mathbf{g}^k)) - \frac{r}{2} (f_s^k(x) - f_t^k(x) - \operatorname{div}(\mathbf{g}^k))^2 dx, \tag{80}$$

$$\text{s.t. } 1 = \lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{n-1} \geq \lambda_n = 0, \tag{81}$$

$$f_t^k(x) \leq f(x, l_i), \quad f_s^k(x) \leq f(x, l_{k+1}) \text{ for } k = 1, 2, \dots, n - 1, \tag{82}$$

$$|\mathbf{g}^k| \leq \alpha, \text{ for } k = 1, 2, \dots, n. \tag{83}$$

Then each variable can be updated iteratively. The ADMM algorithm for (80) is described in Algorithm 2

Algorithm 2 ADMM for Continuous Max-flow (74)

-
- 1: Initialize $(f_t^k)^0, (f_s^k)^0, (g^k)^0$ and $(\lambda_k)^0$
 - 2: **while** stopping criterion is not satisfied **do**
 - 3: **for** $k = 1, \dots, n - 1$ **do**
 - 4: $(f_t^k)^{(\tau+1)} = \min \left\{ (f_t^k)^{(\tau)} - \operatorname{div}(\mathbf{g}^k)^{(\tau)} + \frac{1 - (\lambda_k)^{(\tau)}}{r}, f(x, l_k) \right\}$
 - 5: $(f_s^k)^{(\tau+1)} = \min \left\{ (f_s^k)^{(\tau)} + \operatorname{div}(\mathbf{g}^k)^{(\tau)} + \frac{(\lambda_k)^{(\tau)}}{r}, f(x, l_{k+1}) \right\}$
 - 6: $(\mathbf{g}^k)^{(\tau+1)} = \arg \min_{\|\mathbf{g}^k\| \leq \alpha} \left\| -\frac{(\lambda_k)^{(\tau)}}{r} + (f_s^k)^{(\tau+1)} - (f_t^k)^{(\tau+1)} - \operatorname{div}(\mathbf{g}^k) \right\|^2$, which can be solved by the Chambolle's projection algorithm (Chambolle 2004)
 - 7: $(\lambda_i)^{k+1} = \Pi_{\mathbb{B}} \left((\lambda_i)^{(\tau)} - \tau \left((f_s^k)^{(\tau+1)} - (f_t^k)^{(\tau+1)} - \operatorname{div}(\mathbf{g}^k)^{(\tau+1)} \right) \right)$, where $\Pi_{\mathbb{B}}$ is the projection onto $\mathbb{B} = \{(\lambda_1, \dots, \lambda_{n-1}) \mid 1 \geq \lambda_1 \geq \dots \geq \lambda_{n-1} \geq 0\}$. This can be done by the recursive algorithm in Chambolle et al. (2012)
 - 8: **end for**
 - 9: **end while**
-

Representation by Simplex-Constrained Vector Functions

We will use a different representation for the Potts model in this section. If we set $v_k(x)$ to be the indicator function of the k -th phase and denote $\mathbf{v} \in \Delta^n$ as $\mathbf{v} = (v_1, v_2, \dots, v_n)$ where Δ^n is defined as (18), then the Potts model can be rewritten as:

$$\min_{\mathbf{v} \in \Delta^n} \sum_{k=1}^n \int_{\Omega} f_k(x) v_k(x) dx + \alpha \sum_{k=1}^n \int_{\Omega} |\nabla v_k(x)| dx. \quad (84)$$

Notice that the problem (84) is a non-convex optimization problem. The following convex relaxation has been considered in many publications:

$$\min_{\mathbf{v} \in \tilde{\Delta}^n} \sum_{k=1}^n \int_{\Omega} f_k(x) v_k(x) dx + \alpha \sum_{k=1}^n \int_{\Omega} |\nabla v_k(x)| dx, \quad (85)$$

where

$$\tilde{\Delta}^n = \{ \mathbf{v}(x) = (v_1(x), v_2(x), \dots, v_n(x)) \mid v_k(x) \in [0, 1], \sum_{k=1}^n v_k(x) = 1 \}.$$

Later in this section, we shall show that (84) is a min-cut problem and it is equivalent to a max-flow problem. The max-flow is convex. The dual of the max-flow problem is exactly the convex relaxed problem (85).

Algorithm 3 First-order primal-dual algorithm for image segmentation (Chambolle and Pock 2011)

- 1: Initialize v^0 and q_k^0 .
- 2: **while** stopping criterion is not satisfied **do**
- 3: update each q_k by

$$\begin{aligned}
 q_k^{\tau+1} &= \arg \max_{q_k \in C_\alpha} \int_{\Omega} -\nabla v_k^\tau(x) \cdot q_k(x) - \frac{1}{2\sigma} \|q_k(x) - q_k^\tau(x)\|_2^2 dx \\
 &= \arg \min_{q_k \in C_\alpha} \int_{\Omega} \|q_k(x) - (q_k^\tau(x) - \sigma \nabla v_k^\tau(x))\|_2^2 dx \\
 &= \Pi_{C_\alpha}(q_k^\tau - \sigma \nabla v_k^\tau)
 \end{aligned}$$

- 4: update v by

$$\begin{aligned}
 v^{\tau+1} &= \arg \min_{v \in \tilde{\Delta}^n} \int_{\Omega} \sum_{i=1}^n v_k(x)(f_k(x) + \operatorname{div}(q_k^{\tau+1})) + \sum_{i=1}^n \frac{1}{2\tau} \|v_k(x) - v_k^\tau(x)\|_2^2 dx \\
 &= \Pi_{\tilde{\Delta}^n}(v^\tau - \tau(f + \operatorname{div}(q^{\tau+1})))
 \end{aligned}$$

- 5: **end while**
-

Primal-Dual Formulation for Simplex-Constrained Potts Model

Applying (70) to (85), we can have

$$\min_{v \in \tilde{\Delta}^n} \max_{q_k \in C_\alpha} \sum_{k=1}^n \int_{\Omega} v_k(x)(f_k(x) + \operatorname{div}(q_k)) dx. \tag{86}$$

Since $-\nabla$ is the dual operator of div , we also have

$$\int_{\Omega} v_k(x) \operatorname{div}(q_k(x)) dx = \int_{\Omega} -\nabla v_k(x) \cdot q_k(x) dx, \quad q_k \in C_\alpha. \tag{87}$$

Model (86) is often called as the primal-dual model and is well studied in the literature (Chambolle and Pock 2011; Esser et al. 2010; Zhu and Chan 2008). To solve this primal-dual model, we can apply the primal-dual algorithm by Chambolle and Pock (2011), Esser et al. (2010), and Wu and Tai (2010) which is summarized in Algorithm 3. This algorithm can be viewed as a primal-dual proximal point method for (86). One can see that this algorithm consists of only two steps of projection. The first projection is just a simple element-wise projection and the second projection can also be computed efficiently by Michelot (1986). The convergence of this algorithm is also proved in Chambolle and Pock (2011, Theorem 1).

Dual Formulation for Simplex-Constrained Potts Model

Based on the minimax theorem (Ekeland and Temam 1999, Chapter 6, Proposition 2.4), the min and max in (86) can be interchanged, and there exists at least one saddle point. By observing that

$$\min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \int_{\Omega} v_k(x) (f_k(x) + \operatorname{div}(q_k(x))) dx = \int_{\Omega} \min_{k=1, \dots, n} (f_k(x) + \operatorname{div}(q_k(x))) dx,$$

we can derive the dual model as in (Bae et al. 2011, p. 7):

$$\max_{q_k \in C_{\alpha}} \min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \int_{\Omega} v_k(x) (f_k(x) + \operatorname{div}(q_k(x))) dx \quad (88)$$

$$= \max_{q_k \in C_{\alpha}} \int_{\Omega} \min_{k=1, \dots, n} (f_k(x) + \operatorname{div}(q_k(x))) dx. \quad (89)$$

Suppose q_k^* is the optimal solution of the dual model (89), then the optimal primal variable can be recovered by minimizing the primal-dual energy (86) with q_k fixed:

$$v^* = \arg \min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \int_{\Omega} v_k(x) (f_k(x) + \operatorname{div}(q_k^*(x))) dx. \quad (90)$$

Then it is easy to derive that

$$v_k^*(x) = \begin{cases} 1 & \text{if } k = \arg \min_{k=1, \dots, n} (f_k(x) + \operatorname{div}(q_k^*(x))) \\ 0 & \text{otherwise} \end{cases}. \quad (91)$$

Provided the minimizer $\arg \min_{k=1, \dots, n} (f_k(x) + \operatorname{div}(q_k^*(x)))$ is unique at each point x , it was proved by the minimax theorem that v^* is a global minimizer of the non-convex Potts model (84) (Bae et al. 2011, Theorem 1). It was also shown that an exact global minimizer can be generated in case $\arg \min_{k=1, \dots, n} (f_k(x) + \operatorname{div}(q_k^*(x)))$ for some points have 2 non-unique minimizers (Bae et al. 2011, Proposition 2). In case of three or more non-unique minimizers, it is still an open question whether a binary global minimizer can be generated from the dual solution. However, in practice the minimizer tends to be unique for the vast majority of points.

To overcome the non-smoothness of the dual model (89), Bae et al. (2011) proposed a smooth approximation. Considering the log-sum exponential function

$$f_s(x) = s \log \sum_{k=1}^n e^{x_k/s}, \quad (92)$$

where $s > 0$. When s goes to infinity, $f_s(x)$ will converge to $\max_{k=1,\dots,n}(x_k)$. Then, we have

$$\max_{q_k \in C_\alpha} \int_{\Omega} \min_{k=1,\dots,n} (f_k(x) + \operatorname{div}(q_k(x))) \, dx \tag{93}$$

$$= \max_{q_k \in C_\alpha} \int_{\Omega} - \max_{k=1,\dots,n} (-f_k(x) - \operatorname{div}(q_k(x))) \, dx \tag{94}$$

$$\approx \max_{q_k \in C_\alpha} -s \int_{\Omega} \log \sum_{k=1}^n \exp \left(\frac{-f_k(x) - \operatorname{div}(q_k(x))}{s} \right) \, dx. \tag{95}$$

By using the identity

$$\log \sum_{k=1}^n \mu_k e^{h_k} = \max_{u \in \tilde{\Delta}^n} \left\{ \langle u, h \rangle - \sum_{k=1}^n u_k \log \frac{u_k}{\mu_k} \right\}, \tag{96}$$

we can see that (95) is equivalent to a new primal-dual formulation

$$\begin{aligned} & \max_{q_k \in C_\alpha} - \max_{v \in \tilde{\Delta}^n} \int_{\Omega} \left\{ \sum_{k=1}^n v_k(x) (-f_k(x) - \operatorname{div}(q_k(x))) - s \sum_{k=1}^n v_k(x) \log v_k(x) \right\} \, dx \\ &= \max_{q_k \in C_\alpha} \min_{v \in \tilde{\Delta}^n} \int_{\Omega} \left\{ \sum_{k=1}^n v_k(x) (f_k(x) + \operatorname{div}(q_k(x))) + s \sum_{k=1}^n v_k(x) \log v_k(x) \right\} \, dx, \end{aligned} \tag{97}$$

which is exactly the original primal-dual model (86) plus an entropy penalization. In Bae et al. (2011, Algorithm 1), the authors used a proximal forward-backward splitting (PFBS) algorithm to solve the smoothed primal-dual model (Algorithm 4). One can observe that the Algorithm 4 is a special case of the first-order primal-dual algorithm (Algorithm 3) by choosing $\sigma = \infty$ and $\tau = \delta$. Actually, this algorithm can also be viewed as performing the expectation maximization (EM) methods on the smoothed dual model (95). In Bae et al. (2011), the authors also interpreted the first v update as the expectation step which is often called the softmax activation function and the q update as the maximization step.

Continuous Max-Flow Formulation for Simplex-Constrained Potts Model

Similarly to the integer-valued labeling case, we can also derive a continuous max-flow model corresponding to the simplex-constrained Potts model (84), c.f. Yuan et al. (2010):

Algorithm 4 PFBS for the smoothed primal-dual model (97)

- 1: Initialize v^0 and q_k^0 .
- 2: **while** stopping criterion is not satisfied **do**
- 3: update v by

$$v^{\tau+1} = \arg \min_{v \in \bar{\Delta}^n} \int_{\Omega} \left\{ \sum_{k=1}^n v_k(x) (f_k(x) + \operatorname{div}(q_k^{\tau}(x))) + s \sum_{k=1}^n v_k(x) \log v_k(x) \right\} dx$$

$$\Rightarrow v_k^{\tau+1} = \exp \left(-\frac{f_k + \operatorname{div}(q_k^{\tau})}{s} \right) / \sum_{k=1}^n \exp \left(-\frac{f_k + \operatorname{div}(q_k^{\tau})}{s} \right), \quad k = 1, \dots, n.$$

which can be derived from the KKT condition.

- 4: update each q_k by

$$q_k^{\tau+1} = \arg \max_{q_k \in C_{\alpha}} \int_{\Omega} v_k^{\tau+1}(x) (f_k(x) + \operatorname{div}(q_k(x))) + \frac{1}{2\delta} \|q_k(x) - q_k^{\tau}(x)\|_2^2 dx$$

$$= \arg \max_{q_k \in C_{\alpha}} \int_{\Omega} -\nabla v_k^{\tau+1}(x) \cdot q_k(x) + \frac{1}{2\delta} \|q_k(x) - q_k^{\tau}(x)\|_2^2 dx \quad (98)$$

$$= \Pi_{C_{\alpha}}(q_k^{\tau} + \delta \nabla v_k^{\tau+1}).$$

- 5: **end while**

$$\max_{\lambda} \int_{\Omega} \lambda(x) dx, \quad (99)$$

$$\text{s.t. } h_k \leq f_k, q_k \in C_{\alpha}, \quad k = 1, \dots, n, \quad (100)$$

$$\operatorname{div}(q_k) - \lambda + h_k = 0, \quad k = 1, \dots, n. \quad (101)$$

By introducing the Lagrangian multipliers v_k , we can derive an equivalent primal-dual formulation of (99) as:

$$\min_{v_k} \max_{\substack{\lambda \\ h_k \leq f_k \\ q_k \in C_{\alpha}}} \int_{\Omega} \lambda(x) + \sum_{k=1}^n (\operatorname{div}(q_k(x)) - \lambda(x) + h_k(x)) v_k(x) dx \quad (102)$$

$$= \min_{v_k} \max_{\substack{\lambda \\ h_k \leq f_k \\ q_k \in C_{\alpha}}} \int_{\Omega} \left(1 - \sum_{k=1}^n v_k(x) \right) \lambda dx + \sum_{k=1}^n \int_{\Omega} h_k(x) v_k(x) dx$$

$$+ \sum_{k=1}^n \int_{\Omega} v_k(x) \operatorname{div}(q_k(x)) dx. \quad (103)$$

Since h_k is unbounded below, we should have the optimal v_k is non-negative. Otherwise, the energy functional will go to infinity. What's more, $(1 - \sum_{k=1}^n v_k)$ should also be zero, because λ is unbounded. Therefore, the above primal-dual formulation is equivalent to

$$\min_{v_k} \sum_{k=1}^n \int_{\Omega} f_k(x) v_k(x) + \alpha |\nabla v_k(x)| dx, \tag{104}$$

$$\text{s.t. } v \in \tilde{\Delta}^n, \tag{105}$$

which is exactly the Potts model (85). The inequality (70) is also used here. In Yuan et al. (2010, Algorithm 1), the authors proposed an ADMM-based algorithm to solve model (99). The augmented Lagrangian functional of (99) is

$$\begin{aligned} & \int_{\Omega} \lambda(x) + \sum_{k=1}^n v_k(x) (\text{div}(q_k(x)) - \lambda(x) + h_k(x)) \\ & - \frac{\rho}{2} \sum_{k=1}^n |\text{div}(q_k(x)) - \lambda(x) + h_k(x)|^2 dx. \end{aligned} \tag{106}$$

Then the procedure goes like Algorithm 5 where the last three updates have a simple closed-form solution and the first q_k update can be solve by Chambolle's semi-implicit gradient descent algorithm (Chambolle 2004).

Algorithm 5 ADMM for the continuous max-flow model (99)

- 1: Initialize v^0, q_k^0, h_k^0 and λ_k^0 .
- 2: **while** stopping criterion is not satisfied **do**
- 3: update each q_k by

$$q_k^{\tau+1} = \arg \max_{q_k \in C_{\alpha}} - \frac{\rho}{2} \|\text{div}(q_k) + h_k^{\tau} - \lambda^{\tau} - v_k^{\tau} / \rho\|_2^2.$$

- 4: update each h_k by

$$h_k^{\tau+1} = \arg \max_{h_k \leq f_k} - \frac{\rho}{2} \|h_k + \text{div}(q_k^{\tau+1}) - \lambda^{\tau} - v_k^{\tau} / \rho\|_2^2.$$

- 5: update λ by

$$\lambda^{\tau+1} = \arg \max_{\lambda} \int_{\Omega} \lambda(x) dx - \frac{\rho}{2} \|h_k^{\tau+1} + \text{div}(q_k^{\tau+1}) - \lambda - v_k^{\tau} / \rho\|_2^2.$$

- 6: update each v_k by

$$v_k^{\tau+1} = v_k^{\tau} - \rho (\text{div}(q_k^{\tau+1}) + h_k^{\tau+1} - \lambda^{\tau+1}).$$

- 7: **end while**
-

Representation by Overlapping Functions

Potts Model with Overlapping Binary Functions Representation

The third way of representing multiphase is the binary version of the level-set framework in Vese and Chan (2002). For each $k \in \{1, \dots, n = 2^m\}$, let a^k denote the m -digit binary representation of k . For example, if $m = 3$ and $k = 2$, then $a^k = (0, 1, 0)$. Let $\phi = (\phi^1, \dots, \phi^m) \in \{0, 1\}^m$, then the general model in Vese and Chan (2002) can then be written as:

$$\min_{\phi^j \in \{0,1\}} \int_{\Omega} \sum_{k=1}^n I_{a^k}(\phi(x), x) f_k(x) dx + \alpha \sum_{j=1}^m \int_{\Omega} |\nabla \phi^j| dx, \tag{107}$$

where $f_k(x)$ is the point-wise cost of assigning x to region k and

$$I_{a^k}(\phi(x), x) = \begin{cases} 1 & \text{if } \phi(x) = a^k \text{ for } k = 1, \dots, n \\ 0 & \text{otherwise} \end{cases}. \tag{108}$$

Notice that the regularization term does not correspond exactly to the original Potts regularizer because some boundaries are counted multiple times. It is also possible to represent a number of n regions which is not a power of 2 by choosing m as the smallest integer such that $n < 2^m$ and setting $f_k = \infty$ for the last $2^m - n$ indices.

Extension to More General Cases

A natural extension of the model is to represent the image partition in terms of overlapping integer-valued labeling function $\phi = (\phi^1, \dots, \phi^m)$ where $\phi^j \in \mathcal{L}_j = \{0, \dots, N_j - 1\}$. The total number of phases can be represented is $n = \prod_{j=1}^m N_j$. Let $\{a^k\}_{k=1}^n$ denote an enumeration of all feasible values for ϕ , i.e., for each $k = 1, \dots, n$,

$$a^k = (a_1^k \dots a_m^k) \tag{109}$$

such that $a_j^k \in \mathcal{L}_k$ for $j = 1, \dots, m$. Then region Ω_k can be encoded as

$$\Omega_k = \{x \in \Omega \text{ s.t. } \phi(x) = a^k\}, \quad k = 1, \dots, n. \tag{110}$$

However, this encoding is not unique, as the enumeration $\{a^k\}_{k=1}^n$ can be reordered in any way. There are $n!$ such reorderings and they can be formulated generally using a permutation matrix P as follows

$$[a^1 \dots a^n] \leftarrow [a^1 \dots a^n] \cdot P \tag{111}$$

The choice of permutation may have an effect on the quality of the relaxation. For instance, Bae and Tai (2015) showed that a particular permutation of the four-region model was crucial for producing exact global minimizers of the original problem. One possible choice can be given by

$$k = \sum_{j=1}^m a_j^k \prod_{i=j+1}^m \mathcal{L}_i. \tag{112}$$

For example, if $m = 3$, $\mathcal{L}_1 = 2$, $\mathcal{L}_2 = 3$, $\mathcal{L}_3 = 3$, and $\phi(x) = (1, 2, 3)$, then it is corresponding to region $1 \times 2 \times 3 + 2 \times 3 + 3 = 15$ and the corresponding data term is $f_{15}(x)$. By defining function $f : \mathcal{L}_1 \times \dots \times \mathcal{L}_m \times \Omega \mapsto \mathbb{R}$ as,

$$f(\phi(x), x) = \begin{cases} f_k(x), & \text{if } \phi(x) = a^k, \quad k = 1, \dots, n \\ +\infty, & \text{otherwise,} \end{cases} \tag{113}$$

we can rewrite the Potts model in terms of ϕ as:

$$\min_{\phi} \int_{\Omega} f(\phi(x), x) dx + \alpha \sum_{j=1}^m \int_{\Omega} |\nabla \phi^j(x)| dx. \tag{114}$$

In case $N_1 = \dots = N_m = 2$, the model (114) reduces to the Chan-Vese model (107). Notice that due to the separable form of the regularizer, some boundaries will be counted more than once. Using the similar idea, we can also extend the vector-valued representation in the same way. Let $v^j = (v_1^j, \dots, v_{N_j}^j) : \Omega \mapsto \mathbb{R}^{N_j}$ be a set of unit vector functions which satisfy

$$\sum_{i=1}^{N_j} v_i^j(x) = 1, \quad v_i^j(x) \in \{0, 1\}, \quad \text{for } i = 1, \dots, N_j \text{ and } \forall x \in \Omega. \tag{115}$$

The function $v = (v^1, \dots, v^k)$ with the above constraint can also represent $n = \prod_{k=1}^m N_k$ regions. Let $\{a^k\} \in \mathbf{R}^{N_1} \times \dots \times \mathbf{R}^{N_m}$ denotes an enumeration of all possible v values. Then the data term f can be defined as

$$f(v(x), x) = \begin{cases} f_k(x), & \text{if } v = a^k, \\ +\infty, & \text{otherwise.} \end{cases} \tag{116}$$

The general segmentation model can then be formulated as

$$\min_v \int_{\Omega} f(v(x), x) + \alpha \sum_{j=1}^m \sum_{i=1}^{N_j} \int_{\Omega} |\nabla v_i^j|. \quad (117)$$

An advantage of this representation compared to (114) is that the regularization term more closely resembles the Potts regularization term. What's more, it exactly represents it for boundaries where only one of the v^j changes. For instance, if $m=2$, the boundaries will be counted at most twice, with the majority being counted once.

Convex Relaxation via Convex Envelope for Overlapping Representation

Notice that the models (114) and (117) have non-convex data term. To obtain a convex relaxation, the convex envelop technique is considered in Bae et al. (2013). Let $J(x) : X \rightarrow \mathbb{R}$ be a function defined on an inner product space X . Then the Fenchel conjugate of $J(x)$ is defined as

$$J^*(y) = \sup_{x \in X} \langle y, x \rangle - J(x), \quad (118)$$

and the biconjugate is defined as

$$J^{**}(x) = \sup_{y \in X} \langle x, y \rangle - J^*(y). \quad (119)$$

It can be shown that J^{**} is the largest convex and lower semi-continuous function such that $J^{**} \leq J$. What's more, J^{**} has the same global minimum with J for any proper function J . In Bae et al. (2013, p. 6), the authors construct a convex relaxation of the data term in (117) by computing its convex envelop with x fixed:

$$f^*(p(x), x) = \sup_{q \in \mathbb{R}^m} \langle p, q \rangle - f(q(x), x) \quad (120)$$

$$= \sup_{q \in \mathbb{R}^m} \sum_{j=1}^m p^j(x) q^j(x) - f(q(x), x) \quad (121)$$

$$= \max_{q \in \{a^k\}_{k=1}^n} \sum_{j=1}^m p^j(x) q^j(x) - f(q(x), x), \quad (122)$$

and

$$f^{**}(\phi(x), x) = \sup_{p \in \mathbb{R}^m} \langle \phi, p \rangle - f^*(p(x), x) \tag{123}$$

$$= \sup_{p \in \mathbb{R}^m} \left\{ \sum_{j=1}^m \phi^j(x) p^j(x) + \min_{q \in \{a^k\}_{k=1}^n} \sum_{j=1}^m -p^j(x) q^j(x) + f(q(x), x) \right\} \tag{124}$$

$$= \sup_{p \in \mathbb{R}^m, p_0 \in \mathbb{R}} \left\{ \sum_{j=1}^m \phi^j(x) p^j(x) + p_0(x) \right\} \tag{125}$$

$$\text{with } p_0(x) \leq \sum_{j=1}^m -p^j(x) q^j(x) + f(q(x), x), \text{ for any } q \in \{a^k\}_{k=1}^n. \tag{126}$$

After adding the edge force term, the relaxed model is then written as:

$$\min_{\phi} \max_{p, p_0} \int_{\Omega} p_0(x) + \sum_{j=1}^m \phi^j(x) p^j(x) dx + \alpha \sum_{j=1}^m \int_{\Omega} |\nabla \phi^j|, \tag{127}$$

$$\text{s.t. } p_0(x) + \sum_{j=1}^m u^j(x) p^j(x) \leq f(u(x), x), \quad \forall u(x) \in \{a^k\}_{k=1}^n, \quad \forall x \in \Omega. \tag{128}$$

We want an integral solution ϕ^1, \dots, ϕ^m to the minimization problem (127). However, it cannot in general be expected that the solution is integral at every point. Therefore, we apply a thresholding procedure with parameter $t \in (0, 1]$ as follows

$$(\phi^j)^t(x) = \begin{cases} \lfloor \phi^j \rfloor, & \text{if } \phi^j(x) < \lfloor \phi^j(x) \rfloor + t \\ \lceil \phi^j \rceil, & \text{otherwise} \end{cases} \quad j = 1, \dots, m \tag{129}$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and ceiling functions, respectively. If the constraint set is binary, i.e., $N_1 = \dots = N_m = 2$, this corresponds to the standard thresholding procedure in Chan et al. (2006). In the same way, one can also derive a similar convex relaxation for the model (117) as Bae et al. (2013, p. 7).

The solutions obtained by thresholding are in general not exact but very good approximations to the global minimizer. In case of four regions, it was proven in Bae and Tai (2015) that the thresholding produces an exact global minimizer under some mild conditions on the data term.

Numerical Algorithms for Relaxed Potts Model via Overlapping Representation

Define the set D as

$$\left\{ p = (p_0, \dots, p_m) \mid p_0(x) + \sum_{j=1}^m \phi^j(x) p^j(x) \leq f(u(x), x), \text{ for } \phi(x) \in \{a^k\}_{k=1}^n \text{ and } x \in \Omega \right\}. \quad (130)$$

By using the inequality (70), the relaxed model (127) can be written as (Bae et al. 2013):

$$\min_{\phi} \max_{q_i \in C_{\alpha}, p \in D} \int_{\Omega} p_0(x) + \sum_{j=1}^m \phi^j(x) (p^j(x) + \operatorname{div}(q^j(x))) dx. \quad (131)$$

Notice that this model can be viewed as a primal-dual formulation of the following problem by introducing ϕ^j as Lagrangian multipliers:

$$\max_{q^j \in C_{\alpha}, p \in D} \int_{\Omega} p_0(x) dx, \quad (132)$$

$$\text{s.t. } p^j + \operatorname{div}(q^j) = 0, \text{ for } j = 1, \dots, m. \quad (133)$$

Therefore, we can write the augmented Lagrangian functional as

$$L(p, q, \phi) = \int_{\Omega} p_0 + \sum_{j=1}^m \phi^j (p^j + \operatorname{div}(q^j)) - \frac{\rho}{2} \|p^j + \operatorname{div}(q^j)\|^2 dx, \quad (134)$$

Then we can apply the ADMM algorithm to solve it (Bae et al. 2013, p. 10). In Algorithm 6, the first update involves computing the projection onto D , which can be approximated by Dykstra's algorithm (Boyle and Dykstra 1986). The second update can be solved by the Chambolle's algorithm (Chambolle 2004).

A Continuous Max-Flow Approach for 4-Phase Overlapping Binary Representation

Though formulating the model (107) as a discrete graph cut problem like is difficult in general, Bae and Tai (2009a, 2015) constructed a special graph to simulate the Potts energy for a 4-phase segmentation model with two overlapping binary functions. Suppose the two binary functions are denoted as $\phi^1(x) \in \{0, 1\}$ and $\phi^2(x) \in \{0, 1\}$, and they satisfy

Algorithm 6 ADMM for the relaxed Potts model (127)

- 1: Initialize $(\phi^j)^0, (p^j)^0$ and $(q^j)^0$ for $j = 1, \dots, m$.
- 2: **while** stopping criterion is not satisfied **do**
- 3: update each p by

$$p^{\tau+1} = \arg \max_{p \in D} L(p, q^\tau, (\phi)^\tau)$$

- 4: update each q^j by

$$(\phi^j)^{(k+1)} = \arg \max_{q^j \in C_\alpha} \int_{\Omega} (\phi^j)^{(\tau)} \operatorname{div}(q^j) - \frac{\rho}{2} \|(p^j)^{(\tau+1)} + \operatorname{div}(q^j)\|^2 dx$$

- 5: update each ϕ^j by

$$(\phi^j)^{(\tau+1)} = (\phi^j)^{(\tau)} - \rho((p^j)^{(\tau+1)} + \operatorname{div}((q^j)^{(\tau+1)}))$$

- 6: **end while**
-

$$(\phi^1(x), \phi^2(x)) = \begin{cases} (1, 0), & x \in \Omega_1 \\ (1, 1), & x \in \Omega_2 \\ (0, 0), & x \in \Omega_3 \\ (0, 1), & x \in \Omega_4 \end{cases}. \tag{135}$$

It is shown in Bae and Tai (2015, section 3.4) that this label assignment is crucial for obtaining the exact solution of (107). Then, the 4-phase Potts model can be written as:

$$\begin{aligned} \min_{\phi \in \{0,1\}} \int_{\Omega} & \phi^1 \phi^2 f_2 + \phi^1 (1 - \phi^2) f_1 + (1 - \phi^1) \phi^2 f_4 \\ & + (1 - \phi^1) (1 - \phi^2) f_3 + \alpha (\nabla \phi^1 + \nabla \phi^2) dx. \end{aligned} \tag{136}$$

By constructing a special graph, one can convert this minimization problem into an equivalent continuous min-cut problem (Bae and Tai 2015, proposition 1):

$$\min_{\phi^1 \in \{0,1\}, \phi^2 \in \{0,1\}} \int_{\Omega} (1 - \phi^1) C_s^1 + (1 - \phi^2) C_s^2 + \phi^1 C_t^1 + \phi^2 C_t^2 + \tag{137}$$

$$\max\{\phi^1 - \phi^2, 0\} C^{12} - \min\{\phi^1 - \phi^2, 0\} C^{21} + \alpha |\nabla \phi^1| + \alpha |\nabla \phi^2| dx, \tag{138}$$

where

$$C_s^1 = \max\{f_4(x) - f_2(x), 0\}, \quad C_s^2 = \max\{f_3(x) - f_4(x), 0\}, \tag{139}$$

$$C_t^1(x) = \max\{f_2(x) - f_4(x), 0\}, \quad C_t^2 = \{f_4(x) - f_3(x), 0\}, \quad (140)$$

$$C^{12} = f_1(x) + f_4(x) - f_2(x) - f_3(x), \quad C^{21} = 0. \quad (141)$$

Notice that the definition of C^{12} implies the condition that $f_1(x) + f_4(x) \geq f_2(x) + f_3(x)$. This condition is expected to hold for common L^2 data fidelity term, but by solving a slightly different relaxed problem, it is also possible to handle data terms where the condition is violated (Bae and Tai 2015, Theorem 3).

One can relax the binary constraint of ϕ^1 and ϕ^2 to an interval $[0, 1]$, and the global optimal binary solution can be obtained by thresholding the optimal solution of the relaxed problem (Bae and Tai 2015, Theorem 2). Then, we can obtain a continuous max-flow formulation from the graph:

$$\max_{\{p_s^j, p_t^j, q^j\}_{j=1}^2, p^{12}} \int_{\Omega} p_s^1(x) + p_s^2(x) dx \quad (142)$$

$$\text{s.t. } \operatorname{div}(q^j(x)) - p_s^j(x) + p_t^j(x) + (-1)^{j+1} p^{12}(x), \quad j = 1, 2 \quad (143)$$

$$p_s^j(x) \leq C_s^j(x), \quad p_t^j(x) \leq C_t^j(x), \quad j = 1, 2 \quad (144)$$

$$0 = -C^{21}(x) \leq p^{12}(x) \leq C^{12}(x), \quad (145)$$

$$q^j \in C_{\alpha}, \quad j = 1, 2. \quad (146)$$

If we introduce ϕ^1 and ϕ^2 as Lagrangian multipliers for the first flow conservation constraints $j = 1$ and $j = 2$, respectively, we can obtain a primal-dual model which is equivalent to the original min-cut model (Bae and Tai 2015, section 4.1). Then, we can write the augmented Lagrangian functional as:

$$\begin{aligned} L(p_s^j, p_t^j, q^j, p^{12}, \phi^j) &= \int_{\Omega} p_s^1 + p_s^2 \\ &+ \sum_{j=1}^2 \{\phi^j (\operatorname{div}(q^j(x)) - p_s^j(x) + p_t^j(x) + p^{12}(x))\} \end{aligned} \quad (147)$$

$$- \sum_{j=1}^2 \frac{\rho_j}{2} \{\operatorname{div}(q^j(x)) - p_s^j(x) + p_t^j(x) + p^{12}(x)\}^2, \quad (148)$$

which can be solved efficiently by an ADMM algorithm like the other max-flow models mentioned before. The algorithm (Bae and Tai 2015, Algorithm 1) is presented in Algorithm 7, where the first, second, and fourth update can be explicitly derived, and the third q^j update can be approximated by Chambolle (2004).

Algorithm 7 ADMM for the continuous max-flow model (142)

-
- 1: Initialize $(\phi^j)^0, (p_s^j)^0, (p_t^j)$ and $(q^j)^0$ for $j = 1, 2$.
 - 2: **while** stopping criterion is not satisfied **do**
 - 3: $(p_s^j)^{\tau+1} = \arg \max_{p_s^j \leq C_s^j} L(p_s^j, (p_t^j)^\tau, (q^j)^\tau, (p^{12})^\tau, (\phi^j)^\tau), j = 1, 2$.
 - 4: $(p^{12})^{\tau+1} = \arg \max_{C^{21} \leq p^{12} \leq C^{12}} L((p_s^j)^{\tau+1}, (p_t^j)^\tau, (q^j)^\tau, p^{12}, (\phi^j)^\tau), j = 1, 2$.
 - 5: $(q^j)^{\tau+1} = \arg \max_{q^j \in C_\alpha} L((p_s^j)^{\tau+1}, (p_t^j)^\tau, q^j, (p^{12})^{\tau+1}, (\phi^j)^\tau), j = 1, 2$.
 - 6: $(p_t^j)^{\tau+1} = \arg \max_{p_t^j \leq C_t^j} L((p_s^j)^{\tau+1}, p_t^j, (q^j)^{\tau+1}, (p^{12})^{\tau+1}, (\phi^j)^\tau), j = 1, 2$.
 - 7: $(\phi^j)^{\tau+1} = (\phi^j)^k - \rho_j(\operatorname{div}(q^j)^{\tau+1} - (p_s^j)^{\tau+1} + (p_t^j)^{\tau+1} + (-1)^{j+1}(p^{12})^{\tau+1}), j = 1, 2$.
 - 8: **end while**
-

Extension to the High-Dimensional Graphical Models

When the input data is in a high-dimensional space, discretizing the entire domain is almost impossible. One popular choice for modeling the high-dimensional problems is to use graphical models. In general, a graph model consists of vertices, edges, and weights. Usually, one vertex corresponds to one data point in the input data set. Then, two vertices are adjacent to each other if and only if there is an edge in the graph connecting them. A weight is also assigned to each edge to measure the affinity between two adjacent vertices. If we denote the set of vertices as $V = \{x_i\}_{i=1}^N$ and the neighbors of x_i as \mathcal{N}_i , the set of edges can be represented as $E = \{(x_i, x_j) \in V^2 | x_i \in V \text{ and } x_j \in \mathcal{N}_i\}$, and the weights can be represented as $\omega = \{\omega_{ij} \in \mathbb{R}^+ | (x_i, x_j) \in E\}$. Consequently, a graph can be represented as:

$$G = (V, E, \omega). \quad (149)$$

When the graph is undirected, i.e., the edges have no directions, $(v_i, v_j) \in E$ if and only if $(v_j, v_i) \in E$. For the data labeling problems, people usually use a weighted undirected graph to model the problems (Bühler and Hein 2009; Cour et al. 2005; Osting et al. 2014).

Recently, a body of research has been devoted to formulating differential operators and variational problems on graphs (Gilboa and Osher 2008; Elmoataz et al. 2008; van Gennip and Bertozzi 2012; Lozes and Elmoataz 2014). Such variational problems have been particularly successful for unsupervised or semi-supervised classification of high-dimensional data (Bresson et al. 2012; Bertozzi and Flenner 2012; Hu et al. 2013; Toutain et al. 2014). Total variation can be extended to graphs and be used for clustering data points within each class and regularize the interphases between the classes, analogously to the Potts model for image segmentation. In Bertozzi and Flenner (2012), Merkurjev et al. (2013), Hu et al. (2013), and Garcia-Cardona et al. (2014), the resulting optimization problems were solved using phase field and the MBO scheme. More recently, convex relaxations have been derived (Merkurjev et al. 2015; Yin and Tai 2018; Bae

and Merkurjev 2017) for semi-supervised classification problems that can produce global minimizers or very close approximations.

Constructing a Graph for a Given Data Set

Given a set of data $V = \{x_i\}_{i=1}^N$ and $x_i \in \mathbb{R}^d$, we can view each data point as a vertex in the graph. Then we need to connect some pairs of points by edges and assign weights to them. One popular way is using the k -nearest neighbors method where $k > 0$ and $k \ll N$. For each x_i , we define its neighbors \mathcal{N}_i as the set of k points which are nearest to x_i . If $x_i \in \mathcal{N}_j$ or $x_j \in \mathcal{N}_i$, we connect x_i and x_j by an edge and add the pair (x_i, x_j) to the edge set E . To decide the weight $\omega(x_i, x_j)$, some popular choices are the radial basis function (Schölkopf et al. 2004):

$$\omega(x_i, x_j) = \exp\left(-d(x_i, x_j)^2 / (2\epsilon)\right), \quad (150)$$

where d is a distance metric, Zelnik-Manor and Perona function (Zelnik-Manor and Perona 2005):

$$\omega(x_i, x_j) = \exp\left(-d(x_i, x_j)^2 / (\sigma(x_i)\sigma(x_j))\right), \quad (151)$$

where $\sigma(x_i)$ denotes the local variance of x_i and the cosine similarity (Singhal et al. 2001):

$$\omega(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}. \quad (152)$$

We can further view the weights ω as a matrix $W \in \mathbb{R}^{N \times N}$ such that

$$W_{ij} = \begin{cases} \omega(x_i, x_j) & (x_i, x_j) \in E \\ 0 & (x_i, x_j) \notin E \end{cases}. \quad (153)$$

Since $k \ll N$, the affinity matrix W is a sparse matrix. Let d_i be the sum of the elements in the i th row of W . We can construct a diagonal matrix D as $D = \text{diag}(d_1, \dots, d_N)$. Then, the graph Laplacian operator can be defined as (Chung and Graham 1997)

$$L = D - W. \quad (154)$$

Given a function $u : V \rightarrow \mathbb{R}$, we can also define the differential operators $\nabla : L^2(V) \rightarrow L^2(V, L^2(V))$ and $\text{div} : L^2(V, L^2(V)) \rightarrow L^2(V)$ as (Gilboa and Osher 2008):

$$\nabla u(x_i)(x_j) = W_{ij}(u(x_j) - u(x_i)), \tag{155}$$

and

$$\operatorname{div}(f)(x_i) = \sum_{x_j \in \mathcal{N}_i} W_{ij}(f(x_j)(x_i) - f(x_i)(x_j)). \tag{156}$$

Graphical Potts Model with Simplex-Constrained Representation

Suppose we want to partition the data into n different classes, using the simplex-constrained representation, we introduce n binary functions defined on V : $v_k \in L^2(V)$. Then we can write the relaxed Potts model as (Yin and Tai 2018):

$$\min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \sum_{i=1}^N (v_k(x_i) f_k(x_i) + \alpha \|\nabla v_k(x_i)\|_1). \tag{157}$$

Similar to (70), we can derive the following identity:

$$\sum_{i=1}^N \alpha \|\nabla u(x_i)\|_1 = \sum_{i=1}^N \sum_{x_j \in \mathcal{N}_i} \alpha W_{ij} |u(x_j) - u(x_i)| = \max_{\substack{q \in L^2(V, L^2(V)) \\ \|q\|_\infty \leq \alpha}} \langle u, \operatorname{div}(q) \rangle, \tag{158}$$

where $\|q\|_\infty \leq \alpha$ means $q(x_i)(x_j) \leq \alpha$ for any x_i and x_j . Then, the graphical Potts model (157) can be written in a primal-dual form:

$$\min_{v \in \tilde{\Delta}^n} \max_{\substack{q_k \in L^2(V, L^2(V)) \\ \|q_k\|_\infty \leq \alpha}} \sum_{k=1}^n \sum_{i=1}^N v_k(x_i) f_k(x_i) + \sum_{k=1}^n \langle v_k, \operatorname{div}(q_k) \rangle. \tag{159}$$

In Yin and Tai (2018, Algorithm 1), the authors adopt the primal-dual algorithm, which is very similar to Algorithm 3, to solve it. The algorithm is given in Algorithm 8. The authors of Yin and Tai (2018) also study another quadratic relaxed version of (157):

$$\min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \sum_{i=1}^N \left(v_k(x_i) f_k(x_i) + \alpha \sum_{x_j \in \mathcal{N}_i} W_{ij} |v_k(x_j) - v_k(x_i)|^2 \right) \tag{160}$$

$$= \min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \langle v_k, f_k \rangle + \alpha \langle v_k, L v_k \rangle. \tag{161}$$

Notice that the regularization term in (157) has been modified. To solve the relaxed model (161), Yin and Tai (2018, Algorithm 2) proposed a simple projection gradient

method based on the Barzilai-Borwein step size (Barzilai and Borwein 1988; Dai and Fletcher 2005). Given the value of v_k at the $\tau - 1$ step: $v_k^{(\tau-1)}$, the gradient descent step is performed as:

$$v_k^{(\tau)} = v_k^{(\tau-1)} - \tau \partial J(v_k^{(\tau-1)}) \mathcal{Y}_k^{(\tau-1)}, \quad (162)$$

where J is the objective functional in (161) and the step size $\lambda_k^{(\tau)}$ alternates between

$$\mathcal{Y}_k^{(\tau)} = \frac{\|s_k^{(\tau-1)}\|^2}{\langle s_k^{(\tau-1)}, y_k^{(\tau-1)} \rangle}, \quad (163)$$

and

$$\mathcal{Y}_k^{(\tau)} = \frac{\langle s_k^{(\tau-1)}, y_k^{(\tau-1)} \rangle}{\|y_k^{(\tau-1)}\|^2}, \quad (164)$$

where $s_k^{(\tau)} = v_k^{(\tau)} - v_k^{(\tau-1)}$ and $y_k^{(\tau)} = \partial J(v_k^{(\tau)}) - \partial J(v_k^{(\tau-1)})$. The goal of this Barzilai-Borwein method is to approximate the Newton step without directly computing the Hessian matrices. More details about the derivation and analysis can be found in the original paper Barzilai and Borwein (1988). To ensure the decreasing of the objective functional, Yin and Tai (2018) also suggests to use a non-monotone line search method based on the Armijo-type acceptability test (Bertsekas 1976):

$$J(v_k^{(\tau)}) \leq J(v_k^{(\tau-1)}) + \theta \text{tr}(\partial J(v_k^{(\tau-1)})^T s_k^{(\tau-1)}). \quad (165)$$

Algorithm 8 Primal-dual algorithm for the graphical Potts model (Yin and Tai 2018)

- 1: initialize v_k^0 and q_k^0 for $k = 1, \dots, n$.
- 2: **while** stopping criterion is not satisfied **do**
- 3: update each q_k by

$$\begin{aligned} q_k^{\tau+1} &= \arg \max_{\|q_k\|_\infty \leq \alpha} \langle v_k(x_i), \text{div}(q_k(x_i)) \rangle - \frac{1}{2\sigma} \|q_k - q_k^\tau\|_2^2 \\ &= \Pi_{\|q_k\|_\infty \leq \alpha} (q_k^\tau - \sigma \nabla v_k^\tau) \end{aligned}$$

- 4: update v by

$$\begin{aligned} v^{\tau+1} &= \arg \min_{v \in \tilde{\Delta}^n} \sum_{k=1}^n \langle v_k, f_k + \text{div}(q_k^{\tau+1}) \rangle + \frac{1}{2\tau} \|v_k - v_k^\tau\|_2^2 \\ &= \Pi_{\tilde{\Delta}^n} (v^k - \tau(f + \text{div}(q^{k+1}))) \end{aligned}$$

- 5: **end while**
-

The whole projected gradient algorithm is described in Algorithm 9.

Algorithm 9 Projected gradient method for the relaxed graphical Potts model (Yin and Tai 2018)

```

1: initialize  $v_k^0$  for  $k = 1, \dots, n$ .
2: while stopping criterion is not satisfied do
3:   set  $\tau = 1$ .
4:   update  $v_k^{(\tau)}$  for  $k = 1, \dots, n$  by (162).
5:   while the Armijo condition (165) is not satisfied do
6:      $\tau = 0.8\tau$ .
7:     recompute  $v_k^{(\tau)}$  by (162).
8:   end while
9: end while

```

In Merkurjev et al. (2015) and Bae and Merkurjev (2017), max-flow dual formulations of the graphical extension of Potts model (157) were derived for two and n classes, respectively. In Merkurjev et al. (2015) it was proved that in case of two classes, an exact global minimizer can be obtained by thresholding the solution of the relaxed problem. In case of n classes, this cannot be guaranteed in general, but theoretical and experimental results in Bae and Merkurjev (2017) demonstrated that global solutions, or very close approximations, can be expected in practice. ADMM-based max-flow algorithms, similar to those described for image segmentation problems, were derived in Merkurjev et al. (2015) and Bae and Merkurjev (2017) using graph extensions of the differential operators.

Efficient Inference in CRF Model

One can also solve the Potts model in a probabilistic way. Consider the graphical Potts model using integer labels:

$$\min_{\phi \in \{1, \dots, n\}^N} \sum_{i=1}^N \left(f(x_i, \phi(x_i)) + \alpha \sum_{x_j \in \mathcal{N}_i} R(\phi(x_i), \phi(x_j)) \right). \quad (166)$$

Here v is a random variable taken values in S . Then a conditional random field (CRF) (Lafferty et al. 2001) can be characterized by a Gibbs distribution:

$$Pr(\phi) = \frac{1}{Z} \exp(-E(\phi)), \quad (167)$$

where I is the given image, $E(v)$ is the objective functional in (166), and $Z = \frac{1}{\sum_{v \in S} \exp(-E(v))}$ is a normalization factor. Notice that the region force term and the edge force term in E depend on the input image I . Then, a maximum a

Algorithm 10 Efficient inference in CRF model (Krähenbühl and Koltun 2011)

-
- 1: initialize $Q_i^{(0)} = \frac{1}{Z^{(0)}} \exp(-f(x_i, \phi(x_i)))$ for $i = 1, \dots, N$.
 - 2: **while** stopping criterion is not satisfied **do**
 - 3: **for** $i=1, \dots, N$ **do**
 - 4: update $Q_i^{(\tau+1)}$ by:

$$Q_i^{(\tau+1)}(\phi(x_i) = l) = \exp \left(-f(x_i, l) - \alpha \sum_{x_j \in \mathcal{N}_i} \sum_{l' \in \{l_1, \dots, l_n\}} R(\phi(x_i), l') Q_j^{(\tau)}(l') \right)$$

- 5: **end for**
 - 6: normalize $Q_i^{(\tau+1)}$ to be a valid distribution.
 - 7: **end while**
-

posteriori (MAP) method for (167) is trying to find v^* from S that maximize the posteriori probability $Pr(v|I)$, which is equivalent to minimizing (166). Instead of directly maximizing the (167), in Krähenbühl and Koltun (2011, Algorithm 1), the authors propose an efficient way to approximate the fully connected CRF. Consider the distribution $Q(\phi|I)$ with the following form:

$$Q(\phi|I) = \prod_{i=1}^N Q_i(\phi(x_i)|I), \quad (168)$$

where Q_i is the marginal distribution for $\phi(x_i)$. To find a good approximation $Q(\phi|I)$ to $Pr(\phi|I)$ (167), Krähenbühl and Koltun (2011) iteratively minimize the KL-divergence

$$D(Q||Pr) = \sum_{\phi \in \{l_1, \dots, l_n\}} Q(\phi|I) \log \left(\frac{Pr(\phi|I)}{Q(\phi|I)} \right) \quad (169)$$

among all the distribution Q satisfying (168), which leads to the following update:

$$Q_i(\phi(x_i) = l) = \frac{1}{Z} \exp \left(-f(x_i, l) - \alpha \sum_{x_j \in \mathcal{N}_i} \sum_{l' \in \{l_1, \dots, l_n\}} R(\phi(x_i), l') Q_j(l') \right). \quad (170)$$

The detailed derivation is given in Krähenbühl and Koltun (2011, supplementary material). The approximate algorithm is summarized in Algorithm 10.

Then, the label can be assigned by

$$\phi(x_i) = \arg \max_{l \in \{l_1, \dots, l_n\}} Q_i(l). \quad (171)$$

Conclusion

In this survey, we give a review for different piecewise constant representation methods for Potts model, including integer-valued representation, simplex-constrained vector representation, and overlapping binary representation. For each representation, instead of directly solving the Potts model, we introduce several convex relaxations and dual methods. Many of these methods generalize the max-flow problems on discrete graphs to a continuous setting and have dual relation with the continuous min-cut problem, i.e., the Potts model. By exploiting these dual models, we are able to present very efficient algorithms.

Acknowledgments Tai is supported by NSFC/RGC Joint Research Scheme (N-HKBU214/19), Initiation Grant for Faculty Niche Research Areas(RC-FNRA-IG/19-20/SCI/01) and CRF (C1013-21GF).

References

- Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems* **10**(6), 1217 (1994)
- Appleton, B., Talbot, H.: Globally optimal surfaces by continuous maximal flows. In: *Digital Image Computing: Techniques and Applications: Proceedings of the VIIth Biennial Australian Pattern Recognition Society Conference, DICTA 2003*, pp. 987–996. CSIRO PUBLISHING (2003)
- Bae, E., Merkurjev, E.: Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds. *J. Math. Imaging Vis.* **58**, 468493 (2017)
- Bae, E., Tai, X.-C.: Efficient global minimization for the multiphase Chan-Vese model of image segmentation. In: *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition. LNCS*, vol. 5681, pp. 28–41. Springer (2009a)
- Bae, E., Tai, X.-C.: Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. In: *Scale Space and Variational Methods in Computer Vision, second international conference, SSVM 2009, Voss, 1–5 June. Proceedings*, pp. 1–13 (2009b)
- Bae, E., Tai, X.-C.: Efficient global minimization methods for image segmentation models with four regions. *J. Math. Imaging Vis.* **51**(1), 71–97 (2015)
- Bae, E., Yuan, J., Tai, X.-C., Boykov, Y.: A fast continuous max-flow approach to non-convex multilabeling problems. *UCLA CAM Report 10–62* (2010)
- Bae, E., Yuan, J., Tai, X.-C.: Global minimization for continuous multiphase partitioning problems using a dual approach. *Int. J. Comput. Vis.* **92**(1), 112–129 (2011)
- Bae, E., Lellmann, J., Tai, X.-C.: Convex relaxations for a generalized Chan-Vese model. In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 223–236. Springer (2013)
- Bae, E., Yuan, J., Tai, X.-C., Boykov, Y.: A fast continuous max-flow approach to non-convex multi-labeling problems. In: Bruhn, A., Pock, T., Tai, X.-C. (eds.) *Efficient Algorithms for Global Optimization Methods in Computer Vision. Lecture Notes in Computer Science*, vol. 8293, pp. 134–154. Springer, Berlin/Heidelberg (2014)
- Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. *IMA J. Numer. Anal.* **8**(1), 141–148 (1988)
- Bertozi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Model. Simul.* **10**(3), 1090–1118 (2012)

- Bertsekas, D.P.: On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Autom. Control* **21**(2), 174–184 (1976)
- Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 359–374 (2001)
- Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pp. 648–655. IEEE (1998)
- Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 1222–1239 (2001)
- Boyle, J.P., Dykstra, R.L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: *Advances in Order Restricted Statistical Inference*, pp. 28–47. Springer, New York (1986)
- Bresson, X., Laurent, T., Uminsky, D., von Brecht, J.: Convergence and energy landscape for Cheeger cut clustering. In: *Advances in Neural Information Processing Systems (NIPS)*, p. 13941402 (2012)
- Bühler, T., Hein, M.: Spectral clustering based on the graph p-Laplacian. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 81–88. ACM (2009)
- Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* **20**(1–2), 89–97 (2004)
- Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
- Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T.: An introduction to total variation for image analysis. *Theor. Found. Numer. Methods Sparse Recovery* **9**(263–340), 227 (2010)
- Chambolle, A., Cremers, D., Pock, T.: A convex approach to minimal partitions. *SIAM J. Imaging Sci.* **5**(4), 1113–1158 (2012)
- Chan, T., Vese, L.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
- Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006)
- Chung, F.R., Graham, F.C.: *Spectral Graph Theory*, vol. 92. American Mathematical Society, Providence (1997)
- Coupric, C., Grady, L., Talbot, H., Najman, L.: Combinatorial continuous maximum flow. *SIAM J. Imaging Sci.* **4**(3), 905930 (2011)
- Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 1124–1131. IEEE (2005)
- Dai, Y.-H., Fletcher, R.: Projected barzilai-borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik* **100**(1), 21–47 (2005)
- Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part I: Fast and exact optimization. *J. Math. Imaging Vis.* **26**(3), 261–276 (2006)
- Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part II: Levelable functions, convex priors and non-convex cases. *J. Math. Imaging Vis.* **26**(3), 277–291 (2006)
- Ekeland, I., Temam, R.: *Convex Analysis and Variational Problems*, vol. 28. SIAM, Philadelphia (1999)
- Elmoataz, A., Lezoray, O., Boughleux, S.: Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Trans. Image Process.* **17**(7), 1047–1060 (2008)
- Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.* **3**(4), 1015–1046 (2010)
- Fleming, W.H., Rishel, R.: An integral formula for total gradient variation. *Archiv der Mathematik* **11**, 218–222 (1960)

- Garcia-Cardona, C., Merkurjev, E., Bertozzi, A., Flenner, A., Percus, A.: Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1600–1613 (2014)
- Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984)
- Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2008)
- Hu, H., Laurent, T., Porter, M.A., Bertozzi, A.L.: A method based on total variation for network modularity optimization using the MBO scheme. *SIAM J. Appl. Math.* **73**(6), 2224–2246 (2013)
- Ishikawa, H.: Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(10), 1333–1336 (2003)
- Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with Gaussian edge potentials. In: *Advances in Neural Information Processing Systems*, pp. 109–117 (2011)
- Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
- Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained total variation. In: *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 150–162. Springer (2009)
- Lie, J., Lysaker, M., Tai, X.-C.: Piecewise constant level set methods and image segmentation. In: *International Conference on Scale-Space Theories in Computer Vision*, pp. 573–584. Springer (2005)
- Lie, J., Lysaker, M., Tai, X.-C.: A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Trans. Image Process.* **15**(5), 1171–1181 (2006a)
- Lie, J., Lysaker, M., Tai, X.-C.: A variant of the level set method and applications to image segmentation. *AMS Math. Comput.* **75**(255), 1155–1174 (2006)
- Liu, J., Tai, X.-C., Leung, S., Huang, H.: A new continuous max-flow algorithm for multiphase image segmentation using super-level set functions. *J. Vis. Commun. Image Represent.* **25**(6), 1472–1488 (2014)
- Lozes, O.L.F., Elmoataz, A.: Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Trans. Image Process.* **23**, 3896–3909 (2014)
- Merkurjev, E., Kostic, T., Bertozzi, A.L.: An MBO scheme on graphs for classification and image processing. *SIAM J. Imaging Sci.* **6**(4), 1903–1930 (2013)
- Merkurjev, E., Bae, E., Bertozzi, A.L., Tai, X.-C.: Global binary optimization on graphs for classification of high-dimensional data. *J. Math. Imaging Vis.* **52**(3), 414–435 (2015)
- Michelot, C.: A finite algorithm for finding the projection of a point onto the canonical simplex of n . *J. Optim. Theory Appl.* **50**(1), 195–200 (1986)
- Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
- Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, vol. 153. Springer, New York (2003)
- Osting, B., White, C.D., Oudet, É.: Minimal Dirichlet energy partitions for graphs. *SIAM J. Sci. Comput.* **36**(4), A1635–A1651 (2014)
- Papadimitriou, C.H., Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation, Dover Publications, Inc, Mineola (1998)
- Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of multi-label problems. In: *European Conference on Computer Vision*, pp. 792–805. Springer (2008)
- Pock, T., Chambolle, A., Cremers, D., Bischof, H.: A convex relaxation approach for computing minimal partitions. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 810–817. IEEE (2009)
- Potts, R.B.: Some generalized order-disorder transformations. In: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 106–109. Cambridge University Press (1952)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**(1–4), 259–268 (1992)

- Schölkopf, B., Tsuda, K., Vert, J.-P.: *Kernel Methods in Computational Biology*. MIT Press, Cambridge (2004)
- Schrijver, A.: On the history of the transportation and maximum flow problems. *Math. Program.* **91**(3), 437–445 (2002)
- Singhal, A., et al: Modern information retrieval: a brief overview. *IEEE Data Eng. Bull.* **24**(4), 35–43 (2001)
- Strang, G.: Maximal flow through a domain. *Math. Program.* **26**, 123–143 (1983)
- Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**(1), 146–159 (1994)
- Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Agarwala, A., Rother, C.: A comparative study of energy minimization methods for Markov random fields. In: *ECCV*, pp. 16–29 (2006)
- Toutain, M., Elmoataz, A., Lzoray, O.: Geometric pdes on weighted graphs for semi-supervised classification. In: *13th International Conference on Machine Learning and Applications (ICMLA)*, pp. 231–236 (2014)
- van Gennip, Y., Bertozzi, A.: Gamma-convergence of graph Ginzburg-Landau functionals. *Adv. Differ. Equ.* **17**(11–12), 1115–1180 (2012)
- Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.* **50**(3), 271–293 (2002)
- Wu, C., Tai, X.-C.: Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sci.* **3**, 300–339 (2010)
- Yin, K., Tai, X.-C.: An effective region force for some variational models for learning and clustering. *J. Sci. Comput.* **74**(1), 175–196 (2018)
- Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A continuous max-flow approach to Potts model. In: *European Conference on Computer Vision*, pp. 379–392. Springer (2010)
- Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A spatially continuous max-flow and min-cut framework for binary labeling problems. *Numerische Mathematik* **126**(3), 559–587 (2014)
- Zach, C., Gallup, D., Frahm, J.-M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: *Vision, Modeling and Visualization*, pp. 243–252 (2008)
- Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *Advances in Neural Information Processing Systems*, pp. 1601–1608 (2005)
- Zhao, H., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *J. Comput. Phys.* **127**(1), 179–195 (1996)
- Zhu, M., Chan, T.F.: Fast numerical algorithms for total variation based image restoration. PhD thesis, University of California, Los Angeles (2008)



Shape Spaces: From Geometry to Biological Plausibility

53

Nicolas Charon and Laurent Younes

Contents

Introduction: Shape Spaces	1930
Shape Spaces Under Diffeomorphic Action	1933
Hybrid Models	1936
Description	1936
Elastic Metrics	1937
Growth Models	1943
Introduction	1943
Riemannian Viewpoint	1944
Growth as an Internal Force	1945
A Simple Example	1946
Growth Due to External Action	1947
Constraints, Deformation Modules, and Other Growth Models	1949
Conclusion	1950
Appendix A: Elastic Surface Metric as the Limit of the Laminar Model (section “Elastic Metrics on Surfaces”)	1951
Appendix B: Existence of Optimal Paths (section “Riemannian Viewpoint”)	1953
References	1955

Nicolas Charon is partially supported by NSF 1945224 and NSF 1953267.

Laurent Younes is partially supported by NIH U19AG033655, R01NS102670, and R01AG055121.

N. Charon · L. Younes (✉)

Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA

e-mail: ncharon1@jhu.edu; laurent.younes@jhu.edu

Abstract

This chapter reviews several Riemannian metrics and evolution equations in the context of diffeomorphic shape analysis. After a short review of various approaches at building Riemannian spaces of shapes, with a special focus on the foundations of the large deformation diffeomorphic metric mapping algorithm, the attention is turned to elastic metrics and to growth models that can be derived from it. In the latter context, a new class of metrics, involving the optimization of a growth tensor, is introduced, and some of its properties are studied.

Keywords

Riemannian shape spaces · Shape analysis · Shape evolution · Diffeomorphisms · Morphoelasticity · Growth models

Introduction: Shape Spaces

Shape has long been an object of scientific study, especially in life sciences where it provided a primary element in the differentiation between species. It was – in complement to behavioral patterns – a central factor of the early justification of evolutionary theory and was of course the main subject of D’Arcy Thompson seminal work *On Growth and Form* (Thompson 1917).

The construction of mathematical models of shape spaces, however, was more recent and started with David Kendall’s landmark paper introducing a shape space as a particular Riemannian manifold (Kendall 1984), a construction motivated by the need to provide a formal mathematical framework for statistical analyses of shape datasets. In Kendall’s model, shapes are represented as ordered collections of distinct points with fixed cardinality. The manifold structure is obtained as a quotient space through the action of rotations, translations, and scaling and the metric as the projection of the Euclidean metric to this quotient space. Kendall’s shape space has since been used in a large variety of applications, with increasing numbers of available shape datasets and relevant associated statistical questions (see the recent edition of Dryden and Mardia (2016) for additional details and references).

Kendall’s shape space is however limited by the need to provide a consistent ordering (or labeling) of the points constituting the shape and by the requirement that they form a finite set. Shape datasets are typically formed by unlabeled geometric objects, and using Kendall’s shape space requires defining and indexing (often manually) collections of landmarks for each shape, resulting in an intensive and sometimes imperfectly specified problem. Defining shape spaces whose elements are curves or surfaces requires however more advanced mathematical tools, notably from global analysis (Palais 1968), and a recent description of various formulations of shape spaces in this general context can be found in Bauer et al. (2014b). In spite of the additional mathematical technicality, the construction of these shape spaces follows the general principles leading to Kendall’s space: first define a simple space of geometric objects as an open subset of a normed (or Fréchet) space, where the finite-dimensional space of ordered distinct points is replaced, e.g., by a space of

immersions (or embeddings) from a fixed manifold M (the parameter space) to \mathbb{R}^d , the ambient space. This space (and its norm) is then quotiented by group actions to which shapes must be invariant, bringing in, in addition to previous actions of translations, rotations, and scaling, the infinite-dimensional group of reparametrizations, provided by diffeomorphisms of the parameter space. Another modification to the finite-dimensional framework is that the Euclidean metric, as the base norm, which was a natural choice when working with finite sets of points, now needs to be replaced with some invariant Hilbert metric (if one wants a Riemannian structure at the end) on the space of immersions, for which there are many choices, including the whole family of invariant Sobolev norms. The well-posedness of various concepts in the resulting shape space, such as the non-degeneracy of the metric or the existence of geodesics, indeed depends on this choice. A striking example is the fact that the Riemannian distance between any pair of shapes may trivialize to zero for certain metrics, as initially discovered in Michor and Mumford (2005) in the case of curves and then extended to other shape spaces (Bauer et al. 2020).

From the whole variety of shape spaces that can be built following this construction, a small number actually lead to practical algorithms and numerical implementations, which is an essential requirement when the goal is to analyze shape datasets. For curves, an important example is associated with a class of first-order Sobolev metrics on the space of immersions. One can indeed show that, after quotienting out rotations, translations, and/or scalings, the resulting Riemannian manifold is isomorphic to standard manifolds (such as the infinite-dimensional sphere and Stiefel or Grassmann manifolds) on which geodesic and geodesic distances can be explicitly computed. For curves, the additional cost of adding reparametrization invariance remains manageable, using, e.g., dynamic programming methods. A first example of such metrics was provided in Younes (1996, 1998) with further developments in Younes et al. (2008). A second example was then provided in Klassen et al. (2004) (see Srivastava and Klassen 2016), and the approach was later extended to a one-parameter family including these two examples in Needham and Kurtek (2020) and Younes (2019), chapter 12 (see also Bauer et al. 2014a).

Shape spaces have also been built using a different angle, leveraging the action of the diffeomorphism group of \mathbb{R}^d on a shape space. Diffeomorphisms of their ambient space indeed act transitively on most shapes of interest assuming that one fixes their topology (taking an example, diffeomorphisms of \mathbb{R}^2 can be used to transform any C^1 Jordan plane curve to any other). Using a metric on the diffeomorphism group with suitable properties, one can, given two shapes, compute the diffeomorphism closest to the identity that transforms the first shape into the other, and the distance between the identity and this optimal diffeomorphism also provides a distance between the considered shapes. (This construction will be described in detail in section “Shape Spaces Under Diffeomorphic Action”.) Formally, the considered shape space is provided by all diffeomorphic transformation of a given template. This approach can be seen as an application of Grenander’s metric pattern theory (Grenander and Keenan 1991; Grenander 1993) and as a mathematical formulation of D’Arcy Thompson models (Thompson 1917). It was introduced for shape spaces of images synchronously in Dupuis et al. (1998) (with a precursor

in Christensen et al. 1996) and Trouvé (1995, 1998) and for collections of labeled points in Miller et al. (1999). This formulation, very flexible, has later been applied to various shape spaces, such as unlabeled point sets (Glaunès et al. 2004), curves and surfaces (Vaillant and Glaunès 2005; Glaunès et al. 2008), and vector or tensor fields (Cao et al. 2005, 2006). The reader may also refer to the recent survey in Bauer et al. (2019) that describes in details the two previous approaches in the case of curves and surfaces.

Note that the previous discussion does not include the many methods that provide shape features, i.e., finite- or infinite-dimensional descriptors that can be attached to a given shape, without necessarily providing them with a clear mathematical structure (such as that of a Riemannian manifold) which is one of the main concerns of the construction of shape spaces. Such methods were introduced in computer vision, medical imaging, and biology and are too numerous to cite exhaustively in this chapter. Among the most important ones (a subjective statement), one can cite approaches using complex analysis and the (quasi-)conformal maps to represent surfaces (Gu et al. 2004; Gu and Yau 2008; Zeng and Gu 2011; Zeng et al. 2012; Lui et al. 2014), isometry-invariant descriptors based on distance maps or Laplace-Beltrami eigenvectors (Bronstein et al. 2008a,b; Mémoli 2008; Ovsjanikov et al. 2010; Mémoli 2011), or the shape context (Belongie et al. 2002). The rest of this chapter will however remain focused on shape space approaches.

The construction of shape spaces as described above is based on purely geometric aspects. No physical law or biological mechanism is used to define the various components that constitute the shape space. This non-committal approach is indeed justified, as shape spaces are designed as containers for families of shapes that are not related to each other by a natural process (e.g., there is no physical process by which a finch's beak can transform into the shape of another one). This fact provides the technical advantage that the construction of shape metrics is not constrained by the laws of nature and can therefore be selected so that they guarantee the existence, say, of geodesics, provide nicely behaved gradient flows, etc. This will be illustrated in section “[Shape Spaces Under Diffeomorphic Action](#)”.

On the other hand, biological processes provide many examples in which shapes change with time, in a process that is constrained by well-specified laws. The goal of this chapter is to describe a few among recent attempts at representing such processes as trajectories in the shape spaces above, which, after small modifications or regularization, will be associated with evolution dynamics that behave well enough to allow for long-time analysis and optimal control formulations.

This chapter is organized as follows. Section “[Shape Spaces Under Diffeomorphic Action](#)” provides a summary of the construction of shape spaces through diffeomorphic action. Section “[Hybrid Models](#)” focuses on variations of this construction with metrics that are inspired from elastic materials. Section “[Growth Models](#)” introduces a few examples of growth models in the context of shape spaces. For an extensive introduction to mathematical models of growth, the reader should refer to Goriely (2017), which provides a splendid reference on the topic and in particular on “morphoelasticity.” The representation of shape growth described in section “[Growth Models](#)” will, however, deviate to some extent from that

described in this reference and more generally from the large literature exploring morphoelasticity, as models will be designed in the form of control systems, with a control equation interpreted as a differential equation in shape space and growth or atrophy directly associated with the control.

Shape Spaces Under Diffeomorphic Action

This section provides a summary of the construction of shape spaces based on the principles of D’Arcy Thompson’s theory of transformations (Thompson 1917) and Grenander’s metric pattern theory (Grenander 1993). The fundamental principles of the construction were laid in Dupuis et al. (1998), Trouvé (1995), Grenander and Miller (1998), and the reader may refer to Younes (2019), Miller et al. (2015), Bauer et al. (2019) for more recent accounts of the theory.

Shapes are modeled as embeddings from a fixed Riemannian manifold M into \mathbb{R}^d and therefore have a fixed topology (in practice, $d = 2$ or 3). Typically, M is a unit circle or sphere or a template shape of which one is computing deformations. Denote by $Emb^p(M)$ the set of such C^p embeddings or simply Emb when p and M are fixed. Each element $m \in Emb$ provides a shape equipped with a parametrization. Objects of interest are shapes modulo parametrization (also called “unparametrized shapes”) in which one identifies embeddings m and \tilde{m} when they are related with each other through a change of parametrization, i.e., $\tilde{m} = m \circ \rho$ where ρ is a diffeomorphism of M . In other terms, the shape space is defined as the quotient space of Emb through the right action of the diffeomorphism group of M and will be denoted as \mathcal{S} . Elements of \mathcal{S} will be denoted as $[m]$, for the equivalence class of $m \in Emb$.

Comparisons between shapes rely on the group of transformations acting on Emb or \mathcal{S} , which are modeled as diffeomorphisms of \mathbb{R}^d . Denote by $Diff^p(\mathbb{R}^d)$, or simply $Diff^p$, the group of C^p diffeomorphisms of \mathbb{R}^d and by $Diff_0^p(\mathbb{R}^d)$, or simply $Diff_0^p$, the subgroup of diffeomorphisms that converge to the identity map, denoted $id_{\mathbb{R}^d}$, at infinity (convergence being understood in the C^p sense). If $\varphi \in Diff^p$ and $m \in Emb$, $\varphi \cdot m$ is simply $\varphi \circ m$, and this action commutes with reparametrization, so that one can define $\varphi \cdot [m] = [\varphi \cdot m]$ without ambiguity.

To compare two embeddings (or their associated shapes) m and m' , one considers the transformations $\varphi \in Diff_0^p$ that relate them, i.e., such that $m' = \varphi \cdot m$. One considers that m and m' are similar if one can find some φ relating them that is close to $id_{\mathbb{R}^d}$. This closeness is itself evaluated using a metric on $Diff_0^p$, with a construction described below.

To provide a Riemannian metric, one needs an inner-product norm that evaluates the velocity of time-dependent diffeomorphisms, or “diffeomorphic motions,” taking the form $(x \mapsto \partial_t \varphi(t, x))$ where φ is a function of time and space such that $(x \mapsto \varphi(t, x))$ is at all times an element of $Diff_0^p$. For a given time t , $(x \mapsto \partial_t \varphi(t, x))$ is a C^p vector field on \mathbb{R}^d , and one therefore needs to provide a norm over such vector fields. The norm of the velocity at time t should in principle depend on the diffeomorphism at the same time, $(x \mapsto \varphi(t, x))$, but, for reasons seen below, it will

be desirable for this norm to satisfy the invariance property that, when writing

$$\varphi(t + \delta t, x) = \varphi(t, x) + \delta\varphi(t, x) = (\text{id}_{\mathbb{R}^d} + \delta\varphi(t, \cdot) \circ \varphi^{-1}(t, \cdot)) \circ \varphi(t, x),$$

the cost associated with $\delta\varphi$ is a fixed function of the deformation increment $\delta\varphi \circ \varphi^{-1}$. Passing to the limit, this means that the Riemannian norm of $(x \mapsto \partial_t \varphi(t, x))$ at $(x \mapsto \varphi(t, x))$ is equal to the norm of $(x \mapsto \partial_t \varphi(t, x) \circ \varphi^{-1}(t, x))$ at $\text{id}_{\mathbb{R}^d}$. The vector field $v(t, x) = \partial_t \varphi(t, x) \circ \varphi^{-1}(t, x)$ is called the Eulerian velocity of the diffeomorphic motion $x \mapsto \varphi(t, x)$, and the diffeomorphic motion is recovered from the Eulerian velocity by solving the ordinary differential equation

$$\partial_t \varphi(t, x) = v(t, \varphi(t, x)). \tag{1}$$

To define our Riemannian metric on Diff_0^p , it therefore suffices to specify a Hilbert norm on vector fields. For this purpose, let V denote a Hilbert of vector fields on \mathbb{R}^d that will be assumed, in order to recover elements of Diff_0^p after solving Eq. (1), to be continuously included in the space $C_0^p(\mathbb{R}^d, \mathbb{R}^d)$ of C^p vector fields that vanish at infinity. This means that $V \subset C_0^p(\mathbb{R}^d, \mathbb{R}^d)$ and that, for some constant c , one has (letting $\|\cdot\|_\infty$ denote the supremum norm)

$$\sum_{k=0}^p \|d^k v\|_\infty \leq c \|v\|_V.$$

To satisfy this assumption, V can be built as a Hilbert Sobolev space of high enough order. In addition, since the continuous inclusion implies that V is a reproducing kernel Hilbert space (RKHS) of vector fields, RKHS theory can be used to build a large variety of Hilbert spaces of interest that satisfy the inclusion property (Aronszajn 1950; Kadri et al. 2016; Younes 2019). One can then define the action functional of a diffeomorphic motion $((t, x) \in [0, 1] \times \mathbb{R}^d \mapsto \varphi(t, x) \in \mathbb{R}^d)$ as

$$\int_0^1 \|v(t, \cdot)\|_V^2 dt$$

with $\partial_t \varphi(t, x) = v(t, \varphi(t, x))$. A geodesic diffeomorphic motion is an extremal of this action functional, and a minimizing geodesic motion minimizes the functional subject to fixed boundary conditions at $t = 0$ and $t = 1$. In particular, the geodesic distance between two diffeomorphisms φ_0 and φ_1 is defined as

$$d_V(\varphi_0, \varphi_1) = \inf \left\{ \left(\int_0^1 \|v(t, \cdot)\|_V^2 dt \right)^{1/2} : \partial_t \varphi(t, x) = v(t, \varphi(t, x)), \varphi(0, \cdot) = \varphi_0, \varphi(1, \cdot) = \varphi_1 \right\}. \tag{2}$$

Note that the set over which the infimum is computed may be empty, in which case the distance is infinite. If this set is not empty, then one says that φ_1 is attainable from φ_0 . Diffeomorphisms that are attainable from the identity form a subgroup of $Diff_0^p$, denoted $Diff_V$, and this subgroup is complete for the geodesic distance (Trouné 1995; Younes 2019). (Because not every diffeomorphism in $Diff_0^p$ is attainable from the identity, one is actually building a sub-Riemannian metric on this space. See Arguillère et al. 2014; Younes et al. 2020.)

By construction, the distance is right-invariant, i.e.,

$$d_V(\varphi_0, \varphi_1) = d_V(\text{id}_{\mathbb{R}^d}, \varphi_1 \circ \varphi_0^{-1}),$$

and this implies that it can be used to define a distance on \mathcal{S} via

$$\begin{aligned} d_{\mathcal{S}}([m_0], [m_1]) &= \inf \{ d_V(\text{id}_{\mathbb{R}^d}, \varphi) : [\varphi \cdot m_0] = [m_1] \} \\ &= \inf \{ d_V(\text{id}_{\mathbb{R}^d}, \varphi) : \varphi \cdot m_0 \in [m_1] \}. \end{aligned}$$

The distance on \mathcal{S} can itself be defined directly as

$$d_V([m_0], [\varphi_1]) = \inf \left\{ \left(\int_0^1 \|v(t, \cdot)\|_V^2 dt \right)^{1/2} : \partial_t m(t, \cdot) = v(t, m(t, \cdot)), m(0, \cdot) = m_0, m(1, \cdot) \in [m_1] \right\}. \tag{3}$$

This provides an optimal control problem in \mathcal{S} where the control is the time-dependent vector field v and the state equation the ODE $\partial_t m(t, \cdot) = v(t, m(t, \cdot))$. The optimal trajectory transforms the initial m_0 into an embedding that is a reparametrization of m_1 and provides a minimizing geodesic in \mathcal{S} . If the Sobolev inclusion discussed above holds for $p \geq 1$ at least, the variational problems described in Eqs. (2) and (3) are well defined. The condition that $\int_0^1 \|v(t, \cdot)\|_V^2 dt < \infty$ implies that solutions to the state equations ($\partial_t \varphi = v \circ \varphi$ or $\partial_t m = v \circ m$) exist and are unique (given initial conditions) over the full unit time interval, ensuring that the optimal control problem is well specified. Moreover, as soon as φ_1 (resp. $[m_1]$) is attainable from φ_0 (resp. $[m_0]$), an optimal solution to the considered problem always exists. Finally, under very mild assumptions on initial conditions, solutions of the geodesic equations exist and are uniquely specified by their initial position and velocity, i.e., $m(0, \cdot)$ and $\partial_t m(0, \cdot)$ for $d_{\mathcal{S}}$. The geodesic equation is the Euler-Lagrange equation associated with the variational problem, satisfied by stationary points of Eq. (2) or (3) (equivalently, they are the equations provided by Pontryagin’s maximum principle). In the case considered here, they are special instances of the geodesic equations for right-invariant Riemannian metrics on Lie groups, as described in Arnold (1966, 1978), and are often referred to as



Fig. 1 Four time points of a geodesic evolution in shape space. Note that shapes in this example have multiple components. Contour coloring match across time points and track the evolution of the curve initial parametrization

Euler-Arnold equations (Arnold and Khesin 2021) or Euler-Poincaré equations (Ebin and Marsden 1970; Holm et al. 1998).

In practice, one does not solve this problem exactly, but relaxes the endpoint condition $m(1, \cdot) \in [m_1]$ by adding a penalty term, therefore minimizing

$$\int_0^1 \|v(t, \cdot)\|_V^2 dt + U([m(1, \cdot)], [m_1]) \tag{4}$$

subject to $\partial_t m(t, \cdot) = v(t, m(t, \cdot))$. In many of the applications, the function U takes the form

$$U([m_0], [m_1]) = \|\mathcal{J}_{[m_0]} - \mathcal{J}_{[m_1]}\|_H^2$$

where $[m] \mapsto \mathcal{J}_{[m]}$ is a mapping from \mathcal{S} into a (much larger) Hilbert space H . These “chordal metrics” use representations of embedded curves or surfaces as measures, currents, or varifold. For simplicity, the presentation below will ignore this relaxation step (which is however necessary to make the computation numerically feasible) and work as if the endpoint conditions are exact. The reader is referred to Bauer et al. (2019) or Charon et al. (2020), and to the references within, for more information on chordal metrics.

A two-dimensional example of geodesic is presented in Fig. 1. These geodesics provide the non-linear equivalent of a linear interpolation in Euclidean space.

Hybrid Models

Description

The previous framework can be slightly extended to allow the norm used in the shape space to depend on the shape itself, replacing the control cost in Eq. (3) by

$$\int_0^1 \|v(t, \cdot)\|_{[m(t, \cdot)]}^2 dt,$$

so that the cost depends on both control and state. This still provides a sub-Riemannian distance in shape space, and the problem remains well specified as soon as one ensures that the shape-dependent norms still control the norm on V , so that an inequality ensuring

$$\|v\|_V \leq C \|v\|_{[m]}$$

holds for all $m \in \mathcal{S}$ and $v \in V$ (where the upper bound may be infinite). Typical applications of this construction use a “weak norm” $v \mapsto \llbracket v \rrbracket_{[m]}$ (which, by itself, would not guarantee the existence of solutions to the state equation), possibly motivated by material or biological constraints, “regularized” by the norm on V , therefore taking

$$\|v\|_{[m]}^2 = \kappa \|v\|_V^2 + \llbracket v \rrbracket_{[m]}^2 \quad (5)$$

for some $\kappa > 0$.

The following section discusses several possible choice for $\llbracket v \rrbracket_{[m]}$ in Eq. (5), in which the shape is considered as an elastic material and the norm corresponds to the elastic energy associated with an infinitesimal displacement along v (the reader may refer to, e.g., Ciarlet (1988); Gonzalez and Stuart (2008) for more details on elasticity concepts that are used below). The concept of “hybrid” metrics in Eq. (5) was suggested in Younes (2018b). A similar approach for spaces of images (combined with a “metamorphosis” metric (Miller and Younes 2001; Trouvé and Younes 2005)) was introduced in Berkels et al. (2015), and metrics formed as discrete iterations of small elastic deformations were also studied in Wirth et al. (2011).

Elastic Metrics

Three-Dimensional Case

The energy of a hyperelastic material Ω subject to a deformation φ takes the form (letting $\text{Id}_{\mathbb{R}^d}$ denote the identity matrix in \mathbb{R}^d)

$$E = \int_{\varphi(\Omega)} G(x, \varphi(x)) dx$$

where

$$G(x, \varphi) = W\left(x, d\varphi^T d\varphi - \text{Id}_{\mathbb{R}^3}\right)$$

for a function $W : \Omega \times \text{Sym}^+ \rightarrow [0, +\infty)$ (where Sym^+ is the set of 3 by 3 positive semi-definite matrices) such that $W(x, S) = 0$ if and only if $S = 0$. The matrix

$C = d\varphi^T d\varphi$ is the Cauchy-Green strain tensor, which is such that $u^T C u = |d\varphi u|^2$, and W measures the deviation of this tensor from the identity matrix.

A second-order expansion of G , near $\varphi = \text{id}_{\mathbb{R}^3}$, takes the form (using the fact that $\partial_2 W(x, 0) = 0$)

$$G(x, \varphi) \simeq \frac{1}{2} \partial_2^2 W(x, 0) (dv + dv^T, dv + dv^T) \tag{6}$$

where $v = \varphi - \text{id}_{\mathbb{R}^3}$. Here, $\partial_2 W(x, 0)$ and $\partial_2^2 W(x, 0)$ are the first and second derivatives with respect to the second variable of W , therefore a positive semi-definite symmetric bilinear form on Sym (the space of 3 by 3 symmetric matrices).

This can be used to define an elastic metric on 3D vector fields. Here, Ω is considered as an “unparametrized shape,” taking the role of $[m]$ in the previous sections. Using the previous notation, this corresponds to taking the manifold M to be an open subset of \mathbb{R}^3 (e.g., an open ball), m an embedding of M into \mathbb{R}^3 and identifying $\Omega = m(M)$ to $[m]$. The hybrid norm will therefore be denoted as

$$\|v\|_{\Omega}^2 = \kappa \|v\|_V^2 + \llbracket v \rrbracket_{\Omega}^2$$

and the rest of the discussion focuses on $\llbracket v \rrbracket_{\Omega}$. Based on Eq. (6), one is led to define a 3D elastic metric on vector fields as any norm taking the form

$$\llbracket v \rrbracket_{\Omega}^2 = \int_{\Omega} B(x, \varepsilon(x)) dx$$

where $\varepsilon(x) = (dv(x) + dv(x)^T)/2$ is known as the infinitesimal strain tensor of the deformation and $B(x, \cdot)$ is a positive semi-definite quadratic form on Sym typically referred to as the elastic tensor. Generically, $B(x, \cdot)$ can be represented as a 6×6 symmetric positive semi-definite matrix, that is, with 21 parameters in total at each x . In a majority of applications however, model symmetry assumptions significantly reduce the complexity of this elasticity tensor. In particular, in the case of a uniform and isotropic material, $B(x, \cdot)$ is independent of the position and takes the specific form

$$B(x, \varepsilon) = B(\varepsilon) = \frac{\lambda}{2} \text{trace}(\varepsilon)^2 + \mu \text{trace}(\varepsilon^2) \tag{7}$$

which is the linearization of the energy of a Saint Venant-Kirchhoff material. In that case, the elasticity tensor is only described by the two parameters λ and μ which are called the Lamé coefficients of the material.

To provide another example, consider the case of a partially isotropic and laminar model, introduced in Hsieh et al. (2019, 2021, 2022) under the assumption that Ω can be parametrized by a foliation. More precisely, assume that there exist two surfaces $\mathcal{M}_{\text{bottom}}$ and \mathcal{M}_{top} (bottom and top layers) included in $\partial\Omega$ and a diffeomorphism $\Phi : [0, 1] \times \mathcal{M}_{\text{bottom}} \rightarrow \Omega$ such that $\Phi(\{0\} \times \mathcal{M}_{\text{bottom}}) = \mathcal{M}_{\text{bottom}}$

and $\Phi(\{1\} \times \mathcal{M}_{\text{bottom}}) = \mathcal{M}_{\text{top}}$. Let $\mathcal{M}_s = \Phi(\{s\} \times \mathcal{M}_{\text{bottom}})$, $s \in [0, 1]$, denote “the layer at level s ,” S the transverse vector field $S = \partial_s \Phi$, and N a unit vector field normal to all \mathcal{M}_s . One then introduces the following elasticity tensor:

$$\begin{aligned}
 B(x, \varepsilon) = & \lambda_{\text{tan}} \left(\text{trace}(\varepsilon) - N^T \varepsilon N \right)^2 + \mu_{\text{tan}} \left(\text{trace}(\varepsilon^2) - 2 N^T \varepsilon^2 N + (N^T \varepsilon N)^2 \right) \\
 & + \mu_{\text{tsv}} (S^T \varepsilon S)^2 + 2 \mu_{\text{ang}} \left(S^T \varepsilon^2 S - (N^T \varepsilon S)^2 \right),
 \end{aligned}
 \tag{8}$$

The first two terms in this expression define an isotropic model on each layer. The third term measures a transversal string, evaluated along S . The last term measures an angular strain that vanishes when S is normal to the layers. Here, the coefficients λ_{tan} , μ_{tan} , μ_{tsv} , and μ_{ang} must be constant on each layer m_s (they may depend on s). Note that, if τ_1 and τ_2 are two orthonormal vector fields that are tangent to the layers so that (τ_1, τ_2, N) forms at all points an orthonormal frame, then

$$\text{trace}(\varepsilon) - N^T \varepsilon N = \tau_1^T \varepsilon \tau_1 + \tau_2^T \varepsilon \tau_2$$

and

$$\text{trace}(\varepsilon^2) - 2 N^T \varepsilon^2 N + (N^T \varepsilon N)^2 = (\tau_1^T \varepsilon \tau_1)^2 + (\tau_2^T \varepsilon \tau_2)^2 + 2(\tau_1^T \varepsilon \tau_2)^2$$

so that the first two terms only involve deformations tangent to the layers.

Importantly, the space of such “layered structures” is stable by diffeomorphic action. Indeed, given Ω and Φ as above, and φ a diffeomorphism of \mathbb{R}^3 , one defines the transformed structure by

$$\varphi \cdot (\Omega, \Phi) = (\varphi(\Omega), \varphi \circ \Phi \circ \varphi^{-1}).$$

In particular, S transforms through φ as $\varphi \cdot S = (d\varphi S) \circ \varphi^{-1}$.

Returning to the general case, one must emphasize the fact that the action functional

$$\int_0^1 \int_{\Omega(t)} B(x, \varepsilon(x)) dx dt$$

with $\partial_t \varphi(t, x) = v(t, \varphi(t, x))$ and $\Omega(t) = \varphi(t, \cdot)(\Omega_0)$ is not the energy of a deforming elastic material, in the sense given to it in elasticity theory. In contrast, it may be understood as a sum of infinitesimal elastic energies, for a volume that slowly deforms, and, at each time step, remodels its structure to reach an equilibrium state *without – up to reorientation – changing its elasticity properties*.

Elastic Metrics on Surfaces

The definition of elastic metrics on surfaces can be inferred using a pattern similar to the 3D derivation. Let \mathcal{M} be a surface in \mathbb{R}^3 and consider a one-to-one immersion $\varphi : \mathcal{M} \rightarrow \mathbb{R}^3$ (one can, in this discussion, think of φ as the restriction to \mathcal{M} of a diffeomorphism of \mathbb{R}^3). To define a hyperelastic energy, assume (restricting \mathcal{M} if needed and introducing partitions of unity) that two vector fields τ_1 and τ_2 are chosen on \mathcal{M} such that they form at each point an orthonormal frame, and let $\nu = \tau_1 \times \tau_2$. Let $F(x)$ denote the 3×2 matrix $[d\varphi \tau_1, d\varphi \tau_2](x)$, where the 3D columns are expressed in the canonical basis of \mathbb{R}^3 , and consider energies of the form

$$\int_M W(x, F(x)) d\text{vol}_M(x).$$

Material independence requires that W is invariant when F is multiplied on the right by a 2D rotation matrix, and this implies that W only depends on FF^T . To obtain the expression of the metric, we let $\varphi(x) = x + v(x)$ and make a first-order expansion in v of FF^T with $F = [\tau_1 + dv\tau_1, \tau_2 + dv\tau_2]$ yielding

$$FF^T \simeq \pi_{\mathcal{M}} + \pi_{\mathcal{M}} dv^T + dv\pi_{\mathcal{M}}$$

where $\pi_{\mathcal{M}} = \tau_1\tau_1^T + \tau_2\tau_2^T$ is the orthogonal projection on the tangent plane to \mathcal{M} at x . The Riemannian elastic metric should therefore be taken as a quadratic form of $\eta_{\mathcal{M}} := (\pi_{\mathcal{M}} dv^T + dv\pi_{\mathcal{M}})/2$. Expressing this operator in the basis (τ_1, τ_2, ν) , one sees that it depends on the 5 quantities $a_{11} = \tau_1^T dv\tau_1$, $a_{22} = \tau_2^T dv\tau_2$, $a_{12} + a_{21} = \tau_1^T dv\tau_2 + \tau_2^T dv\tau_1$, $a_{13} = \nu^T dv\tau_1$, and $a_{23} = \nu^T dv\tau_2$, yielding 15 free parameters for the “elastic norm” $\llbracket v \rrbracket_{\mathcal{M}}$. (Like in the previous section, an identification is made between the unparametrized surface $\mathcal{M} = m(M)$ and the equivalence class $[m]$.)

The norm is isotropic if it satisfies $\llbracket Rv \rrbracket_{\mathcal{M}} = \llbracket v \rrbracket_{\mathcal{M}}$ for any 3D rotation that leaves ν invariant. This implies that the matrix $\mathbf{a} = \begin{pmatrix} a_{11} & (a_{12} + a_{21})/2 \\ (a_{12} + a_{21})/2 & a_{22} \end{pmatrix}$ is transformed by a 2D rotation as $\mathbf{a} \mapsto R^T \mathbf{a} R$ and the vector $\mathbf{b} = \begin{pmatrix} \nu^T dv\tau_1 \\ \nu^T dv\tau_2 \end{pmatrix}$ as $\mathbf{b} \mapsto \mathbf{b} R$. Using usual invariance arguments, this requires that the squared norm must be a (quadratic) function of $\text{trace}(\mathbf{a})$, $\text{trace}(\mathbf{a}^2)$, and $|\mathbf{b}|^2$, yielding

$$\llbracket v \rrbracket_{\mathcal{M}}^2 = \int_{\mathcal{M}} \beta(x, \eta_{\mathcal{M}}) d\text{vol}(x) \tag{9}$$

with

$$\beta(x, \eta_{\mathcal{M}}) = \lambda_{\text{tan}} \text{trace}(\mathbf{a})^2 + \mu_{\text{tan}} \text{trace}(\mathbf{a}^2) + \mu_{\text{tsv}} |\mathbf{b}|^2. \tag{10}$$

The three different terms of this metric can be also interpreted as penalties on the changes of local area, metric tensor, and normal vector, respectively, as pointed out in Jermyn et al. (2012) (see also their intrinsic expressions derived in Appendix). Jermyn et al. (2012) focus on the special case $\lambda_{\text{tan}} = 1/16$, $\mu_{\text{tan}} = 0$, $\mu_{\text{tsv}} = 1$, which can be shown to be isometric to a Euclidean metric under a “square root normal transform.”

To consider another example, let $\lambda_{\text{tan}} = 0$ and $\mu_{\text{tan}} = \mu_{\text{tsv}} = 1$. Then

$$\begin{aligned} \beta(x, \eta, \mathcal{M}) &= a_{11}^2 + a_{22}^2 + \frac{1}{2}(a_{12} + a_{21})^2 + a_{13}^2 + a_{23}^2 \\ &= a_{11}^2 + a_{22}^2 + a_{12}^2 + a_{21}^2 + a_{13}^2 + a_{23}^2 - \frac{1}{2}(a_{12} - a_{21})^2 \\ &= \text{trace}(dvdv^T) - \frac{1}{2}(a_{12} - a_{21})^2 \end{aligned} \quad (11)$$

The first term, $\text{trace}(dvdv^T)$, corresponds to the H^1 metric on \mathcal{M} , used, e.g., in Younes (2018b). This metric, without the correction term $\frac{1}{2}(a_{12} - a_{21})^2$, is not an elastic metric. It belongs however to a larger class of metrics, studied in Su et al. (2020), where the correction term is added to Eq. (10) with a fourth parameter.

Such elastic metrics can be used in combination with the LDDMM metric through the hybrid setup described above. As an illustration, Fig. 2 provides a comparison of the geodesic trajectories between two surfaces, obtained with the pure LDDMM model and a hybrid model using the elastic term given by Eq. (11).

The norm in Eq. (9) can also be obtained as a limit of the laminar elastic model of the previous section and the energy in Eq. (8), which is shown in Appendix by also providing an intrinsic expression of the elastic norm. This in part justifies the terminology of elastic metrics given to this framework in the related literature.

Elastic Metrics on Curves

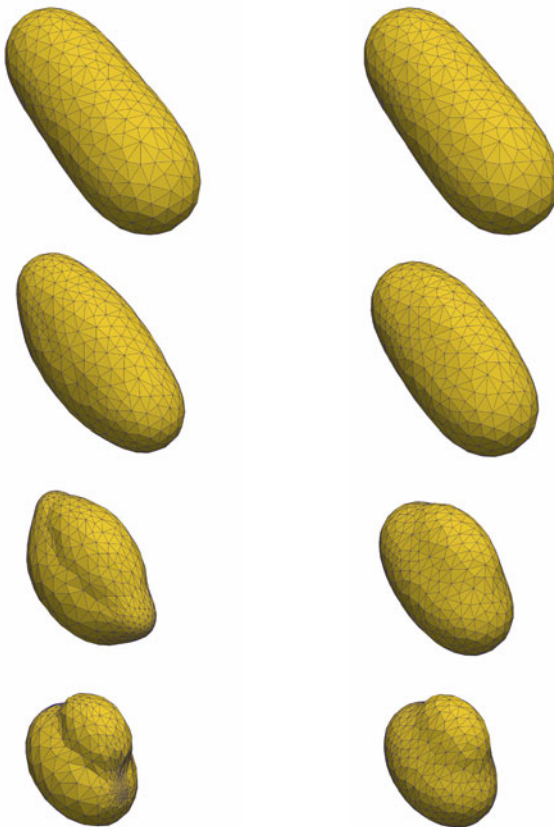
If \mathcal{M} is a 3D curve, the same analysis shows that elastic metrics should depend on the products $\tau^T dv\tau$, $v_1^T dv\tau$, and $v_2^T dv\tau$, where τ is a unit tangent on M and (τ, v_1, v_2) is a continuous positively oriented frame defined along the curve. Denote $\partial_s v = dv\tau$ for the derivative with respect to arc length, as introduced, e.g., in Michor and Mumford (2007). The metric must also be invariant to rotations of the normal frame (v_1, v_2) and changes of orientation on \mathcal{M} , which requires the metric to take the form

$$\llbracket v \rrbracket_{\mathcal{M}}^2 = \int_{\mathcal{M}} \beta(x, \partial_s v) d\text{vol}(x) \quad (12)$$

with

$$\beta(x, \partial_s v) = \mu_{\text{tan}}(\tau^T \partial_s v)^2 + \mu_{\text{tsv}}((v_1^T \partial_s v)^2 + (v_2^T \partial_s v)^2).$$

Fig. 2 Comparison between geodesics between surfaces using a pure LDDMM and a hybrid LDDMM/elastic metric. First column: Four time points of an LDDMM geodesic ($t = 0, t = 0.3, t = 0.7, \text{ and } t = 1$). Second column: Same time points for the hybrid geodesic. One can note a difference in the intermediate shapes and (as indicated by the triangulation) higher local contraction associated with the LDDMM metric. The hybrid metric uses the expression provided in Eq. (11)



The special case of planar curves has been extensively discussed. In this case, letting ν denote the unit normal, the metric has two parameters, with

$$\beta(x, \partial_s \nu) = \mu_{\tan} (\tau^T \partial_s \nu)^2 + \mu_{\text{tsv}} (\nu^T \partial_s \nu)^2.$$

When $\mu_{\tan} = \mu_{\text{tsv}} = 1$, one gets $\beta(x, \partial_s \nu) = |\partial_s \nu|^2$. The resulting metric was introduced in Younes (1996, 1998) and shown to be isometric to a flat metric using a square root transform. This metric was called “ H_0^1 ” in Mumford and Michor (2006) and further studied in Younes et al. (2008). The case $\mu_{\tan} = 1, \mu_{\text{tsv}} = 1/4$ was considered in Mio et al. (2007), Srivastava and Klassen (2016), and a similar square root transform was seen to provide an isometry with a flat space in this case also. This isometry was extended to the general case in Younes (2018a, 2019) and in Needham and Kurtek (2020) (another isometry was also introduced in Bauer et al. 2014a). The reader is referred to the cited references for more details on the exact expression of the isometry. Figure 3 provides an example of geodesic evolution for a hybrid metric, to be compared with Fig. 1.



Fig. 3 Four time points of a geodesic evolution in shape space for a hybrid metric. The initial and final shapes are the same as those in Fig. 1, but one can note, in particular, that the elliptical shapes are better conserved during the motion

Growth Models

Introduction

The previous section described various metrics in shape space that are built as a regularized linearized elastic energy. Optimal paths (i.e., geodesics) associated with these metrics prefer different trajectories from those associated with the “standard” spaces discussed in section “[Shape Spaces Under Diffeomorphic Action](#)” and tend to inherit some of the properties suggested by the elastic intuition. However, not all trajectories of interest need to be geodesics for some metric or satisfy a least-action principle. In particular, including external actions, with in particular possible mechanisms describing growth (Following common terminology, we consider growth as a general shape change mechanism, also including atrophy, as a “negative growth.”), will provide shape analysis methods with additional capability of modeling transformations typically observed in biology or medicine.

A leading model for shape change in the framework of elasticity theory introduces the notion of *morphoelasticity* in which shapes are subject to the action of a “growth tensor,” which partly accounts for the derivative of the deformation, $d\varphi$ (see Goriely (2017) for an extensive introduction to the subject and for references). Letting G denote the growth tensor, one writes $d\varphi = AG$, where A completes the growth tensor to provide a valid differential $d\varphi$, in a way that would minimize the elastic energy (so that one applies the elastic cost to $A^T A - \text{Id}_{\mathbb{R}^d}$ rather than to $d\varphi^T d\varphi - \text{Id}_{\mathbb{R}^d}$). This approach does not necessarily lead to the trivial solution $A = \text{Id}_{\mathbb{R}^d}$ because the growth tensor G is not necessarily “compatible,” i.e., there may not always exist a transformation φ such that $d\varphi^T d\varphi = G^T G$.

Considering small deformations, i.e., linearizing $d\varphi = AG$ for φ and G close to the identity, and writing $\varphi = \text{id}_{\mathbb{R}^d} + v$, $A = \text{Id}_{\mathbb{R}^d} + a$, and $G = \text{Id}_{\mathbb{R}^d} + g$, one gets, simply, $dv = a + g$. So, for a given tensor g , the vector field v must minimize an expression of the form

$$\int_{\Omega} B(x, (dv + dv^T - g - g^T)/2) dx \quad (13)$$

where B was discussed in section “[Three-Dimensional Case](#)”. There is no loss of generality in assuming that g is symmetric, which will be done in the following. The minimum of Eq. (13) is not always zero, i.e., the equation $\frac{dv+dv^T}{2} = g$ does not always have a solution. A necessary condition (which is sufficient when Ω is simply connected) is that $\nabla \times (g \times \nabla g) = 0$ (row-wise curl application, followed by column-wise; see, e.g., Gonzalez and Stuart 2008).

Riemannian Viewpoint

Returning to the Riemannian situation discussed in shape spaces, the metric was defined as $\|v\|_\Omega^2 = \kappa \|v\|_V^2 + \llbracket v \rrbracket_\Omega^2$ with $\llbracket v \rrbracket_\Omega^2$ given by the right-hand side of Eq. (13) with $g = 0$. One can apply the same approach here, letting

$$\llbracket v \rrbracket_\Omega^2 = \inf_g \int_\Omega B(x, (dv + dv^T)/2 - g) dx.$$

Obviously, this definition has little interest unless one restricts the space of growth tensors under consideration (otherwise, $\llbracket v \rrbracket_\Omega^2 = 0$ since one can take $g = (dv + dv^T)/2$). Letting $\mathcal{G}(\Omega)$ denote a set of tensor fields ($x \mapsto g(x) \in \text{Sym}(\mathbb{R}^d)$), one can define

$$\llbracket v \rrbracket_\Omega^2 = \inf_{g \in \mathcal{G}(\Omega)} \int_\Omega B(x, (dv + dv^T)/2 - g) dx$$

which is not trivial in general. If $\mathcal{G}(\Omega)$ forms a vector space, then $\llbracket v \rrbracket_{[\Omega]}$ is a semi-norm on V .

Note that one can also switch the focus to the growth tensor and define, for $g \in \mathcal{G}(\Omega)$,

$$\|g\|_\Omega^2 = \min_{v \in V} \left(\kappa \|v\|_V^2 + \int_\Omega B(x, (dv + dv^T)/2 - g) dx \right),$$

which defines a norm on growth tensors. The introduction of the regularization by the V norm ensures that the minimum is attained at a unique $v \in V$, that one can denote $v_{g,\Omega}$, which depends linearly on g and is such that $\kappa \|v_{g,\Omega}\|_V^2 \leq \|g\|_\Omega^2$. One can therefore consider evolution equations in the form

$$\begin{cases} \partial_t \varphi(t, x) = v_{g(t), \Omega(t)}(\varphi(t, x)) \\ \Omega(t) = \varphi(t, \Omega(0)) \end{cases}$$

which are well posed (starting with $\varphi(0, \cdot) = \text{id}_{\mathbb{R}^3}$) as long as

$$\int_0^1 \|g(t)\|_{\Omega(t)}^2 dt < \infty.$$

This framework therefore provides two formally equivalent optimal control problems. In the first one, one minimizes, with respect to $v(\cdot)$,

$$\int_0^1 \|v(t)\|_{\Omega(t)}^2 dt, \tag{14}$$

subject to $\varphi(1, \Omega_0) = \Omega_1$, $\varphi(0, \cdot) = \text{id}_{\mathbb{R}^3}$, $\partial_t \varphi(t, \cdot) = v(t, \varphi(t, \cdot))$, and $\Omega(t) = \varphi(t, \Omega_0)$. In the second one, one minimizes, with respect to $g(\cdot)$,

$$\int_0^1 \|g(t)\|_{\Omega(t)}^2 dt, \tag{15}$$

subject to $\varphi(1, \Omega_0) = \Omega_1$, $\varphi(0, \cdot) = \text{id}_{\mathbb{R}^3}$, $\partial_t \varphi(t, \cdot) = v_{g(t)}(\varphi(t, \cdot))$, $g(t) \in \mathcal{G}(\Omega(t))$, and $\Omega(t) = \varphi(t, \Omega_0)$. Both problems are, in addition, equivalent to minimizing, with respect to both $v(\cdot)$ and $g(\cdot)$,

$$\kappa \int_0^1 \|v(t)\|_V^2 dt + \int_0^1 \int_{\Omega(t)} B(x, (dv(t, x) + dv(t, x)^T)/2 - g(t, x)) dx \tag{16}$$

subject to $\varphi(1, \Omega_0) = \Omega_1$, $\varphi(0, \cdot) = \text{id}_{\mathbb{R}^3}$, $\partial_t \varphi(t, \cdot) = v(t, \varphi(t, \cdot))$, $g(t) \in \mathcal{G}(\Omega(t))$, and $\Omega(t) = \varphi(t, \Omega_0)$.

When $\mathcal{G}(\Omega)$ is a vector space, the minimum value of these optimal control problems with given Ω_0 and Ω_1 is symmetric in Ω_0 and Ω_1 , and its square root satisfies the triangular inequality. This minimum is always larger to that obtained with $B = 0$ and therefore cannot be zero unless $\Omega_0 = \Omega_1$. (Note that the minimum can be infinite if the problem is unfeasible.) Under suitable assumptions, solutions of this optimal control problem always exist. A precise statement of this result and a sketch of its proof are provided in the appendix.

Growth as an Internal Force

Some additional notation is needed here. Denote the topological dual of a Hilbert space H , with inner product $\langle \cdot, \cdot \rangle_H$, by H^* , and if $\mu \in H^*$ is a linear form and if $h \in H$, denote their pairing by $(\mu | h)$ (i.e., $\mu(h)$). Riesz's representation theorem gives an isometric correspondence between H and H^* with, denoting by $K_H : H^* \rightarrow H$ the operator that associates to a linear form μ the unique vector $h \in H$ such that $(\mu | \tilde{h}) = \langle h, \tilde{h} \rangle_H$ for all $\tilde{h} \in H$, $\|h\|_H^2 = (K_H^{-1}h | h)$. This construction will be applied to $H = V$.

Introduce the (finite-dimensional) linear operator $\beta(x)$ operating on symmetric 3×3 matrices such that $B(x, S) = \langle S, \beta(x)S \rangle$ (with $\langle S, S' \rangle = \text{trace}(SS')$), and

define, for a tensor field $x \mapsto S(x)$,

$$\beta_\Omega(S) = \int_\Omega \beta(x)S(x)dx.$$

Defining $dv = (dv + dv^T)/2$, one has

$$\int_\Omega B(x, (dv + dv^T)/2 - g)dx = (\mathfrak{d}^* \beta_\Omega dv \mid v) - 2(\mathfrak{d}^* \beta_\Omega g \mid v) + (\beta_\Omega g \mid g).$$

Letting $\mathfrak{j}_{g,\Omega} = \mathfrak{d}^* \beta_\Omega g$, one has

$$v_{g,\Omega} = (\kappa K_V^{-1} + \mathfrak{d}^* \beta_\Omega \mathfrak{d})^{-1} \mathfrak{j}_{g,\Omega}. \tag{17}$$

This relation provide an alternative way of modeling the growth process. One can indeed, following Hsieh et al. (2022), directly define a ‘‘yank’’ (derivative of a force) \mathfrak{j} as a control, with $v_{\mathfrak{j}} = (\kappa K_V^{-1} + \mathfrak{d}^* \beta_\Omega \mathfrak{d})^{-1} \mathfrak{j}$, and use the running cost

$$\int_0^1 (\mathfrak{j}(t) \mid v_{\mathfrak{j}(t)})dt$$

with $\partial_t \varphi_{\mathfrak{j}}(t, x) = v_{\mathfrak{j}(t)}(\varphi_{\mathfrak{j}}(t, x))$. One can then show that the finiteness of the cost implies that the ODE has solutions over all time interval. One can also prove that optimal control \mathfrak{j} always exist in this case.

Note that this problem is different from the one described in Eqs. (14), (15) and (16). In that case, one has

$$\|g\|_\Omega^2 = (\beta_\Omega g \mid g) - (\mathfrak{j}_g \mid v_g),$$

showing that the geodesics for the $\|\cdot\|_\Omega$ metric (which remain to be explored) are likely to behave differently than those studied in Hsieh et al. (2022).

A Simple Example

Assume that the growth tensor is scalar, i.e., $g(x) = \rho(x)\text{Id}_{\mathbb{R}^3}$, and that $g(x) = 0$ on $\partial\Omega$, to avoid keeping track of boundary terms. Also assume that the elastic energy on Ω is homogeneous and isotropic (Eq. (7)), which implies that $B(x, g(x))$ is proportional to $\rho(x)^2$, the proportionality constant being, using the Lamé coefficients, equal to $3(3\lambda/2 + \mu)$. Letting $\xi = 3\lambda/2 + \mu$ and using the bilinearity of $B(x, \cdot)$ and the fact that $\text{trace}(dv) = \text{trace}(dv^T) = \nabla \cdot v$, a direct computation yields

$$B(x, (dv + dv^T)/2 - g) = B(x, (dv + dv^T)/2) - 2\xi\rho(x)\nabla \cdot v(x) + 3\xi\rho(x)^2$$

Integrating by parts, one has

$$\int_{\Omega} \rho(x) \nabla \cdot v(x) dx = - \int_{\Omega} \nabla \rho(x)^T v(x) dx$$

so that, using the previous notation,

$$\mathbb{J}_{g,\Omega} = -\xi \nabla \rho.$$

One therefore finds that

$$\|\rho \text{Id}_{\mathbb{R}^3}\|_{\Omega}^2 = 3 \int_{\Omega} \rho(x)^2 dx - \int_{\Omega} \nabla \rho(x)^T (\kappa K_V^{-1} + \mathfrak{d}^* \boldsymbol{\beta}_{\Omega} \mathfrak{d})^{-1} \nabla \rho(x) dx.$$

Similarly, the minimum in ρ of $B(x, (dv(x) + dv(x)^T)/2 - \rho(x) \text{Id}_{\mathbb{R}^3})$ is attained at $\rho = \nabla \cdot v/3$ and

$$\|v\|_{\Omega}^2 = \kappa \|v\|_V^2 + \int_{\Omega} B(x, (dv(x) + dv(x)^T)/2 - \nabla \cdot v(x)/3) dx$$

Growth Due to External Action

Shape variations resulting from a growth tensor as described above may be caused by external effects (e.g., impact of a disease) and do not need to follow a least-action principle such as described in the previous paragraph. More likely, the growth tensor will follow its own course, according to a process influenced by elements that are independent of the material properties of the deforming shape. The growth tensor evolution cannot be completely independent of the shape, however, since it must be supported by the time-dependent domain $\Omega(t)$. It is also possible that changes in the geometry of the shape impact how growth behaves.

All this results in evolution systems with coupled evolution equations, typically involving moving domains. In Bressan and Lewicka (2018), a scalar growth is assumed, with the relationship $\nabla \cdot v = \rho$, consistent with section “A Simple Example”. The growth function depends on another function, u , representing the “concentration of morphogen,” so that $\rho = \alpha \circ u$ for a fixed function α . This morphogen concentration follows a partial differential equation (PDE), namely, $\Delta u = w - u$, with Neumann’s boundary conditions, where w itself is a density advected by the motion, i.e., satisfying $\partial_t w + \nabla \cdot (vw) = 0$, which provides the coupling between growth and shape change. Initial conditions are the initial domain Ω_0 and the initial value of w and w_0 . One can then show that, when starting with a domain Ω_0 with smooth enough boundary and with a smooth enough density w_0 , a solution to the growth system exists over some finite interval $[0, T]$ for some (small enough) T .

In Hsieh et al. (2021, 2022), the additional regularization term $\|v\|_V$ described in this chapter is added, using the formulation in Eq. (17)

$$(\kappa K_V^{-1} + \mathfrak{d}^* \beta_\Omega \mathfrak{d})v = \mathfrak{j},$$

where \mathfrak{j} is the modeled control (as seen in our simple example of section “A Simple Example”, \mathfrak{j} has an interpretation similar to that of $-\nabla \rho$). Hsieh et al. (2022) model \mathfrak{j} as a function $\mathfrak{j}(\varphi, \theta)$, for some time-independent parameter θ , providing coupled equations

$$\begin{cases} \partial_t \varphi(t, x) = v(t, \varphi(t, x)) \\ (\kappa K_V^{-1} + \mathfrak{d}^* \beta_\Omega \mathfrak{d})v(t, \cdot) = \mathfrak{j}(\varphi(t, \cdot), \theta) \end{cases}$$

The system is shown to have a unique solution $t \mapsto \varphi(t, \cdot)$ over arbitrary large time intervals, for any fixed θ , provided that $\mathfrak{j}(\varphi, \theta)$ is Lipschitz in φ for the $(1, \infty)$ norm. Denoting this solution by $\varphi(t, \cdot; \theta)$, this property allows for the specification of optimization problems over the parameter θ involving the transformation $\varphi(1, \cdot; \theta)$.

A more complex system is introduced in Hsieh et al. (2021) in which \mathfrak{j} is itself modeled based on a solution of a “reaction-diffusion-convection” equation on the moving domain Ω . Ignoring a few technicalities, \mathfrak{j} is given by $\mathfrak{j} = \nabla(Q(p))$ where Q is a fixed function and p satisfies

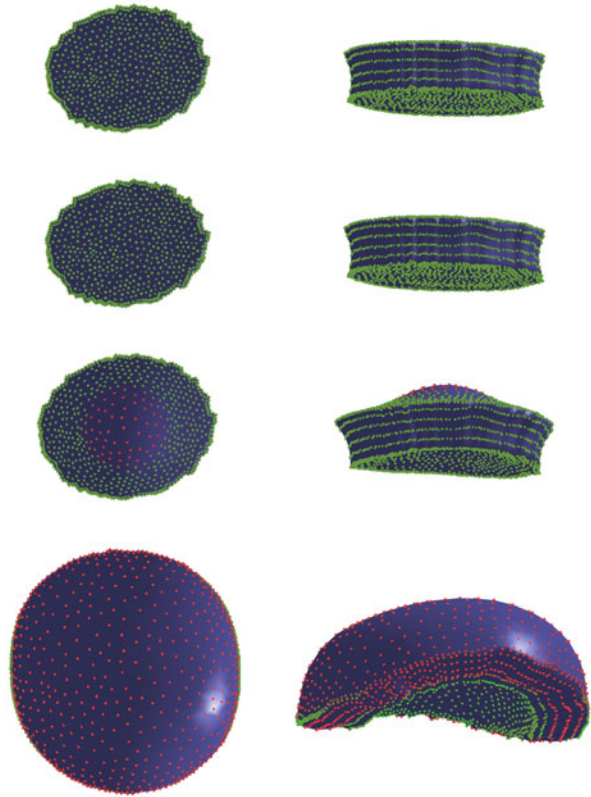
$$\partial_t p = \nabla \cdot (S_\varphi \nabla p - pv) + R(p)$$

where R , the reaction function, is fixed and S_φ , the diffusion matrix, is allowed to evolve with the transformation φ . One can then formulate suitable conditions under which the system

$$\begin{cases} \partial_t \varphi(t, x) = v(t, \varphi(t, x)) \\ (\kappa K_V^{-1} + \mathfrak{d}^* \beta_\Omega \mathfrak{d})v(t, \cdot) = \nabla(Q(p)) \\ \partial_t p = \nabla \cdot (S_\varphi \nabla p - pv) + R(p) \end{cases} \tag{18}$$

has solutions over arbitrary time intervals for a given initialization $p_0 = p(0, \dots)$. The determination of this initial condition for an optimal behavior at time 1 is tackled in Hsieh (2021), where the existence of solutions of the optimization problem is shown. Figure 4 provides an example of growth process obtained as solution of this system.

Fig. 4 Growth model from Hsieh (2021) applied to a 3D volume. Dots are colored proportionally to the magnitude of p in Eq. (18). Rows 1 to 4 provide two views of the evolving shapes at times $t = 0$, $t = 0.33$, $t = 0.67$, and $t = 1.0$. (Images generated from code developed by Dai-Ni Hsieh)



Constraints, Deformation Modules, and Other Growth Models

Specific behavior can be enforced in a deformation process by constraining the values of the vector field at given locations in the shape. Theoretical bases for constrained and sub-Riemannian versions of LDDMM were introduced in Arguillère et al. (2014, 2015), Arguillère and Trélat (2017), and a survey of such methods is provided in Younes et al. (2020). Among such approaches, deformation modules (Gris et al. 2018; Lacroix et al. 2021) offer a generic framework in which various types of behaviors can be defined by combining suitable constraints in a modular manner. Referring to the publications above for more details, the example of “implicit elastic modules” is closely related to this chapter’s discussion. For such modules, the vector field v is obtained as a minimizer of

$$v \mapsto \lambda \|v\|_V^2 + \sum_{k=1}^m |\varepsilon_v(x_k) - S_k(h_k)|^2$$

where $\varepsilon_v = (dv + dv^T)/2$, x_1, \dots, x_N are control points that are attached to (and move together with) the evolving shape and $h \mapsto S_k(h)$ are symmetric matrices, parametrized by a control h , inducing a desired behavior (e.g., dilation) near the control points. This norm therefore introduces a finite set of (soft) constraints on the strain tensor.

A different approach at modeling growth can be found in Kaltenmark (2016) and Kaltenmark and Trouvé (2019). In this work, a growing shape at a given time t is defined as a transformation q_t of a co-dimension-one foliation X , which encodes the full growth process. During the evolution, only the restriction of q_t to the set X_t formed by leaves at time $s \leq t$ of the foliation is relevant to describe the growing shape. The value of $q_t(x)$ remains constant until t reaches the foliation index of x , so that the function q_0 encodes all future initializations of the growth process. This process can be constructed through an evolution equation in the form $\partial_t q_t = v(t, q_t)$, and an example is developed in Kaltenmark and Trouvé (2019) to model animal horn growth.

Conclusion

Starting from the notion of shape spaces built along the principles of Grenander's metric pattern theory and the action of diffeomorphism groups, this chapter surveyed a few recent efforts to incorporate physical constraints in the modeling of trajectories in such spaces. It first discussed the class of hybrid models that consist in combining the original shape space metric induced by the deformation group with other more physically informed metrics, in particular those derived from linear elasticity theory. A second general approach is to further constrain shape evolution via the introduction of a growth model underlying the morphological transformation.

One of the main motivation behind all of these works is to advance the ability of shape space frameworks to model physical or biological processes while still preserving the advantages of the geometric shape space metric setting. Indeed, this enables the formulation of the dynamics of those processes as control systems and provides adequate regularization norms to ensure existence and smoothness of solutions in many cases. Furthermore, by considering the associated optimal control problems, those same models can often lead to natural and well-posed approaches for tackling the inverse problem of, e.g., determining the causes/sources of morphological changes based on some observed shape evolution. The ideas provided by the present chapter are examples of emerging efforts toward cross-fertilization between the fields of shape analysis, mathematical biology, biomedical engineering, and material science.

Appendix A: Elastic Surface Metric as the Limit of the Laminar Model (section “Elastic Metrics on Surfaces”)

Given an oriented surface \mathcal{M}_0 in \mathbb{R}^3 , and its unit normal vector field denoted v_0 , one can generate a foliated 3D volume as the set of points $\Phi(s, x_0) = x_0 + s\delta v_0(x_0)$, $x_0 \in \mathcal{M}_0, s \in [0, 1]$, and Φ is a diffeomorphism for small enough $\delta > 0$. In this case, the unit normal N to the layer $\mathcal{M}_s = \Phi(\{s\} \times \mathcal{M}_0)$ at the point $x = \Phi(x_0, s) \in \Omega$ is also $N(x) = v_0(x_0)$. It coincides, up to a factor δ , with $S = \partial_s \Phi$ and satisfies $dNN = 0$. Let $v_0 : \mathcal{M}_0 \rightarrow \mathbb{R}^3$ be a vector field on \mathcal{M}_0 , and define its extension v to Ω by $v(\Phi(s, x_0)) = v_0(x_0)$, so that v satisfies $dvN = 0$. See Fig. 5 for an illustration. Let $\sigma_0 = dv_0$ denote the shape operator on the surface \mathcal{M}_0 and similarly σ_s the shape operator of layer \mathcal{M}_s , i.e., the restriction of dN to the tangent space of \mathcal{M}_s . Recall that the shape operator on a surface is a symmetric operator.

Write $v = v_T + v_N N$ where $v_T \in \mathbb{R}^3$ is tangent to the layers and v_N is scalar. If τ and $\tilde{\tau}$ are vectors tangent to the layers, we have

$$\tilde{\tau}^T dv\tau = \tilde{\tau}^T dv_T\tau + (\nabla_{v_N}^T \tau)(\tilde{\tau}^T N) + v_N \tilde{\tau}^T dN\tau = \tilde{\tau}^T dv_T\tau + v_N \tilde{\tau}^T dN\tau \quad (19)$$

In particular, letting $\varepsilon_T = (dv_T + dv_T^T)/2$, and for $\{\tau_1, \tau_2\}$ an orthonormal basis of the tangent plane to \mathcal{M}_s at x , one has

$$\tau_1^T \varepsilon \tau_1 + \tau_2^T \varepsilon \tau_2 = \tau_1^T dv_T \tau_1 + \tau_2^T dv_T \tau_2 + v_N [\tau_1^T dN \tau_1 + \tau_2^T dN \tau_2].$$

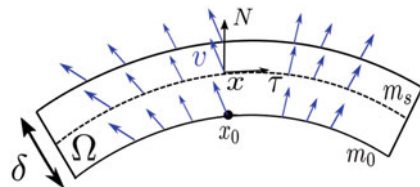
With the sum of the first two terms, one recognizes the divergence of v_T on the surface \mathcal{M}_s which will be denoted by $\nabla_{\mathcal{M}_s} \cdot v_T$. Similarly, the term within brackets is the divergence of the shape operator on \mathcal{M}_s which equals $-2H_{\mathcal{M}_s}$ where $H_{\mathcal{M}_s}$ is the mean curvature of \mathcal{M}_s . Therefore, one deduces that, on \mathcal{M}_s :

$$\tau_1^T \varepsilon \tau_1 + \tau_2^T \varepsilon \tau_2 = \nabla_{\mathcal{M}_s} \cdot v_T - 2v_N H_{\mathcal{M}_s}.$$

Moreover, as $dvN = 0$, it follows that $N^T \varepsilon N = 0$ and thus, on \mathcal{M}_s :

$$\text{trace}(\varepsilon) = \tau_1^T \varepsilon \tau_1 + \tau_2^T \varepsilon \tau_2 + N^T \varepsilon N = \nabla_{\mathcal{M}_s} \cdot v_T - 2v_N H_{\mathcal{M}_s}.$$

Fig. 5 Cross-sectional schematic representation of the thin shell layered elastic domain with the deformation field v in blue



Similarly, looking at the second term in Eq. (8) and using Eq. (19), one has

$$\begin{aligned}
 & (\tau_1^T \varepsilon \tau_1)^2 + (\tau_2^T \varepsilon \tau_2)^2 + 2(\tau_1^T \varepsilon \tau_2)^2 \\
 &= (\tau_1^T dv_T \tau_1 + v_N \tau_1^T \sigma_s \tau_1)^2 + (\tau_2^T dv_T \tau_2 + v_N \tau_2^T \sigma_s \tau_2)^2 + 2(\tau_1^T \varepsilon_T \tau_2 + v_N \tau_1^T \sigma_s \tau_2)^2 \\
 &= (\tau_1^T dv_T \tau_1)^2 + (\tau_2^T dv_T \tau_2)^2 + 2(\tau_1^T \varepsilon_T \tau_2)^2 \\
 &\quad + v_N^2 \left[(\tau_1^T \sigma_s \tau_1)^2 + (\tau_2^T \sigma_s \tau_2)^2 + 2(\tau_1^T \sigma_s \tau_2)^2 \right] \\
 &\quad + 2v_N \left[(\tau_1^T dv_T \tau_1)(\tau_1^T \sigma_s \tau_1) + (\tau_2^T dv_T \tau_2)(\tau_2^T \sigma_s \tau_2) + 2(\tau_1^T \varepsilon_T \tau_2)(\tau_1^T \sigma_s \tau_2) \right].
 \end{aligned}$$

In this computation, one uses the fact that the operator dN restricted to the to the tangent space to \mathcal{M}_s at x (i.e., the space spanned by τ_1 and τ_2) coincides with σ_s . Now, by symmetry, one has $\tau_1^T dv_T \tau_1 = \tau_1^T \varepsilon_T \tau_1$ and $\tau_2^T dv_T \tau_2 = \tau_2^T \varepsilon_T \tau_2$. Moreover, recalling that for any 2×2 symmetric tensors ω and $\tilde{\omega}$, one has $\text{trace}(\omega \tilde{\omega}) = \omega_{1,1} \tilde{\omega}_{1,1} + \omega_{2,2} \tilde{\omega}_{2,2} + 2\omega_{1,2} \tilde{\omega}_{1,2}$, one gets

$$\begin{aligned}
 (\tau_1^T \varepsilon \tau_1)^2 + (\tau_2^T \varepsilon \tau_2)^2 + 2(\tau_1^T \varepsilon \tau_2)^2 &= \text{trace}(\varepsilon_T^2) + v_N^2 \text{trace}(\sigma_s^2) + 2v_N \text{trace}(\varepsilon_T \sigma_s) \\
 &= \text{trace}((\varepsilon_T + v_N \sigma_s)^2).
 \end{aligned}$$

Now, using the symmetry of ε and the fact that $N^T \varepsilon N = 0$:

$$N^T \varepsilon^2 N = |\varepsilon N|^2 = (\tau_1^T \varepsilon N)^2 + (\tau_2^T \varepsilon N)^2. \tag{20}$$

If τ is tangent to the layers, one has $\tau^T dv_N = 0$ and

$$\begin{aligned}
 \tau^T dv^T N &= N^T dv \tau = N^T dv_T \tau + (\nabla v_N^T \tau)(N^T N) + v_N N^T dN \tau \\
 &= N^T dv_T \tau + \nabla v_N^T \tau.
 \end{aligned} \tag{21}$$

Moreover, since $N^T v_T = 0$, it follows that $N^T dv_T \tau = -v_T^T dN \tau$. Using this together with Eq. (20), with Eq. (21), and with the fact that dN is symmetric, one deduces that

$$\begin{aligned}
 N^T \varepsilon^2 N - (N^T \varepsilon N)^2 &= ((-dN^T v_T + \nabla v_N)^T \tau_1)^2 + ((-dN^T v_T + \nabla v_N)^T \tau_2)^2 \\
 &= |-\sigma_s v_T + \nabla_{\mathcal{M}_s} v_N|^2
 \end{aligned}$$

where $\nabla_{\mathcal{M}_s}$ is the gradient operator on \mathcal{M}_s .

Based on all the above expressions, one can finally rewrite Eq. (8) at $x = \Phi(x_0, s)$ as

$$\begin{aligned}
 B(x, \varepsilon) &= \lambda_{\tan} \left(\nabla_{\mathcal{M}_s} \cdot v_T - 2H_{\mathcal{M}_s} v_N \right)^2 + \mu_{\tan} \text{trace}((\varepsilon_T + v_N \sigma_s)^2) \\
 &\quad + 2 \mu_{\text{ang}} |-\sigma_s v_T + \nabla_{\mathcal{M}_s} v_N|^2,
 \end{aligned} \tag{22}$$

and using by a change of variables in the integral expression of the energy, one further has

$$\frac{1}{\delta} \int_{\Omega} B(x, \varepsilon) dx = \frac{1}{\delta} \int_0^1 \int_{\mathcal{M}_0} B(x_0 + s\delta v_0, \varepsilon) |J_{\Phi}(s, x_0)| d\text{vol}_{m_0}(x_0) ds$$

where $|J_{\Phi}(s, x_0)|$ denotes the Jacobian determinant of Φ at (s, x_0) . As $\partial_s \Phi(s, x_0) = \delta v_0(x_0)$ and $d_{x_0} \Phi(s, x_0) = \text{Id} + s\delta d v_0(x_0)$, one gets $d_{x_0} \Phi(0, x_0) = \text{Id}$ where Id denotes here the identity on the tangent space to m_0 at x_0 . Therefore, $|J_{\Phi}(0, x_0)| = \delta$ for all $x_0 \in m_0$. Consequently, taking the limit $\delta \rightarrow 0$ in the above and using the continuity of B and J_{Φ} lead to the following expression of the elastic metric on the surface \mathcal{M}_0 :

$$\llbracket v \rrbracket_{\mathcal{M}_0}^2 = \int_{m_0} B(x_0, \varepsilon) d\text{vol}_{\Phi_0}(x_0)$$

with B given by Eq. (22) (with $s = 0$). Furthermore, it can be easily checked, based on their expressions in the frame (τ_1, τ_2, N) , that the three terms in $B(x_0, \varepsilon)$ correspond precisely, up to multiplicative constants, to the ones of Eq. (10), thus showing that the elastic metric in Eq. (9) can be also recovered as the thin shell limit of the 3D laminar model introduced in section “Three-Dimensional Case”.

Appendix B: Existence of Optimal Paths (section “Riemannian Viewpoint”)

Considering the minimization problem introduced in Eqs. (14) to (16), this section proves that, under suitable assumptions, optimal solutions exist. These assumptions are as follows.

- (1) Let $p \geq 1$. The Hilbert space V is continuously embedded in the Banach space $C_0^p(\mathbb{R}^3, \mathbb{R}^3)$ of p times continuously differentiable vector fields that vanish (with their first p derivatives) at infinity, with the norm

$$\|v\|_{p,\infty} = \sum_{k=0}^p \max\{|d^k v(x)| : x \in \mathbb{R}^3\}.$$

- (2) V is also continuously embedded in $H^1(\mathbb{R}^3, \mathbb{R}^3)$, the Sobolev space of square-integrable functions with square-integrable first derivatives.
- (3) The mapping $x \mapsto B(x, \cdot)$ from \mathbb{R}^3 to the set of positive semi-definite quadratic forms is continuous in x . In particular, $|B(x, \cdot)|$ is bounded on compact subsets of \mathbb{R}^3 .
- (4) There exists a constant c such that $B(x, S) \geq c|S|^2$ for all $S \in \text{Sym}$ and all $x \in \mathbb{R}^3$.

- (5) The sets $\mathcal{G}(\Omega)$, defined over compact subsets of $\Omega \subset \mathbb{R}^3$, satisfy the following conditions.
 - (5)-i If $\Omega \subset \tilde{\Omega}$, then $\mathcal{G}(\Omega) \subset \mathcal{G}(\tilde{\Omega})$.
 - (5)-ii Define, for $\delta > 0$, $\Omega^\delta = \{x : \text{dist}(x, \Omega) \leq \delta\}$. Then $\bigcap_{\delta>0} \mathcal{G}(\Omega^\delta) = \mathcal{G}(\Omega)$.
 - (5)-iii $\mathcal{G}(\Omega)$ is a strongly closed convex subset of $H_{Sym} := L^2(\mathbb{R}^3, \text{Sym}(\mathbb{R}^3))$.

For example, the sets $\mathcal{G}(\Omega) = \{g \text{ Id}_{\mathbb{R}^3} : g \in L^2(\Omega)\}$ satisfy condition (5).

Making these assumptions, let $v_n(\cdot) \in L^2([0, 1], V)$ and $g_n \in L^2([0, 1], H_{Sym})$ be minimizing sequences for the considered problem. To shorten notation, let $\varepsilon_n = (dv_n + dv_n^T)/2$. Because v_n is bounded in $L^2([0, 1], V)$, one can replace it by a subsequence that converges weakly to some v in that space, and using arguments developed in Dupuis et al. (1998), Trouvé (1995), and Younes (2019), the flows φ_n associated with v_n converge uniformly in time and uniformly on compact sets in space to the flow φ associated with v . From weak convergence and weak lower semicontinuity of the norm, one has

$$\int_0^1 \|v\|_V^2 dt \leq \liminf \int_0^1 \|v_n\|_V^2 dt$$

and from the convergence of the flows, one has $\varphi(1, \Omega_0) = \Omega_1$ because this holds for each φ_n .

Based on the assumptions made on B , one has, for all $x \in \mathbb{R}^3$ and $t \in [0, 1]$:

$$\begin{aligned} c|g_n(t, x)|^2 \leq B(x, g_n(t, x)) &\leq \left(B(x, \varepsilon_n(t, x) - g_n(t, x))^{1/2} + B(x, \varepsilon_n(t, x))^{1/2} \right)^2 \\ &\leq 2 \left(B(x, \varepsilon_n(t, x) - g_n(t, x)) + B(x, \varepsilon_n(t, x)) \right) \end{aligned}$$

Because of the convergence of φ_n , there exists a compact set $\tilde{\Omega} \subset \mathbb{R}^d$ that contains all the $\Omega_n(t)$, $n \in \mathbb{N}$, $t \in [0, 1]$. This implies that there exist constants C, C' such that, for all $n \in \mathbb{N}$ (using the boundedness of $B(x, \cdot)$ on compact sets):

$$\int_0^1 \int_{\Omega_n(t)} B(x, \varepsilon_n(t, x)) dx dt \leq C \int_0^1 \int_{\Omega_n(t)} |\varepsilon_n(t, x)|^2 dx dt \leq C' \int_0^1 \|v_n(t)\|_{H^1}^2 dt.$$

By the continuous embedding of V into H^1 , $\|v_n(t)\|_{H^1}$ is bounded up to a multiplicative constant by $\|v_n(t)\|_V$, which implies that the above term is bounded independent of n . The same holds for

$$\int_0^1 \int_{\Omega_n(t)} B(x, \varepsilon_n - g_n) dx dt = \frac{1}{4} \int_0^1 \int_{\Omega_n(t)} B(x, dv_n + dv_n^T - 2g_n) dx dt$$

as (v_n, g_n) is a minimizing sequence for the functional in Eq. (16). This implies that the sequence $\int_0^1 \|g_n\|_{H_{Sym}}^2 dt$ is bounded and that one can assume, using a subsequence if needed, that $g_n \rightharpoonup g$ in $L^2([0, 1], H_{Sym})$.

It remains to prove that $g(t) \in \mathcal{G}(\Omega(t))$ to show that (v, g) provides a solution of the minimization problem. Fixing $\delta > 0$, one can restrict the minimizing sequence to those large enough n for which $\max\{|\varphi_n(t, x) - \varphi(t, x)|, t \in [0, 1], x \in \bar{\Omega}\} < \delta$, so that $\Omega_n(t) \subset \Omega^\delta(t)$ for all n and t .

Let

$$\Gamma(\Omega(\cdot), \delta) = \{\tilde{g}(\cdot) : \tilde{g}(t) \in \mathcal{G}(\Omega^\delta(t)), \text{ for a.e } t \in [0, 1]\},$$

so that $g_n \in \Gamma(\Omega(\cdot), \delta)$. This is a convex set, which follows directly from our hypotheses on the sets $\mathcal{G}(\Omega)$, and it is closed in $L^2([0, 1], H_{Sym})$. Indeed, if $\tilde{g}_n \in \Gamma(\Omega(\cdot), \delta)$ converges to $\tilde{g} \in L^2([0, 1], H_{Sym})$, then a subsequence converges for almost all $t \in [0, 1]$, and since each $\mathcal{G}(\Omega^\delta(t))$ is closed in H_{Sym} it results that $\tilde{g}(t) \in \mathcal{G}(\Omega^\delta(t))$ for almost all t . Now, as strongly closed convex sets are also weakly closed in $L^2([0, 1], H_{Sym})$ (see Hytönen et al. 2016), one deduces from $g_n \rightharpoonup g$ that $g \in \Gamma(\Omega(\cdot), \delta)$. Since this is true for all $\delta > 0$, one has, taking a sequence $\delta_n \rightarrow 0$, that $g(t) \in \mathcal{G}(\Omega(t))$ for almost all $t \in [0, 1]$.

This concludes the proof that (v, g) is a minimizer of Eq. (16).

References

- Arguillere, S., Trélat, E.: Sub-Riemannian structures on groups of diffeomorphisms. *J. Inst. Math. Jussieu* **16**(4), 745–785 (2017). Cambridge University Press
- Arguillère, S., Trélat, E., Trounev, A., Younes, L.: Shape deformation and optimal control. *ESAIM: Proc. Surv.* **45**, 300–307 (2014). EDP Sciences
- Arguillère, S., Trélat, E., Trounev, A., Younes, L.: Shape deformation analysis from the optimal control viewpoint. *Journal de mathématiques pures et appliquées* **104**(1), 139–178 (2015). Elsevier Masson
- Arnold, V.I.: Sur un Principe Variationnel pour les Ecoulements Stationnaires des Liquides Parfaits et ses Applications aux Problèmes de Stabilité non linéaires. *J. Mécanique* **5**, 29–43 (1966)
- Arnold, V.I.: *Mathematical Methods of Classical Mechanics*. Springer, New York, NY, (1978)
- Arnold, V.I., Khesin, B.A.: *Topological Methods in Hydrodynamics*, vol. 125. Springer Nature, New York, NY, (2021)
- Aronszajn, N.: Theory of reproducing kernels. *Trans. Am. Math. Soc.* **68**, 337–404 (1950)
- Bauer, M., Bruveris, M., Marsland, S., Michor, P.W.: Constructing reparameterization invariant metrics on spaces of plane curves. *Diff. Geom. Appl.* **34**, 139–165 (2014a). Elsevier
- Bauer, M., Bruveris, M., Michor, P.W.: Overview of the geometries of shape spaces and diffeomorphism groups. *J. Math. Imaging Vis.* **50**(1–2), 60–97 (2014b). Springer
- Bauer, M., Charon, N., Younes, L.: Metric registration of curves and surfaces using optimal control. In: *Handbook of Numerical Analysis*, vol 20, pp 613–646. Elsevier (2019)
- Bauer, M., Harms, P., Preston, S.C.: Vanishing distance phenomena and the geometric approach to SQG. *Archive Ration. Mech. Anal.* **235**(3), 1445–1466 (2020). Springer
- Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI* **24**(24), 509–522 (2002)

- Berkels, B., Effland, A., Rumpf, M.: Time discrete geodesic paths in the space of images. *SIAM J. Imaging Sci.* **8**(3), 1457–1488 (2015). <https://doi.org/10.1137/140970719>
- Bressan, A., Lewicka, M.: A model of controlled growth. *Archive Ration. Mech. Anal.* **227**(3), 1223–1266 (2018). ISSN 1432-0673
- Bronstein, A., Bronstein, M., Bruckstein, A., Kimmel, R.: Analysis of two-dimensional non-rigid shapes. *Int. J. Comput. Vis.* **78**(1), 67–88 (2008a). ISSN 09205691
- Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-rigid Shapes*. Springer Science & Business Media, New York, NY, (2008b)
- Cao, Y., Miller, M.I., Winslow, R.L., Younes, L.: Large deformation diffeomorphic metric mapping of vector fields. *IEEE Trans. Med. Imaging* **24**(9), 1216–1230 (2005). IEEE
- Cao, Y., Miller, M.I., Mori, S., Winslow, R.L., Younes, L.: Diffeomorphic matching of diffusion tensor images. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), p. 67. IEEE (2006)
- Charon, N., Charlier, B., Glaunès, J., Gori, P., Roussillon, P.: Fidelity metrics between curves and surfaces: currents, varifolds, and normal cycles. In: *Riemannian Geometric Statistics in Medical Image Analysis*, pp. 441–477. Elsevier (2020)
- Christensen, G.E., Rabbitt, R.D., Miller, M.I.: Deformable templates using large deformation kinematics. *IEEE Trans. Image Proc.*, **5**(10), 1435–1447, (1996)
- Ciarlet, P.G.: *Three-Dimensional Elasticity*, vol. 20. Elsevier (1988)
- Dryden, I.L., Mardia, K.V.: *Statistical Shape Analysis: With Applications in R*, vol. 995. Wiley (2016)
- Dupuis, P., Grenander, U., Miller, M.I.: Variational problems on flows of diffeomorphisms for image matching. *Q. Appl. Math.* **LVI**(4), 587–600 (1998)
- Ebin, D.G., Marsden, J.E.: Groups of diffeomorphisms and the motion of an incompressible fluid. *Ann. Math.* **92**, 102–163 (1970)
- Glaunès, J., Trouvé, A., Younes, L.: Diffeomorphic matching of distributions: a new approach for unlabelled point-sets and sub-manifolds matching. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 2, p. II. IEEE (2004)
- Glaunès, J., Qiu, A., Miller, M.I., Younes, L.: Large deformation diffeomorphic metric curve matching. *Int. J. Comput. Vis.* **80**(3), 317–336 (2008)
- Gonzalez, O., Stuart, A.M.: *A First Course in Continuum Mechanics*, vol. 42. Cambridge University Press (2008)
- Goriely, A.: *The Mathematics and Mechanics of Biological Growth*, vol. 45. Springer, New York, (2017)
- Grenander, U.: *General Pattern Theory*. Oxford Science Publications (1993)
- Grenander, U., Keenan, D.M.: On the shape of plane images. *Siam J. Appl. Math.* **53**(4), 1072–1094 (1991)
- Grenander, U., Miller, M.I.: Computational anatomy: an emerging discipline. *Q. Appl. Math.* **56**(4), 617–694 (1998)
- Gris, B., Durrleman, S., Trouvé, A.: A sub-Riemannian modular framework for diffeomorphism-based analysis of shape ensembles. *SIAM J. Imaging Sci.* **11**(1), 802–833 (2018). Society for Industrial and Applied Mathematics
- Gu, X.D., Yau, S.-T.: *Computational Conformal Geometry*, vol. 1. International Press Somerville (2008)
- Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.-T.: Genus surface, z. conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* **23**(8), 949–958 (2004)
- Holm, D.D., Marsden, J.E., Ratiu, T.S.: The Euler–Poincaré equations and semidirect products with applications to continuum theories. *Adv. Math.* **137**(1), 1–81 (1998)
- Hsieh, D.-N.: On model-based diffeomorphic shape evolution and diffeomorphic shape registration. PhD thesis, Johns Hopkins University (2021)
- Hsieh, D.-N., Arguillère, S., Charon, N., Miller, M.I., Younes, L.: A model for elastic evolution on foliated shapes. In: *International Conference on Information Processing in Medical Imaging*, pp. 644–655. Springer, Cham (2019)

- Hsieh, D.-N., Arguillère, S., Charon, N., Younes, L.: Diffeomorphic shape evolution coupled with a reaction-diffusion PDE on a growth potential. *Q. Appl. Math.* (2021). ISSN 0033-569X, 1552-4485. <https://doi.org/10.1090/qam/1600>
- Hsieh, D.-N., Arguillère, S., Charon, N., Younes, L.: Mechanistic modeling of longitudinal shape changes: equations of motion and inverse problems. *SIAM J. Appl. Dyn. Syst.* **21**(1), 80–101 (2022). SIAM
- Hytönen, T., Van Neerven, J., Veraar, M., Weis, L.: *Analysis in Banach Spaces*, vol. 12. Springer (2016)
- Jermyn, I.H., Kurtek, S., Klassen, E., Srivastava, A.: Elastic shape matching of parameterized surfaces using square root normal fields. In: *European Conference on Computer Vision*, pp. 804–817. Springer (2012)
- Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., Audiffren, J.: Operator-valued kernels for learning from functional response data. *J. Mach. Learn. Res.* **17**(20), 1–54 (2016)
- Kaltenmark, I.: *Geometrical Growth Models for Computational Anatomy*. PhD thesis, Université Paris-Saclay (ComUE) (2016)
- Kaltenmark, I., Trouvé, A.: Estimation of a growth development with partial diffeomorphic mappings. *Q. Appl. Math.* **77**(2), 227–267 (2019)
- Kendall, D.G.: Shape manifolds, Procrustean metrics and complex projective spaces. *Bull. Lond. Math. Soc.* **16**, 81–121 (1984)
- Klassen, E.P., Srivastava, A., Mio, W., Joshi, S.H.: Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(3), 372–383 (2004). ISSN 0162-8828
- Lacroix, L., Charlier, B., Trouvé, A., Gris, B.: IMODAL: creating learnable user-defined deformation models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12905–12913 (2021)
- Lui, L.M., Zeng, W., Yau, S.-T., Gu, X.: Shape analysis of planar multiply-connected objects using conformal welding. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1384–1401 (2014). IEEE
- Mémoli, F.: Gromov-Hausdorff distances in Euclidean spaces. In: *CVPR Workshop on Nonrigid Shape Analysis* (2008)
- Mémoli, F.: Gromov-wasserstein distances and the metric approach to object matching. *Found. Comput. Math.* **11**(4), 417–487 (2011)
- Michor, P.W., Mumford, D.: Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms. *Doc. Math.* **10**, 217–245 (2005)
- Michor, P.W., Mumford, D.: An overview of the riemannian metrics on spaces of curves using the hamiltonian approach. *Appl. Comput. Harmonic Anal.* **23**(1), 74–113 (2007)
- Miller, M.I., Younes, L.: Group actions, homeomorphisms, and matching: a general framework. *Int. J. Comput. Vis.* **41**(1–2), 61–84 (2001). Kluwer Academic Publishers
- Miller, M.I., Joshi, S.C., Christensen, G.E.: Large deformation fluid diffeomorphisms for landmark and image matching. In: Toga, A. (ed.) *Brain Warping*, pp. 115–131. Academic Press (1999)
- Miller, M.I., Trouvé, A., Younes, L.: Hamiltonian systems and optimal control in computational anatomy: 100 years since D’Arcy Thompson. *Annu. Rev. Biomed. Eng.* **17**, 447–509 (2015) Publisher: Annual Reviews.
- Mio, W., Srivastava, A., Joshi, S.: On shape of plane elastic curves. *Int. J. Comput. Vis.* **73**(3), 307–324 (2007). Springer
- Mumford, D.B., Michor, P.W.: Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.* **8**(1), 1–48 (2006)
- Needham, T., Kurtek, S.: Simplifying transforms for general elastic metrics on the space of plane curves. *SIAM J. Imaging Sci.* **13**(1), 445–473 (2020)
- Ovsjanikov, M., Mérigot, Q., Mémoli, F., Guibas, L.: One point isometric matching with the heat kernel. In: *Computer Graphics Forum*, vol 29-5, pp. 1555–1564. Wiley Online Library (2010)
- Palais, R.S.: *Foundations of Global Non-linear Analysis*. Benjamin, New York (1968)
- Srivastava, A., Klassen, E.P.: *Functional and Shape Data Analysis*. Springer, New York, NY, (2016)

- Su, Z., Bauer, M., Preston, S.C., Laga, H., Klassen, E.: Shape analysis of surfaces using general elastic metrics. *J. Math. Imaging Vis.* **62**(8), 1087–1106 (2020)
- Thompson, D.W.: *On Growth and Form*. Dover Publications, New York, (1917)
- Trouvé, A.: Action de groupe de dimension infinie et reconnaissance de formes. *Comptes Rendus de l'Académie des Sciences. Série I. Mathématique* **321**(8), 1031–1034 (1995). ISSN 0764-4442
- Trouvé, A.: Diffeomorphism groups and pattern matching in image analysis. *Int. J. Comput. Vis.* **28**(3), 213–221 (1998)
- Trouvé, A., Younes, L.: Metamorphoses through lie group action. *Found. Comput. Math.* **5**(2), 173–198 (2005). Springer
- Vaillant, M., Glaunès, J.: Surface matching via currents. In: Christensen, G.E., Sonka, M. (eds.) *Proceedings of Information Processing in Medical Imaging (IPMI 2005)*. Lecture Notes in Computer Science. Springer (2005). Issue: 3565
- Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A continuum mechanical approach to geodesics in shape space. *Int. J. Comput. Vis.* **93**(3), 293–318 (2011). ISSN 1573-1405. <https://doi.org/10.1007/s11263-010-0416-9>
- Younes, L.: A distance for elastic matching in object recognition. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique* **322**(2), 197–202 (1996)
- Younes, L.: Computable elastic distances between shapes. *SIAM J. Appl. Math.* **58**(2), 565–586 (1998). Society for Industrial and Applied Mathematics
- Younes, L.: Elastic distance between curves under the metamorphosis viewpoint. arXiv preprint arXiv:1804.10155 (2018a)
- Younes, L.: Hybrid riemannian metrics for diffeomorphic shape registration. *Ann. Math. Sci. Appl.* **3**(1), 189–210 (2018b)
- Younes, L.: *Shapes and Diffeomorphisms*. Applied Mathematical Sciences, 2nd edn. Springer, Berlin/Heidelberg (2019). ISBN 978-3-662-58495-8. <https://doi.org/10.1007/978-3-662-58496-5>
- Younes, L., Michor, P.W., Shah, J., Mumford, D.: A metric on shape space with explicit geodesics. *Rend. Lincei Math. Appl.* **19**, 25–57 (2008)
- Younes, L., Gris, B., Trouvé, A.: Sub-Riemannian methods in shape analysis. In: *Handbook of Variational Methods for Nonlinear Geometric Data*, pp. 463–495. Springer, Cham (2020)
- Zeng, W., Gu, X.D.: Registration for 3D surfaces with large deformations using quasi-conformal curvature flow. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2457–2464. IEEE (2011)
- Zeng, W., Lui, L.M., Luo, F., Fan-Cheong Chan, T., Yau, S.-T., Gu, D.X.: Computing quasiconformal maps using an auxiliary metric and discrete curvature flow. *Numer. Math.* **121**(4), 671–703 (2012). Springer

Index

A

- AA-model, 319
- Accelerated alternative descent (AAD), 631
- Accelerated proximal gradient (APG)-based method, 628
- Acceleration, 741
- Accuracy requirement, 81
- Active contour model, 429
- Adaptive fuzzy c-means (AFCM), 1205
- Additive operator splitting method (AOS), 435, 446, 459
- Additive Schwarz method, 384
- Additive white Gaussian (AWG) noise, 5, 43, 45, 46, 53
- ADMM-Net, 891–894
- Adversarial loss, 826
- Adversarial regularization, 1141–1146
- AE-OT model, 1701–1703
- AGP flow properties, 1640
- Air bubble detection simulation, 373–374
- Alexandrov theorem, 1662
- Allen-Cahn (AC) equation, 428
- Alternating direction method of multipliers (ADMM), 9, 36–41, 45, 121–122, 140, 146, 153–156, 169, 177, 178, 505, 689, 691, 743, 917, 957, 1206, 1218, 1219, 1232, 1237, 1903
 - black-box modules, 180–181, 198–201
 - denoising modules, image restoration based on, 183–186
 - intricate compression problems, modular strategies for, 191–197
 - lossy compression, operational ratedistortion optimization, 189
 - restoration by compression, 189–191
 - splitting structure, 181–183
 - unconstrained Lagrangian optimizations, 178–180
- Alternating methods of multipliers (ADMM), 1157
- Alternating projection (AP) algorithms, 147–149
- Alzheimer’s disease (AD) classification, 1430–1432
- Ambrosio–Tortorelli approximation, 961
- Amnesic mild cognitive impairment (aMCI), 1432
- Amplitude based metric for Gaussian measurements, 145
- Analytic reconstruction methods, 1186
- Angle distortion, 1683
- Angle-preserving mapping, 1745
- Anisotropic diffusion model, 361
- Anisotropic models, 247–249
- Anisotropic TV regularization, 8
- Annulus conformal map (ACM), 1507
- Applied harmonic analysis, 1096
- Approximation transform, 1859–1861
 - inpainting, 1877–1882
 - level set reconstruction, 1871–1873
 - salt and pepper noise removal, 1873–1878
- Archive images, 872
- Arclength function, 1456
- Area distortion, 1683
- Area-preserving mapping, 1745
- Armijo backtracking line search, 1593
- Armijo backtracking procedure, 1610
- Armijo condition, 82
- Armijo rule, 1618
- Arrow-Hurwich algorithm, 759
- Artifact spread function (ASF), 574
- Artificial intelligence (AI), 696

- Attraction-repulsion functionals, 1799
- Augmentation networks, 1199
- Augmented Lagrangian function, 643
- Augmented Lagrangian method, for total variation related image restoration models, 508–510
- TV- L^2 restoration, 510–515
- high order models, 528–531
- multichannel image restoration, 524–527
- non-quadratic fidelity, 519–523
- numerical experiments, 541–546
- Augmented Lagrangian method (ALM), 743, 1903
- Aujol-Gilboa-Papadakis (AGP), 1639–1640
- Autoencoder, 973, 974, 980, 982, 986, 989, 992, 1302, 1692, 1694
- Automatic colorization methods, 844, 873
- Autoregressive models, 788–790
- Axisymmetric wavelets, 1402
- B**
- Background geometry, 1756
- Background retrieval, 168
- Backprojection algorithm, 1196
- Backprojection operator, 1192
- Backpropagation, 761
- Badshah-Chen selective segmentation model, 485
- Balanced discrepancy principle, 954
- Balancing principle, 954
- Banach manifold, 1358
- Banach space, 390, 708, 733, 744, 916, 1069–1074, 1138, 1590, 1611, 1719, 1794, 1795
- Band-limited shearlets, 1107
- Barzilai-Borwein techniques, 573
- BAT-Fill, 792
- Batch normalization (BN), 887
- layers, 496, 1190
- Bayesian framework, 1025
- Bayesian inversion, 757
- Bayesian posterior distributions, 225
- Bayesian reconstruction, 762
- Bayesian statistical framework, 1236
- Bayesian statistics, 1134
- Bayes rule, 784
- Bayes' theorem, 755
- Beer-Lambert law, 350
- Beltrami coefficient, 685, 1415–1417, 1419, 1421–1423, 1426, 1457–1459, 1489, 1490, 1492, 1494, 1496, 1500, 1505, 1506, 1509, 1511, 1512, 1746, 1748, 1782, 1785
- Beltrami differential, 1491, 1493
- Beltrami equation, 1415, 1417, 1421, 1457, 1488, 1490–1492, 1748, 1782
- Beltrami holomorphic flow (BHF), 1492–1493
- Benamou-Brenier dynamic fluid, 1670–1671
- Benamou-Brenier method, 1665
- Bernoulli distribution mixture model, 1040–1041
- BERT, 792
- Bias field correction, 1204, 1205, 1216, 1225, 1228–1230
- Bias field estimation, 1205–1209, 1211, 1213, 1232
- Biconjugate based scheme, 1868
- Bidirectional texture function (BTF), 1025, 1026, 1028, 1050, 1057, 1058
- compound Markov model, 1030–1031
- illumination invariants, 1053–1054
- local Markov and mixture models, 1042–1049
- measurement, 1029–1030
- multi-spectral/multi-channel image restoration, 1056–1057
- principal Markov model, 1031–1041
- reflectance model, 1028
- texture compression, 1053
- texture editing, 1053
- texture synthesis and enlargement, 1050–1053
- (un)supervised image recognition, 1054–1056
- BiGAN, 1664
- BigBiGAN, 1664
- Biholomorphic maps, 1742
- Bi-level approaches, 959
- Bilevel learning, 1136–1138
- Bilevel optimization, in imaging, 917–919
- alternative optimality conditions, 921–924
- infinite-dimensional case, 924–932
- numerical experiments, 934–938
- patch-dependent and scale-dependent regularization parameter, 936
- scalar regularization parameter, 935
- solution algorithms, 924
- SSIM quality measures, 937
- standard constraint qualification conditions, failure of, 920–921
- total variation Gaussian denoising, 919–924
- validation dataset reconstructions, 938
- Bilinear form, 625, 642
- Bilinearity, 1946
- Binary cross-entropy loss, 701
- Binary integer nonconvex quadratic programming, 658
- Biomedical analysis, 1291

- Black-box modules, ADMM, 180–181
 distributed representations, 198–201
- Blind phase retrieval (BPR), 140, 141, 169, 170
 fast iterative algorithms, 147–167
 mathematical formula, 141–144
 optimization problems and proximal mapping, 145–147
- Block-diagonal matrix-valued function, 33
- Blood vessels, tracking of, 1560–1563
- Blurred signal-to-noise ratio (BSNR), 43, 46, 53
- Bochner integrals, 1798
- Boltzmann distribution likelihood, 1308
- Borel probability measures, 1795
- Born approximation, 279
- Bouligand (B-) stationarity, 922
- Boundary information, 1004
- Bounded linear operator, 916
- Bourgain theorem, 1664
- BourGAN, 1664
- Box constraint, TV-L2 restoration, 516–518
- Box-Cox transformation, 336
- Brain spherical conformal mapping, 1786
- BrainWeb, 1225–1227, 1229, 1230
- Bregman algorithm, 418, 419, 1016
- Bregman distance, 320, 339, 341, 460, 1085, 1139, 1140
- Bregman divergence, 711–713, 718, 720–723, 739, 741, 742
 symmetrised, 725
- Bregman iterations, 98, 104, 111, 115, 120, 460
 as iterative regularisation methods, 106–107
- Bregman Itoh–Abe (BIA) method, 108
- Bregman proximal methods, 98, 99, 108, 127
 incremental & stochastic, 110, 116
- Bregman splitting, 449
- Brenier’s approach, 1668–1669
- Brenier theorem, 1662
- Bridge sampling, 1342
- Broyden’s method, 955
- B-splines, 1371, 1452
- Buddha surface model, 1684, 1685
- Bungert-Hait-Papadakis-Gilboa (BHPG), 1647–1649
- C**
- Caffarelli’s theorem, 1673
- Canny edge, 1846–1848
- Carathéodory’s theorem, 1837
- Cardiac MRI, 1473
- Cartan connections, 1536
- Cauchy-Green strain tensor, 1938
- Cauchy-Riemann equation, 1415, 1488
- Cauchy’s inequality, 713, 727, 1088
- CelebA dataset, 935
- Cell decomposition, 1691
- Chambolle–Lions model, 961, 964
- Chambolle–Pock algorithm, 957, 963
- Chambolle–Pock method, 711, 1389
- Chambolle–Pock network (CP-Net), 898
- Chambolle–Pock scheme, 570
- Chambolle’s algorithm, 1915
- Chan–Vese (CV) active contour model, 492
- Chan–Vese (CV) algorithm, 435
- Chan–Vese (CV) model, 361, 406, 428, 431, 433, 435, 436, 451, 452, 1003, 1013–1014, 1387, 1443, 1446, 1447
- Chan–Vese segmentation model, 449
- Chan–Vese two phase model, 446
- Cheeger cut problem, 1636
- Chest image, 369
- Chrominance, 588–590
- CIELAB color space, 850
- CIEXYZ color space, 850
- Cifar-10 dataset, 88
- Circle packing, 1758
- Circle patterns, 1764–1767
- Clarke (C-) stationary, 924
- Classical ADMM, 628, 629
- Classical continuous shearlet systems, 1101–1103
- Classifier probability score, 493
- CNN-regularization, 1139
- Cocoercivity, 743
- Coded ptychography (CP), 170
- Code parallelization, 579
- Coefficient of variations (CV), 1230, 1231
- Coefficient sequence, 1097
- Coefficient vectors, 642
- Cohen-Gilboa (CG), 1644–1646
- Coherent diffractive imaging (CDI), 140
- Colon conformal flattening, 1787
- Color diffusion, 589–601
- Color imaging, 237, 238, 849
- Color perception, 822
- Colorization
 from dataset, 607–608
 mathematical modeling of, 587–588
 methods, categories, 851
- Compensated convex based transforms
 approximation transform, 1859–1861, 1871–1882
 convex based algorithms, 1862–1864
 Moreau envelope based algorithm, 1865–1867

- Compensated convex based transforms (*cont.*)
 sampled smooth manifolds, intersection of, 1868–1870
 smoothing transform, 1843–1844
 stable multiscale intersection transform, of smooth manifolds, 1853–1856
 stable multiscale medial axis map, 1856–1859
 stable ridge/edge transform, 1844–1853
 upper transform of singleton set of \mathbb{R}^2 , 1867–1868
- Complemented subspace, 1070
- Compound Markov model, 1030–1031
- Compound Markov random field model (CMRF), 1030
- Compressive sensing magnetic resonance imaging (CS-MRI), 880
- Computational conformal geometric methods, for vision, 1754, 1755
 circle patterns, discrete conformal geometry of polyhedral surfaces derived from, 1764–1767
 cosine laws, 1756, 1757
 discrete curvature, 1757, 1758
 discrete surface, 1755
 harmonic maps, 1767–1770
 Hodge decomposition, 1770–1772
 medical imaging, 1785–1787
 of polyhedral surfaces derived from, 1758–1764
 shape space, 1773–1779
 surface registration, 1780–1785
- Computational quasi-conformal geometry, 1485
- Computed tomography (CT), 239, 240, 348, 350, 1066, 1125–1127
 CNN-MAR, 357–360
 image formation and metal artifacts, 350–353
 NMAR, 353–355
 SMAR, 355–357
 volume reconstruction, 364
- Computer-aided design (CAD), 360
- Condat-Vũ method, 743
- Condition number, 1214
- Conditional autoregressive transformer, 830
- Conditional distribution, 780
- Conditional log-likelihood, 785
- Conditional random field (CRF), 1888, 1922
- Conditional variational autoencoders (CVAE), 785, 786
- Cone-adapted continuous shearlet systems, 1103–1104
- Cone-adapted discrete shearlet systems, 1105–1107
- Conformal factor, 1415, 1744
- Conformal maps, 1414–1415, 1488, 1743–1746, 1752
- Conformal parameterization, 1495–1498, 1501–1505, 1507–1508
- Conformal transformation group, 1773
- Conformal welding method, 1663
- Conjugate gradient (CG) method, 1251, 1266, 1267, 1270
- Conjugate operators, 144
- Constant matrix coefficients, 1046
- Constrained minimization problem, 540
- Constraint qualification, 923
- Contextual image processing, 1528
- Continuity equation, 1718
- Continuous right inverse, 1070
- Continuous shearlet systems
 classical, 1101–1103
 cone-adapted, 1103–1104
 wavefront set, resolution of, 1104–1105
- Contour identification, 681
- Contrast transfer function (CTF), 983
- Contrast-to-noise ratio (CNR), 573
- Conventional algorithms based on variational methods, 1003–1010
- Convergence, 1639
 analysis, 515–516, 647, 1078–1079, 1082–1088
 in expectation, 71
 gap, 732–733, 737, 740
 of iterative algorithms, 168
 linear, 729, 731
 properties, 408–410
 rates, 1655
 strong, 729, 730, 737, 740
 theory, 718–733, 1140
 weak, 729, 730, 737, 740
- Convex analysis, 1795
- Convex based algorithms, 1862–1864
- Convex function, 656
- Convex model, 323
- Convex multiphase image segmentation model, 460
- Convex non-convex (CNC) variational models, 9, 11, 27–28, 41, 43
 ADMM, 36–41
 construction of matrix B , 25–26
 FB strategy, for non-separable CNC models, 35–36
 FB strategy, for separable CNC models, 32–35

- non-separable models, 49–57
- separable models, 46–49
- solution components, 29–30
- sparsity-inducing non-separable regularizers, 24
- sparsity-inducing separable regularizers, 16–22
- Convex programming, 156–160
- Convex variational regularization, 1073
- Convolution network, 994, 995
- Convolution neural networks (CNNs), 696
- Convolutional layer, 67, 87
- Convolutional long short term memory (ConvLSTM), 497
- Convolutional neural network (CNN), 67, 350, 358, 359, 369, 370, 375, 752, 854, 886, 887, 889, 896, 898, 904, 932, 1002, 1005, 1012, 1015, 1016, 1075, 1189, 1298, 1300, 1301, 1303, 1305–1307, 1309–1314
 - corrected image, 359
 - prior, 359
 - training, 357–359, 369
- Convolutional neural network based MAR (CNN-MAR), 350, 357, 358, 369, 370, 374, 375
 - performance, 369
 - reference images, 369
 - vs. SMAR, 369–371
- Coorbit theory, 1105
- Coordinate charts, 1742
- Coordinate-descent, 99
- Corresponding analysis based approach, 951
- Corrupted data, 1185
- Cosine laws, 1756, 1757
- Cosmic microwave background radiation (CMB), 778
- Cost function, 6, 1245
- Coupled approaches, 608–617
- Covariance matrices, 1045, 1049
- Covariance matrix, 1256
- Covariance Wiener Filtering (CWF), 973
- Cross-entropy loss, 829
- Cryo-electron microscopy (Cryo-EM) image denoising, 971
 - EMPIAR-10028, real dataset, 983–984
 - EMPIAR-10028, results for, 990
 - evaluation method, 984
 - network architecture and hyperparameter, 986
 - RNAP, results for, 986–990
 - RNAP, simulation dataset, 981–983
- c*-transform, 1804
- Cumulative distribution, 316
- Curvature-based regularity, 1008
- Curvature parameter, 1830
- Curvature tensor, 1540
- D**
- D’Arcy-Thompson’s theory of transformations, 1933
- Data acquisition, 369
 - geometry, 1712
 - system, 4
- Data consistency, 758
- Data-driven reconstruction, 1068–1069
- Data fidelity, 912
- Data-informed (DI) regularization, 1237
 - derivation, 1238–1245
 - deterministic properties, 1250–1254
 - image deblurring, 1256–1265
 - image denoising, 1265
 - relative errors, 1269
 - statistical data-informed inverse framework, 1245–1250
 - statistical properties, 1254–1256
 - X-ray tomography, 1265
- Data manifold distance, 1144
- 3D causal simultaneous autoregressive model (3DCAR), 1042–1046, 1052, 1054, 1057, 1058
- 2-D convolution matrices, 45
- Deblurring, 184, 186, 192, 915, 1256–1265
- Deconvolution, 1238, 1240, 1243, 1244
- Deep active contour network (DACN), 1016
- Deep Bilevel Optimization Neural Networks (BOONet), 933
- Deep conditional generative modeling, 785
- Deep convolutional neural network (DCGAN), 855, 1664
- Deep learning, 124, 127–128, 752, 754, 776, 778, 1074–1075, 1124–1129, 1187–1190, 1199, 1421, 1661, 1662, 1684, 1690
 - based methods, 569
 - colorization methods, 850
 - data domain, 1193
 - image colorization (*see* Image colorization)
 - models, 1474
 - variational models, 1011–1017
- Deep Network Shearlet Edge Extractor (DeNSE), 1127
- Deep neural networks (DNN), 881, 886, 933, 1647, 1709
 - direct method, 342–343
 - indirect method, 339–341

- Deep regularizers, 1138
 - adversarial regularization, 1141–1145
 - regularization properties of learned regularizers, 1138–1141
 - total deep variation, 1145–1149
- DeepFLASH, 1313–1314
- Deformable template
 - deformation operators, 1715–1716
 - general variational formulation, 1716
 - metamorphosis, 1724–1725
 - time discretised data, 1716–1717
- Deformation model, 1464
- Deformation operators, 1715–1716, 1730–1732
- Degradation model, 5
- Delaunay triangulation, 1679, 1680, 1693, 1761, 1762
- Denoising, 914, 915, 924, 1265–1266
 - modules, 183–186
- DenseUNet, 1016
- Density estimation, 779
- Dental image, 369
- Descent inequality, 719, 734
- Deterministic properties, DI regularization, 1250–1254
- Dice similarity coefficient (DSC), 1012
- Dictionary based approaches, 950–953
- Dictionary learning, 1137
- Diffeomorphic mapping, 1291, 1292
 - achievements and applications, 1314–1315
 - autoencoders, 1302
 - challenges, 1315
 - CNNs, 1298
 - FCN, 1300
 - LDDMM, 1293
 - LSTM, 1302
 - problem statement and framework, 1293–1295
 - RNNs, 1302
 - supervised methods, 1296, 1310–1314
 - SVF, 1293
 - U-Net, 1300–1301
 - unsupervised methods, 1295, 1296, 1303–1310
- Diffeomorphism(s), 1415, 1723, 1743, 1931, 1933–1935
 - group, 1374–1375
- Different color spaces, 863
 - colorization results with, 872
- Different luminance-chrominance spaces, 848
- Differentiable linearized alternating direction method of multipliers (D-LADMM), 885
- Differential games, 679
- Diffraction tomography (DT), 274
- Diffusion matrix, 1948
- Diffusion-weighted MRI (DW-MRI), 1569–1571
- Digamma function, 316
- Digital 2D shearlet transform, 1116–1118
- Digital breast tomosynthesis (DBT), 570
 - 3D imaging, 571
- Digital shearlet systems, 1116–1119
- Digital topology, 1451–1455
- Dilation operator, 1101
- 3-Dimensional cone beam CT (CBCT), 360, 364
- Dirac delta, 828
- Dirac measure, 627, 1676, 1678, 1684
- Direct splitting approach, 390
- Dirichlet boundary condition, 625, 1447, 1458
- Dirichlet energy, 1634
- Dirichlet-Neumann method, 382
- Discrepancies, 1797–1803
- Discrete Calderón condition, 1103
- Discrete conformal geometry, of polyhedral surfaces
 - circle patterns, 1764–1767
 - vertex scaling, 1758–1764
- Discrete conformal metrics, 1761
- Discrete convex model, 323
- Discrete curvature, 1757, 1758
- Discrete divergence operator, 508
- Discrete first-order optimality condition, 637
- Discrete Fourier transformation (DFT), 142, 292
- Discrete gradient operator, 507
- Discrete Hessian operator, 528
- Discrete methods, 1439, 1465
- Discrete metrics, 1761
- Discrete natural conformal parameterization (DNCP), 1503
- Discrete probability densities, 781, 828
- Discrete set, 1442
- Discrete shearlet systems
 - band-limited shearlets, 1107
 - compactly supported shearlets, 1107–1108
 - cone-adapted, 1105–1107
 - frame properties, 1106–1108
 - sparse approximation, 1109–1110
- Discrete uniformization, 1762
- Discrete Yamabe flow, 1759
- Discretization, 663
- Discretized functions, 642
- Discretized optimal control problems, 636
- Discriminator, 826
 - class, 979–980

- Disease diagnosis and classification, quasiconformal geometry, 1430–1433
- Disk-type point clouds, 1512
- Distance-driven approach, 563
- Distributed representations, black box modules
 general framework, 198–199
 holographic compression of images, modular optimizations, 199–201
- Distribution
 density, 828, 1045
 matrix t , 1045
- Distribution-based losses, 824, 828
- Diverse inpainting, 780, 796, 804–807
- Diverse structure generator, 789
- 3D moving average model, 1046–1047
- Domain decomposition, 380–381
 non-overlapping, 381–383, 397–406
 non-smooth and non-separable optimization problems, 385–390
 overlapping, 383–385, 392–397
 for preudal total variation, 391–406
 for primal total variation, 406–420
 for smoothed total variation, 390–391
- Domain decomposition methods (DDMs), 164–167
- Double backpropagation, 1148
- Double-nonlinear eigenvalue problem, 1635
- Douglas–Rachford splitting (DRS), 743
- Dropout layer, 87, 1190
- 3D surface registration, 1424–1427
- Duality problem (DP), 1667
- Dual level-set selective segmentation model, 475
- 2D visualization, 992, 994
- Dijkstra’s algorithm, 1915
- Dynamic inverse problems, image reconstruction
 data driven approaches, 1727
 deformable templates, metamorphosis, 1724–1725
 deformation operators, 1732
 flow of diffeomorphisms and intensities, 1723–1724
 learning deformation operators, 1730
 PDE based motion models, 1718–1722
 spatiotemporal inverse problems, 1711–1717
 spatiotemporal reconstruction, with LDDMM, 1725–1727
 temporal modelling, data driven reconstruction without, 1729
- Dynamic programming approach, 1372–1374
- Dynamic surface tracking, 1785
- Dynamic Yamabe flow, 1760
- DZ-model, 321
- E**
- Edge set, 243–245, 261
- Eigenvectors of nonlinear operators, 1634
- Eikonal-type equation, 1446
- Elastic energy, 1008
- Elastic metrics
 on curves, 1941–1942
 on surfaces, 1940
 three-dimensional case, 1937–1939
- Elastic models, 1464
- Elliptic, 712
- Elliptical distribution, 978, 980
- Empirical risk minimization, 761
- Encoder-decoder U-Net deep network, 861
- End-to-end approaches, 854, 855
- Energy minimization, 1205–1209, 1211, 1215, 1217, 1223, 1224, 1228, 1232, 1487, 1505
- Ensemble LPIPS (E-LPIPS), 794
- Entropic regularization, 1806
- Entropy function, 1796
- Entropy minimization, 1229
- EPDiff, 1339
- Equivalent minimization form, 630
- Ergodic, 732, 737
 sequence, 731, 732
- Euclidean Brownian motion, 1342
- Euclidean distance, 826, 1443, 1446, 1447, 1669, 1696, 1800, 1801, 1837, 1842, 1865
- Euclidean geometry, 1762
- Euclidean metric, 1663, 1681, 1930, 1931, 1941
- Euclidean norm, 508, 529, 824, 915, 924, 1829, 1836
- Euclidean space, 145, 507, 1371, 1377, 1669, 1673, 1677, 1690, 1836
- Euclidean triangle, 1755, 1759
- Euler–Arnold equations, 1936
- Euler–Lagrange equation, 319, 340, 341, 391, 430, 431, 436, 446, 447, 470, 472, 541, 946, 959, 1935
- Eulerian derivative, 1591, 1600
- Eulerian noise, 1328, 1333, 1336, 1339
- Eulerian velocity, 1934
- Eulers elastic based model, 536–539
- Euler’s elastic energy, 1008
- Evaluation metrics, 835
- Evidence lower bound (ELBO), 784
- Exemplar-based image colorization, 853
- Exp-model, 322

- Expectation-maximization (EM) algorithm, 1041, 1048, 1049, 1055, 1205
 Expectation-maximization scheme, 853
 Experimental order of convergence (EOC), 666
 Explicit temporal models, 1712–1713
 Exponential curve, 1556–1559
 Extended ptychographic engine (ePIE) algorithm, 149–151
 External regularization, 1558
 Extremal quasi-conformal map, 1418
- F**
- f -divergence, 1796
 Féjer-monotonicity, 719, 737
 Fan-beam geometries, 554
 Fast Fourier transform (FFT), 506, 512, 535, 538
 Fast inexact proximal (FIP) method, 628
 Fast iterative algorithms
 ADMM, 153–156
 AP algorithms, 147–149
 convex programming, 156–160
 ePIE-type algorithms, 149–151
 proximal algorithms, 151–153
 second order algorithm, 160–162
 subspace method, 162–167
 Fast landmark-aligned spherical harmonic parameterization (FLASH), 1498–1500
 Fast numerical methods, 433
 Fast spherical quasi-conformal parameterization (FSQC), 1498, 1501
 Feature reconstruction loss, 826
 Feedforward neural network, 884
 Feld-Aujol-Gilboa-Papadakis (FAGP), 1641–1644
 Fenchel conjugate, 710, 1795–1797, 1809
 Fenchel–Rockafellar duality, 1795
 Fermat’s rule, 510
 FETI approach, 404–406
 Fiber orientation density functions (FODF), 1569–1570
 Field of Experts (FoE) regularizer, 934, 1136
 Filter factor, 1258
 Filtered back projection (FBP), 350, 364, 559
 Filtering, 1192
 Finite difference operators, 44
 Finite dimensional ADMM algorithm, 628
 Finite element approach, 402–404
 Finite element approximation and error estimates, 632–641
 Finite element discretizations, 652
 Finsler models, 1562
 First order Hamilton–Jacobi PDEs and optimization problems, 212–223
 First-order optimality condition, 637
 First-order Polygamma function, 316
 First-order structure-promoting regularizers, 250
 FISTA, 402–404
 Flat samples, 168
 Flexible algorithms for image registration (FAIR), 692
 Forward-backward (FB) minimization algorithms, CNC variational models, 30, 31
 ADMM, 41
 for non-separable CNC models, 35–36
 for separable CNC models, 32–35
 Forward models, 278
 Forward operator, 1712
 Fourier amplification matrix, 458
 Fourier coefficients, 535, 1798
 Fourier diffraction theorem, 286
 Fourier domain, 1102, 1103, 1106, 1111, 1194
 Fourier integral operators (FIO), 753
 Fourier method, 1821, 1822
 Fourier ptychography method (FP), 142
 Fourier transform, 285, 387, 512, 527, 1102, 1108, 1119
 matrix, 4
 Fréchet manifold, 1351
 Fréchet inception distance (FID), 795, 834, 867
 Fractional order TV, 330–331
 Frame operator, 1097
 Frame theory, 1097
 Frequency, 277
 Frequency-resolved optical gating (FROG), 143
 Frobenius norm, 358, 918
 Fubini’s theorem, 1799
 Fuchsian group, 1779
 Full approximation scheme (FAS), 167, 438
 Full waveform inversion (FWI), 275, 289
 Full-Width at Half Maximum (FWHM), 573
 Fully convolutional network (FCN), 1300
 Functional MRI (fMRI), 237, 1473
 Fundamental condition, 719, 733
 Fuzzy c-means (FCM) algorithm, 1205
- G**
- Gâteaux derivative, 449, 472
 Gâteaux differentiability, 928
 Gabor wavelets, 1527
 Game formulation, 686
 Game-theoretic approach, 681

- Game theory, 678–681
 deep learning, 696–702
 image registration, 683–696
 image restoration and segmentation, 681–682
- Gamma distribution, 315, 317, 321, 323, 324, 328, 1394, 1395
- Gamma function, 315, 342
- Gamma noise, 315, 316, 337, 1393–1396
- Gap
 duality, 732
 generic, 718
 Lagrangian, 732
 partial, 732
- Gauge frame, 1559
- Gauss-Bonnet condition, 1759, 1760
- Gauss-Bonnet theorem, 1757
- Gaussian curvature, 955, 1744, 1746, 1753, 1768
- Gaussian distribution, 314, 315, 324, 336, 978, 1237, 1271, 1661
- Gaussian kernel, 1448
 methods, 1403
- Gaussian measure, 1255
- Gaussian mixture model (GMM), 1005, 1041
- Gaussian noise, 176, 183, 184, 194, 314, 316, 325, 339, 387, 510, 541, 543–545, 683, 976, 983, 989, 993, 995, 1390
 noise vector, 1043
- Gauss-Newton algorithm (GN), 160, 161
- Gauss-Newton method, 1458
- Gauss-Seidel relaxation, 440
- Generalised Cauchy inequality, 734
- Generalised iterative soft thresholding (GIST), 743
- Generalised Lasso path, 955
- Generalized GN (GGN), 162
- Generalized Hölder's inequality, 1471
- Generalized Kullback-Leibler divergence, 320
- Generalized Lax-Oleinik formula, 215
- Generalized Tikhonov regularization, 1187
- Generalized Weierstrass theorem, 510
- General reflectance function (GRF), 1027
- General SO-model, 318
- Generating function, 712
 standard, 712
- Generative adversarial network-based losses, 826–827
- Generative adversarial networks (GANs), 696, 702, 777, 780–782, 855, 971, 973–975, 1664–1665
 AE-OT model, 1701–1703
 architecture, 697
 competition vs. collaboration, 1694–1696
 framework, 1694
 generative vs. discriminative algorithms, 696
 generator and discriminator, 700
 for image generation, 698
 for image segmentation, 699–702
 JS-GAN, 975
 memorization vs. learning, 1696
 mode collapsing, 1697–1700
 model loss, 700–702
 PD-GAN, 783–784
 PGGAN, 995–997
 PiiGAN, 782
 theory and numerics, 697
 training, 702
 WGAN, 976
 WGANgp, 976
- Generative Adversarial Networks for Pluralistic Image Inpainting (PiiGAN), 782
- Generative latent-based models, 779
- Generative methods, 778, 780, 782, 796
- Generator, 826
- Generic regularization, 1006–1008
- Genus one closed surfaces, 1777–1778
- Genus-0 closed triangle meshes
 conformal parameterization, 1495–1498
 quasi-conformal parameterization, 1498–1500
- Genus-0 point clouds, 1511
- Geodesic distance, 1447
- Geodesic Active Contour Model (GAC), 430
- Geodesic active contours (GAC), 1010, 1014–1015
- Geodesic boundary value problem, on parametrized curves, 1371–1372
- Geodesic contour model, 471
- Geodesic distance, 1357, 1361, 1363–1365, 1367, 1371, 1374, 1377
- Geometrical constraints, 1447–1449
 characterisation, 1442–1443
 convex models, geodesic distances, 1446
 convex segmentation models, 1446
 dual level model, 1444
 moments constraint for segmentation, 1445–1446
 moving band model, 1443
 simple variational model, 1443
- Geometric group action, 1722
- Geometric methods, 1465
- Geophysics, 239
- Gestalt theory of shape perception, 774

- Gibbs sampling, 792
Giotto Class digital system, 574
 Global segmentation model, 479
 Global smoother, 440
 GMRES, 668, 671
 Γ -convergence, 1821
 Gradient based architectures, 755–756
 Gradient descent (GD), 358
 Gradient flow, 1637
 Gradient function, 13
 Gradient mapping, 1668
 Graduated non-convexity (GNC) strategy, 10
 Gramian matrix, 1214
 Granite principal field synthesis, 1036
 Graphics processing unit (GPU), 380
 Grayscale, 586
 Green's function G , 280
 Grenander's metric pattern theory, 1933, 1950
 Gronwall lemma, 1641
 Ground truth (GT), 801
 Growth models, 1943–1944, 1946
 - constraints, deformation modules, 1949, 1950
 - external action, 1947–1948
 - growth as an internal force, 1945–1946
 - Riemannian viewpoint, 1944–1945
 Growth tensor, 1943, 1944, 1946, 1947
- H**
- H^1 -Laplacian model, 959
 Hölder inequality, 392
 Hölder's inequality, 641
 Hölder space, 1673
 Hörmander's theorem, 1543
 Haar measure, 1101
 Hadamard form, 1587, 1592
 Hadamard product, 920
 Hadamard space, 744
 Hadamard structure theorem, 1591, 1596
 Haker-Tannenbaum-Angent method, 1665
 Hamiltonian dynamics, 1330
 Hamiltonian flow, 1544–1549, 1573–1576
 Hamiltonian systems and landmark dynamics, 1330–1332
 Hamilton-Jacobi equations, 1833
 Hamilton-Jacobi partial differential equations (HJ PDEs), 212
 - decomposition problems, application to, 220–223
 - min-plus algebra for HJ PDEs and non-convex regularizations, 216–220
 - multi-time HJ PDEs and image decomposition models, 214–215
 - single time HJ PDEs and image denoising models, 213–214
 - viscous Hamilton-Jacobi PDEs and Bayesian estimation, 224–229
 Hardamard structure theorem, 1617
 Harmonic analysis, 952
 Harmonic maps, 1767–1770
 Hausdorff distance, 1692, 1842, 1861, 1867
 Hausdorff-Lipschitz continuity, 1830, 1842, 1843
 Hausdorff measure, 1003, 1386
 Heisenberg group, 1530
 Helgason-Ludwig consistency conditions, 1193
 Hessian matrix, 160, 161, 661, 1679, 1680, 1862
 Hessian operator, 44, 45, 160
 Hessian Shatten 2-norm, 43, 54
 Heterogeneous ADMM (hADMM), 645
 Hierarchical vector quantized variational autoencoder, 788, 789
 High dimensional problems, 168
 High Efficiency Video Coding (HEVC), 177
 High genus closed surface, 1778–1779
 High order models, augmented Lagrangian methods
 - Eulers elastic based model, 536–539
 - mean curvature-based model, 539–541
 - second order total variation model, 528–530
 - total generalized variation model, 531–536
 Hilbert space, 656, 711, 712, 714, 736, 741, 1070–1072, 1074, 1079, 1096, 1186, 1603, 1711, 1719, 1722, 1723, 1798, 1936, 1945
 - norms, 1067
 Hilbert transform, 1191, 1192
 Hodge decomposition, 1770–1772
 - theorem, 1671
 Holomorphic function, 1493
 Holomorphic one-form group, 1771, 1772
 Holomorphic quadratic differential, 1749–1750
 Homeomorphism group, 1773
 Homogeneous boundary condition, 632, 633
 Homogeneous space, 1366–1367, 1528, 1530, 1562
 Homomorphic filtering, 1206
 Hopf differential, 1768
 Hounsfield units (HU), 366
 H^1 -semi norm, 240
 - directional H^1 -semi norm, 248
 - weighted H^1 -semi norm, 246
 Huber contamination noise model, 976–977

- Huber function, 12, 13, 17, 19, 25
 Huber loss, 825
 Huber regularization, 919, 925, 926
 Huber-type functional, 949
 Hybridizable discontinuous Galerkin method, 282
 Hybrid level set method, 1388
 Hybrid methods, 1465
 Hybrid models
 description, 1936
 elastic metrics, 1937–1942
 Hyperbolic geometry, 1762
 Hyperbolic interpretation, 1762
 Hyperbolic surface, 1746
 Hyperbolic tangent function, 1299
 Hyperbolic Yamabe flow, 1763
 Hyper-elastic material, 1937
 Hyperspectral images, 1398–1399
- I**
- Identity matrix, 952
 I-divergence model, 320
 Ill-posedness, 553, 1067–1068
 Image colorization, 586, 822, 848
 with channels coupling, 605–607
 detailed architecture, 831–833
 distribution-based losses, 824, 828
 error-based losses, 824–826
 evaluation metrics, 835
 generalization to archive images, 842–844
 generative adversarial network-based losses, 826–827
 proposed colorization framework, 831–836
 qualitative evaluation, 838–841
 quantitative evaluation, 836–838
 quantitative evaluation metrics, colorization methods, 833–836
 Image deblurring, 1256–1265
 Image decomposition, 947
 adaptive balancing, 1167–1175
 applications and challenges, 1157–1160, 1178
 definition, 1156
 diffusion methods, 1160
 Fourier and wavelet methods, 1160–1162
 machine learning, 1166–1167
 properties, 1175–1177
 variational problems, 1162–1166
 Image denoising, 119, 124, 681, 944, 1265, 1688
 student-t regularised, 119, 129
 Image inpainting, 774, 779, 915, 1122–1124
 learning-based methods, 777–778
 model-based inpainting, 775–777
 ImageNet classifier, 762
 Image processing, 1832, 1840, 1844, 1859, 1862, 1873
 Image reconstruction, 912
 Image registration and fusion, quasi-conformal theory, 1422–1423
 Image registration, game theory, 683–685
 bias correction, 688–696
 game approach, 686–687
 game model, 689–691
 iterative algorithm, 691
 MRI images, 692–694
 non-game approach, 686, 689
 perfusion CT registration, 693, 695
 simple registration model, 685–687
 Image registration method, 1465
 Image restoration, 8, 944
 denoising modules, 183–186
 Image segmentation, 428, 429, 1204, 1438, 1888
 geometrical conditions, 469, 470
 quasi-conformal theory, 1419–1421
 Image transformers, 790–791
 bidirectional and autoregressive transformers, diverse image inpainting with, 792–793
 high-fidelity pluralistic image completion with transformers, 791–792
 Imperfect data, 1185
 Importance weighted autoencoder (IWAE), 1663
 Improved signal-to-noise ratio (ISNR), 46, 49, 54
 Improvement of signal to noise ratio (ISNR), 541, 543
 Impulsive noise, 519
 Inception score (IS), 794
 Incident field u^{inc} , 278
 Incomplete data, 1185
 INDIE approach, 1089
 inertia, 733
 corrected, 743
 partial, 743
 Inexact accelerated block coordinate descent (iABCD), 631
 Inexact block symmetric Gauss-Seidel iteration, 653–656
 Inexact heterogeneous ADMM (ihADMM) algorithm, 630, 642–645
 convergence results, 645–652
 Inexact majorized accelerated block coordinate descent (imABCD), 631, 652–662
 Inf-convolution TV (ICTV), 327
 Infimal convolution, 12, 22

- Infimal-convolution total variation (ICTV), 913
- Infinite-dimensional case, bilevel problem, 924
 dualization, 929
 existence and properties, 926–927
 nonlocal problems, 930–932
 stationarity conditions, 927–929
- Infinitesimal strain tensor, 1938
- Initialization, 663
- Inpainting, 348, 353–354, 774–776, 779, 780, 782, 784, 785, 787, 788, 790, 793, 795, 798–800, 802, 803, 805–809, 915, 1877–1882
 diversity, 802–803
- Instability, 1067
- Intensity based metric for Gaussian measurements, 145
- Intensity based metric for Poisson measurements, 145
- Intensity inhomogeneity, 1204–1207, 1226, 1227, 1403–1405
- Interactive methods, 1440
- Interpolation, 439
- Inverse Cayley transform, 1501
- Inverse Fourier transform, 512, 1688
- Inverse mapping, 1668
- Inverse matrix, 1213
- Inverse NDFT, 293
- Inverse problems, 107, 236, 240, 709–710, 717, 752, 774, 778, 809, 1066, 1185
 convergence analysis, 1078–1079
 data-driven reconstruction, 1068–1069
 deep learning, 1074–1075
 extensions, 1079–1080
 ill-posedness, 1067–1068
 linear, 709
 NETT approach, 1080–1090
 nonlinear, 131, 710
 null-space networks, 1076–1078
 regularization methods, 1072–1074
 regularizing networks, 1075–1080
 right inverses, 1070–1072
 Inverse problems, learned regularizers deep regularizers, 1138–1149
 shallow learned regularizers, 1136–1137
- Inverse scale space (ISS) flow, 107
- Isomap, 992
- Isometric transformation group, 1773
- Isometries
 normalization, 1372
- Isothermal coordinates, 1743, 1753
- Isotropic models, 245–247
- Isotropic total variation, 913
- Isotropic TV-L2 model, 43
- ISTA-Net, 888–891, 902–905
- Iteration complexity analysis, 646
- Iteration index, 179
- Iterative algorithm, 691, 1035
- Iterative hard thresholding (IHT) algorithm, 885
- Iterative horizontalization method, 1375–1377
- Iteratively reweighted least-squares (IRLS), 1237
- Iterative principal field synthesis, 1033–1037
- Iterative reconstruction methods, 1186–1187
- Iterative shrinkage/soft thresholding algorithms (ISTA), 627
- Iterative shrinkage-threshold algorithm (ISTA), 883, 885
- Iterative soft thresholding, 742
 generalised, 743
- Itoh–Abe method, 116
- J**
- Jaccard similarity (JS) index, 1226
- Jacobian equation, 1666, 1668
- Jacobian matrix, 1759
- Jacobi identity, 1578
- Jensen-Shannon divergence, 974, 978
- Jensen’s inequality, 731, 732
- Joint probability distribution, 1048
- Joint reconstruction, 241, 265
- Joint segmentation and registration models
 existing methods, 1466–1468
 motivations, 1463–1466
 nonlocal characterisation of weighted total variation, 1468–1473
- Joint total variation (JTV), 246, 255
- K**
- Kaczmarz method, 120, 1278
 sparse, 111–113, 128
- Kantorovich potential, 1695
- Kantorovich problem of optimal transport, 1804
- Kantorovich-Rubinstein distance, 1805
- Kantorovich-Rubenstein duality, 781
- Kantorovich’s approach, 1667–1668
- Karush-Kuhn-Tucker (KKT) conditions, 636, 649
- Karush-Kuhn-Tucker multipliers, 931
- Karush-Kuhn-Tucker theory, 930
- KdV equation, 1644
- Kendall’s model, 1930
- Kendall’s space, 1930
- Kernels, 1800, 1802
- K-means clustering algorithm, 1204
- K-means method, 1389

- K -quasiconformal map, 1749
 Kramers-Kronig relation (KKR), 168
 Kronecker delta function, 1037, 1039
 Krylov-based methods, 631, 645
 k -space coverage, 286
 Kulback-Leibler (KL) divergence, 781, 855, 974, 1081, 1083, 1302, 1793, 1796–1797
 fidelity, 510, 520
 Kullback–Leibler (KL) loss, 828
- L**
- LabRGB strategy, 869
 Lagrange multiplier, 409, 511, 533, 540, 628, 637, 921, 923, 930, 1219
 Lagrangian, 714
 dynamics, 1330
 formalism, 928
 function, 629, 643
 noise, 1328, 1333, 1335
 rate-distortion optimization, 203
 Lambert-Beer's law, 561
 Lambertian surface reflectance, 1054
 Lamé coefficients, 1946
 Landmark-matching Teichmüller map, 1493
 Landweber algorithm, 760, 1186
 Landweber method, 742, 1076
 Langevin dynamics, 1327
 Laplace equation, 1425, 1512, 1515
 Laplace operator, 381
 Laplace-Beltrami eigenvectors, 1932
 Laplacian approximation, 1487
 Laplacian distribution, 978
 Laplacian operator, 1306, 1513, 1515, 1769
 Large deformation diffeomorphic metric mapping (LDDMM), 1292–1294, 1328, 1725, 1728, 1730–1732
 Lary–Lions regularizations, 1839
 Lasry-Lions double envelopes, 1833
 Lax-Oleinik formula, 213–216, 219
 Layered neural network, 1075
 Learnable descent algorithm (LDA), 898–902
 Learned algorithm, for specified optimization problem, 882–885
 Learned approximate message passing (LAMP), 885
 Learned ISTA (LISTA), 884, 885
 Learned iterative reconstruction, 753, 755
 gradient based architectures, 755–756
 initialization, 763
 learned operator, architectures for, 763
 learned step length, 765
 parameter sharing, 764
 preconditioning, 765
 primal-dual networks, 758
 proximal based architectures, 757–758
 scalable training, 765
 training procedure, 760–763
 Learned perceptual image patch similarity (LPIPS), 794, 795, 802, 834, 836, 837, 839–841, 844, 865
 Learned regularization functionals, 1080–1082
 Learned regularizers, for inverse problems
 deep regularizers, 1138–1149
 shallow learned regularizers, 1136–1137
 Learned synthesis regularization, 1090
 Learning-based methods, 314, 777–778
 Learnt post processing (LPP), 570
 Least-squares approach, 1238
 Least squares semidefinite programming (LSSDP), 631
 Lebesgue decomposition, 926, 1794, 1796
 Lebesgue measurable function, 1490
 Lebesgue measure, 1254, 1669, 1670, 1676, 1681
 Left-invariant vector fields, 1576
 Left ventricle (LV), 1009
 Legendre-Fenchel transform, 1838, 1863–1867
 Legendre transform, 1669
 Lemma
 Brezis–Crandall–Pazy, 730
 Opial, 730
 Levenberg-Marquardt method (LM), 160, 161
 L^2 fidelity term, 6
 l'Hôpital's rule, 1814
 L -hypersurface, 955, 959, 960
 Lie groups, 1364–1365
 Lie-Cartan connection, 1538–1541, 1577–1578
 (sub)-Riemannian geometry, 1543–1544
 in left-invariant coordinates, 1541–1542
 Limited angle computed tomography,
 1125–1127, 1192–1193
 data domain, learning in, 1193
 image domain, learning in, 1194
 knowledge of operator, 1194
 learned backprojection, 1195
 Limited-angle tomography, 556
 Line source, 282, 294
 Linear acquisition model, 6
 Linear Beltrami solver (LBS), 1426, 1491–1492
 Linear combinations, 1492
 Linear equation, 512
 Linear forward operator, 709
 Linear Independence Constraint Qualification Condition (LICQ), 921
 Linear isometries, 1361
 Linear map, 1543

- Linear operator, 1945
 Linear operators, 44, 144
 Linear-quadratic elliptic PDE-constrained
 optimal control problem, 624
 Linear regularization, 1078
 Linear right inverse, 1070–1072
 Linear second-order differential operator, 625
 Linear system, 4, 5, 644
 Lipschitz approximation, 1664
 Lipschitz boundary, 386, 1632
 Lipschitz condition, 976
 Lipschitz constant, 664, 1078, 1804, 1839,
 1859
L-Lipschitz continuity, 1811
 Lipschitz continuous functions, 1076, 1470
 Lipschitz continuous gradient, 656
 Lipschitz function, 432, 781, 1832
 Lipschitz smooth function, 62
 Local binary patterns (LBP), 1005
 Local Fourier analysis (LFA), 443, 455
 Local mesh method, 1487
 Local minimization problem, on the coarsest
 level, 491
 Log-likelihood maximization, 829, 831
 Long short term memory (LSTM), 1302
 Loss function, 66
 Low dose computed tomography, 1197–1198
 Low-Dose Parallel Beam (LoDoPaB)-CT,
 1197
 Low-order polynomial, 163
 LRShape, 1308–1309
 L^2 -TV model, 387, 388, 421
 Luminance channel, 848
 Luminance-chrominance space, 849
 Lumped mass matrix, 642
- M**
- Machine learning, 910, 1157, 1187, 1430
 with region-based active contour model,
 492
 Magnetic resonance imaging (MRI), 236,
 237, 242, 709, 1066, 1204, 1205,
 1207, 1208, 1215, 1224, 1225,
 1232
 diffusion, 709
 velocity-encoded, 709
 Mangasarian-Fromowitz Constraint
 Qualification Condition (MFCQ),
 921
 Manifold
 distribution principle, 1686–1689
 Hadamard, 743
 Riemannian, 743
 Manifold learning, 1688
 auto-encoder, 1692–1694
 ReLU DNN, 1690–1691
 Marginal density, 1045
 Marginal probability distribution, 1048
 Markov random field (MRF), 854, 910, 1027,
 1030, 1031, 1893
 Masked language model (MLM), 792
 Mass conservation law, 1671
 Mass matrices, 642
 Mass preserving group action, 1723
 Material-appearance editing, 1053
 Mathematical morphology, 1834
 Matrix analysis, 1214
 Ma-Trudinger-Wang's theorem, 1673
 Maurer-Cartan form, 1537
 Max-flow models, 1917
 Max Pooling layer, 87
 Max pooling strategy, 68
 Maximal magnification factor, 1489
 Maximum a posteriori (MAP) estimators, 184,
 211
 Maximum a-posteriori likelihood estimator,
 1134
 Maximum a posteriori probability (MAP), 314
 AA-model, 319
 I-divergence model, 320
 SO-model, 319
 Maximum a posteriori (MAP) solution, 1031
 Maximum mean discrepancies (MMDs), 1792,
 1800
 Maximum pseudo-likelihood equation, 1039
 McCann's displacement, 1669–1670
 Mean absolute error (MAE), 825
 Mean curvature-based model, 539–541
 Mean curvature flow (MCF), 1565, 1567,
 1569
 Mean square error (MSE), 824–825, 833, 866,
 961, 982, 984, 986, 987, 990, 993,
 995, 996, 1304
 criteria, 919
 Measurable Riemann mapping theorem,
 1490–1491
 Measure-preserving map, 1666
 Mechanical blur, 915
 Medical image segmentation, 1002
 boundary information, 1004
 Chan-Vese model inspired loss function,
 1013, 1014
 data term, 1004–1007
 generic regularization, 1006–1008
 geodesic active contour inspired loss, 1015
 geodesic active contour-inspired loss, 1014

- learning hyper-parameters, in end-to-end framework, 1016
- learning hyper-parameters, in two-stage framework, 1015–1016
- Mumford-Shah model inspired loss function, 1013
- regional information, 1005–1007
- and registration, quasi-conformal theory, 1418–1424
- regularization term, 1006–1010
- targeted regularization terms, object properties, 1008–1010
- variational models inspired network modules, 1011–1013
- Medical imaging, 1785–1787
- Medium-induced blur, 915
- Mercer's theorem, 1797
- Mesh parameterization, 1486
 - quasi-conformal geometry, 1494–1511
- Metal artifact, 348, 353, 355, 357, 360, 373–375
 - reduction of, 1196–1197
- Metal artifact reduction (MAR), 348, 357
 - algorithm, 349, 355, 360
 - performance, 365
- Metal extraction, 355
- Metamorphosis, 1724–1725, 1727
- Metric tensor, 1541
- Meyers model, 947
- Microcalcification, 574, 575
- Microlocal analysis, 1104
- Min-plus algebra, 216–220
- Minimax concave penalty function, 12, 15, 19, 47
- Minimization problem, 512, 518, 527, 530–531, 533–535
- Minimum distance function (MDF), 955
- Mirror descent, 98, 102, 103, 742
 - Stochastic mirror descent (SMD) method, 111
- Mirror prox, 742
- Misfit functional, 289
- Misregistration, 262
- Mixed Nash equilibrium (MNE), 697
- Mixture
 - distribution, 1040
 - Gaussian, 1047
- Mixture density network (MDN), 856
- MNIST dataset, 698, 699
- Möbius transformation, 1748, 1779, 1782
- Model based deep learning (MODL) approach, 1089
- Model-based inpainting, 775–777
- Model-based iterative algorithms, 553
- Modern image segmentation algorithms, 1205
- Modified energy, 1224
- Modified Inception Score (MIS), 794
- Modified TV, 326
- Modular ADMM-based strategies, *see* Alternating direction method of multipliers (ADMM)
- Moment constraints, 1445–1446
- Momentum, 1190
- Monge-Ampère equation, 1668, 1670
- Monge-Brenier theory, 1662
- Monge-Kantorovich theory, 1662
- Monge's problem, 1665–1666
- Monotone
 - operator, 714
 - strongly, 725
- Monte-Carlo method, 1684, 1697, 1702
- Monte Carlo/quasi Monte Carlo methods, 1587
- Montpellier, F., 586
- Moore-Penrose inverse, 1072, 1074
- Mordukhovich (M-) stationary, 923
- Moreau envelope, 12, 22, 23, 33, 34, 1833, 1834, 1836, 1839, 1862, 1865–1867
 - gradient, 13
- Moreau-Yosida function, 929
- Morozov's discrepancy principle, 1241, 1244, 1252
- Morpho-elasticity, 1943
- Morphological component analysis (MCA), 952, 953
- Morphometric mapping, 1424
- Motion model, 1714, 1732–1733
 - general variational formulation, 1714
 - parametrised, 1714
 - PDE, 1718–1722
- Moving least squares (MLS), 1487, 1511
- Multi-block convex optimization problems, 630
- Multichannel TV restoration, 524–527
- Multi-class cross-entropy loss, 701
- Multi-contrast MRI, 237, 238
- Multi-dimensional data modeling, 1025
- Multigrid algorithm, 442
- Multigrid method (MG), 167, 436, 446
- Multigrid with modified smoother (MG1m), 459
- Multigrid with typical local smoother (MG1), 459
- Multi-modal problems, 1474
- Multi-orientation image processing, geometric flows
 - $\nu = 1$ and Hamiltonian flows for Riemannian geodesic problem on G , 1544–1549

- Multi-orientation image processing, geometric flows (*cont.*)
 - homogeneous space \mathbb{M}_d of positions and orientations, 1549–1550
 - image analysis applications for $G = SE(d)$, 1559–1572
 - left-invariant coordinates, Lie-Cartan connection (dual) in, 1541–1542
 - Lie groups $G = \mathbb{R}^d \rtimes T$ and left-invariant processing and left-invariant connection on $T(G)$, 1530–1533
 - metric models on \mathbb{M}_d , shortest curves and spheres, 1550–1554
 - (sub)-Riemannian geometry, (partial) Lie-Cartan connections for, 1543–1544
 - shortest curve application, 1560–1563
 - straight curve application, 1563–1572
 - straight curve fits, 1554–1559
 - Multi-parameter approaches, in image processing, 958–963
 - balancing principle and balanced discrepancy principle, 954–955
 - dictionary based approaches, 950–953
 - generalised Lasso path, 955
 - L -hypersurface, 955
 - multi-parameter discrepancy principle, 953
 - numerical solution, 957–958
 - parameter learning, 956–957
 - parameter selection, 953–957
 - PDE based approaches, 946–950
 - Multi-parameter discrepancy principle, 953
 - Multiple image inpainting, 778
 - autoregressive models, 788–790
 - datasets, 796
 - GANs, 780–784
 - image transformers, 790–793
 - inpainting masks, 797
 - qualitative performance, 803–808
 - quantitative performance, 797–803
 - single image evaluation metrics to diversity evaluation, 793–795
 - variational autoencoders and conditional variational autoencoders, 784–788
 - Multiplicative intrinsic component
 - optimization (MICO), 1207, 1224, 1225, 1227, 1229–1232
 - bias field rectification capabilities, 1232
 - decomposition of MR images, 1207–1208
 - energy formulation, 1209–1211
 - execution of, 1215
 - mathematical description, 1208–1209
 - modified MICO formulation with weighting coefficients for different tissues, 1224
 - numerical stability, matrix analysis, 1213–1215
 - optimization of energy function and algorithm, 1211–1213
 - proposed TV based MICO model and solver, 1217–1222
 - spatial regularization, 1217
 - spatiotemporal regularization for 4D segmentation, 1222–1224
 - Multiplicative noise removal model
 - DNN method, 338–343
 - MAP based models, 319–320
 - m th root transformation model, 323
 - multi-tasks, 334–335
 - non-convex regularization, 330–334
 - root and inverse transformation based models, 320–325
 - sparse regularization, 327–330
 - statistical property based models, 318–319
 - TV regularization, 325–327
 - Multiplicative operator splitting (MOS) method, 435
 - Multiplicative Schwarz method, 383
 - Multiply-connected open triangle meshes
 - conformal parameterization, 1507–1508
 - quasi-conformal parameterization, 1508–1510
 - Multiscale medial axis map, 1831, 1856–1859
 - Multi-tasks
 - fractional transformation, 335
 - nonlocal methods, 335–338
 - root transformation, 334–335
 - Multi-time HJ PDEs, 214–215
 - Multivariate functions, 1098
 - Multivariate Gaussian densities, 1048
 - Mumford and Shah energy, 432
 - Mumford-Shah functional, 682
 - Mumford–Shah model, 431, 432, 948, 957, 961, 1003, 1013, 1386, 1468, 1471, 1889
 - Mumford–Shah regularisation, 945
 - m -V model, 320
- N**
- Nakagami distribution, 321
 - Nash equilibrium (NE), 679–682, 686, 690, 691, 697, 780, 1694
 - Nash equilibrium problem (NEP), 679
 - Nash game framework, 680
 - Nash strategies, 680

- National Elevation Dataset, 1871
- Natural holomorphic coordinates, 1750
- Natural language processing (NLP), 790
- Negative log-likelihood, 789
- Nesterov-acceleration, 108
- Nesterov updates, 1190
- Network cascades, 1088–1089
- Network modules, 1011–1013
- Network Tikhonov (NETT), 1135, 1138, 1141, 1149, 1151
 - photoacoustic tomography reconstruction, 1142
 - training setup, 1141
- Network Tikhonov (NETT) approach
 - convergence analysis, 1082–1088
- INDIE approach, 1089
- learned regularization functionals, 1080–1082
- learned synthesis regularization, 1090
- MODL, 1089
- network cascades, 1088–1089
- variational networks, 1088
- Neumann boundary condition, 391, 438, 447–449, 476, 479, 946, 961, 1618, 1645
- Neumann networks, 759
- Neural network, 64, 1074
 - as operators, 1656
- Neural network optimization
 - deep neural networks, 933
 - deep unrolling within optimization, 933
- Neural Network Tikhonov (NETT) approach, 933
- Newton method, 160
- Newton–Raphson algorithm, 164
- Nodal quadrature formula, 626
- Noise Gaussian distribution, 6
- Noise inference, from evolution of moments, 1341–1342
- Noise spectral correlation, 1047
- Non-convex regularization
 - fractional order TV, 330–331
 - non-convex TGV, 332
 - nonconvex sparse regularizer model, 331–333
- Nonconvex sparse regularizer model, 331–333
- Noncooperative games, 679, 680
- Non-differentiable function, 33
- Non-dissipative stochastic shape models, 1333–1336
- Non-ergodic convergence, 737
- Non-existence, 1067
- Non-game approach, 686, 689
- Nonlinear conjugate gradient (NLCG)
 - algorithm, 163–164
- Nonlinear eigenfunctions, 1635
- Nonlinear eigenvalue problem, 1634
- Nonlinear flows, 1636
- Nonlinear least square (NLS) problems, 161
- Nonlinear reconstruction, 1195
- Nonlinear vector-valued functions, 44
- Nonlocal methods
 - direct method, 337–338
 - indirect method, 336–337
- Nonlocal problem (NLP), 1472
- Non-local TV, 326
- Non-manifold Laplacian method, 1487
- Non-negative, 712
- Non-overlapping domain decomposition, 381, 397–398, 419–420
 - Dirichlet-Neumann method, 382
 - FETI approach, 404–406
 - finite difference, 398–400
 - finite element approach, FISTA, 402–404
 - parallelism, 383
 - subdomain problems, 400–402
 - variational formulation, 383
- Non-parametric Markov random field, 1032
 - with fast iterative synthesis, 1035–1037
 - with iterative synthesis, 1033–1035
- Non-parametric methods, 1025
- Non-potential game, 691
- Non-quadratic fidelity, 519–523
- Non-singularity of matrix, 1214
- Non-smooth optimization problems, 10
- Nonsmooth second-order conditions, 724–727
- Nonuniform discrete Fourier transform (NDFT), 292
- Non-uniqueness, 1067
- Normalized correlation coefficient (NCC), 692
- Normalized gradient differences (NGD), 684, 687
- Normalized Lebesgue measure, 1807
- Normalized local cross correlation (NLCC), 1304
- Normalized mean square error (NMSE), 885
- Normalized metal artifact reduction (NMAR), 349, 354, 355, 367, 373, 374
 - algorithm, 354–355
 - inpainting of metal traces, in normalized sinogram, 353–354
 - performance, 368
 - vs. SMAR, 365–368
- Nossek-Gilboa (NG), 1636–1637
- NTIRE challenge, 852
- Null-space, 1636
 - networks, 1068, 1076–1078

O

- Observable subspace, 1240
 - Omni-tomography, 240
 - One-homogeneous functionals, 1632
 - One level selective segmentation model, 477
 - Operational rate-distortion optimizations, 203–205
 - Optical blur, 915
 - Optimal control problems, 626
 - Optimal discriminator, 827
 - Optimal flow frameworks, 1473
 - Optimal transport (OT), 1661, 1792, 1804
 - Benamou-Brenier dynamic fluid, 1670–1671
 - Brenier’s approach, 1668–1669
 - computational algorithm, 1676–1685
 - damping Newton’s method, 1680–1684
 - encoder-decoder architecture, 1663
 - generative adversarial networks, 1664, 1694–1703
 - hybrid models, 1664
 - Kantorovich’s approach, 1667–1668
 - manifold distribution principle, 1686–1689
 - manifold learning, 1688–1694
 - McCann’s displacement, 1669–1670
 - Monge’s problem, 1665–1666
 - Monte-Carlo method, 1684
 - numerical method, 1665
 - optimal transportation map, 1662–1663, 1673–1676
 - Otto’s calculus, 1671–1673
 - regularized optimal transport, 1805–1815
 - semi-discrete optimal transport map, 1676–1680
 - Optimal transportation map, 1662–1663, 1673, 1783
 - convex target domain, 1673
 - male face, 1682
 - non-convex target domain, 1674–1676
 - semi-discrete optimal transport map, 1676–1680
 - Optimisation, 98
 - Bregman iterations, 106
 - large-scale stochastic, 111
 - mathematical, 98
 - stochastic, 111
 - Ordinary differential equation (ODE), 1293, 1725, 1727
 - Orientation score, 1531–1533
 - Orlicz space, 1808
 - Orthonormal basis, 1797
 - Orthonormal vectors fields, 1939
 - Otto’s calculus, 1661, 1671–1673
 - Outer semicontinuous, 729, 730
 - Over-fitting, 938
 - Overlapping domain decomposition, 392–397
 - additive Schwarz method, 384
 - derivation of, 413
 - multiplicative Schwarz method, 383
 - subdomain problems, 397
 - variational formulation, 384–385
 - Over-relaxation, 743
- P**
- Pansharping, 238, 239
 - Parabolic scaling matrix, 1101
 - Parallelizable global conformal parameterization (PGCP), 1497, 1503, 1504
 - Parallel level sets, 243
 - Parallel momentum, 1534
 - Parallel velocity, 1534
 - Parameter estimation, 1048–1049
 - Parameter regularisation, 954
 - Parametrised motion models, 1714
 - Parasitic scattering, 168
 - Parseval frame, 1107, 1115
 - Partial differential equation (PDE), 709, 776, 1450, 1456, 1460, 1461, 1466, 1709, 1710, 1727, 1733, 1947
 - implementation and reconstruction, 1720–1722
 - joint motion estimation and reconstruction, 1719–1720
 - physical motion constraints, 1718–1722
 - Partial Lie-Cartan connection, 1544
 - Particle swarm optimization (PSO), 363
 - Patch-based methods, 602–605, 610
 - Patch-based sampling methods, 1025
 - PDE based approaches, 946–950
 - PDE-constrained shape optimization problems, 1590
 - Peak signal-to-noise ratio (PSNR), 294, 365, 801, 833, 866, 919, 938, 983–987, 990, 993, 995, 997, 1875–1878
 - Perceptual diversity loss, 783
 - Perceptual quality, 801–802
 - Perron-Frobenius theory, 1636
 - Perona function, 1919
 - PET-CT, 236, 237
 - PET-MR, 236, 237
 - PhantomNet, 1126
 - Phase retrieval (PR), 140
 - 4-Phase segmentation model, 453, 1915
 - Photoacoustic tomography (PAT), 757, 1141, 1184
 - Photonics media, 915
 - Physical imaging, 1056

- Piecewise constant level-set method (PCLSM), 1891
- Piecewise constant Mumford-Shah (PCMS) model, 1386–1388, 1392, 1393
- Piece-wise linear function (PLF), 893
- Piecewise linear (PL) metric, 1756, 1758–1762
- PixColor model, 856
- Pixel-driven approach, 563
- Plücker’s conoid, 1869
- Plücker surface, 1870
- Planar curves, 536
- Planar mapping, 1781
- Plane wave, 279
- Pluralistic image completion, 786
- Poincaré model, 1767
- Point cloud parameterization, 1487
 - conformal and quasi-conformal geometry, 1509–1515
- Point source, 281
- Point spread function (PSF), 142, 1257
- Point-wise approximations, 1072
- Pointwise error, 1830
- Poisson distribution, 141, 1276, 1393, 1395
- Poisson noise, 315, 318, 387, 510, 519, 543, 915, 1393–1395, 1397, 1711
- Poisson problem, 381, 382
- Polish space, 1794, 1795
- Polyak-Lojasiewicz condition, 79, 80
- Polyhedral surfaces, 1756
 - circle patterns, 1764–1767
 - vertex scaling, 1758–1764
- Positive-definite matrix, 1214
- Positron emission tomography (PET), 236, 242, 709
- Posterior probability, 1030
- Post-processing networks, 1068
- Potential game, 680, 690–691
- Potts Markov random field, 1037
- Potts model, 710, 1888
 - continuous max-flow formulation, 1900–1903
 - convex relaxation via convex envelope, 1913–1914
 - for integer-valued functions, 1893–1895
 - with overlapping binary functions representation, 1911
 - graph cuts for integer-labeled, 1895–1899
 - high dimensional space, 1918–1919
 - with simplex-constrained representation, 1920–1922
- Potts-Voronoi Markov random field, 1038–1040
- Preconditioned ADMM, 743
- Preconditioner, 714
- Prewhitening, 251–252
- PR772 particle dataset, 1283
- Primal–dual
 - fixed point method, 743
 - splitting
 - block-adapted, 716–718, 723–724, 727–728
 - Bregman-proximal, 715, 738
 - explicit, 742
 - inertial, 735
 - proximal, 710, 714, 715, 738
- Primal–dual active set (PDAS), 627
- Primal–dual Bregman-proximal splitting (PDBS), 715, 716, 724, 730, 735–738, 740
- Primal–dual explicit spitting (PDES), 742
- Primal–dual FB (PDFB), 36–38, 45
- Primal–dual fixed point method (PDFP), 743
- Primal–dual hybrid gradient (PDHG), 250, 251, 934
 - method, 122–126
- Primal dual hybrid gradient CS network (PDHG-CSNet), 898, 899
- Primal–dual network (PD-Net), 758, 896–898
- Primal–dual proximal splitting (PDPS), 713, 715, 729, 735, 737, 738, 743
 - algorithm formulation, 715–716
 - block-adapted, 716–718, 723–724, 727–728
 - modified, 738
 - optimality conditions and proximal points, 714
- Primal total variation, domain decomposition, 406–408
 - convergence properties, 408–410
 - (pre)dual, 412–420
 - subspace minimization, 410–412
- Principal geodesic analysis (PGA), 1309
- Principal single model Markov random field, 1032
- Principle component analysis (PCA), 1026
- Probabilistic Diverse GAN for Image Inpainting (PD-GAN), 783–784
- Probability density function (PDF), 45, 315, 316, 323, 324
- Probability distribution, 1047
- ProbDR, 1307–1308
- Probe drift, 167
- Problem
 - dual, 742
 - min-max, 708
 - primal, 708
 - saddle point, 708
- Prospective methods, 1205

- Proximal algorithms, 151–153
 Proximal alternating linearized minimization (PALM), 151
 Proximal alternating predictor corrector (PAPC), 743
 Proximal based architectures, 757–758
 Proximal heterogeneous block implicit-explicit (PHeBIE), 151–152
 Proximal mapping, 147, 716
 Proximal operator, 146, 250
 Proximal point method, 714, 742
 Proximal point network, 886–888
 Proximity hull, 1833, 1862
 Proximity operator, 12
 Prox-operators, 958
 Prox-simple, 715, 716, 741, 742
 Pseudo-likelihood approximation, 1039
 Ptychographic phase retrieval, 142
 Ptychography, 165
 p -Wasserstein distance, 1805
 Pythagoras theorem, 1072
- Q**
- Quadratic assignment problems (QAPs), 658
 Quadratic envelopes, 1835
 Quadratic function, 653, 660
 Quadratic optimization problem, 512, 534
 Quadratic regularisation, 946
 Quantitative analysis, 365
 Quantitative evaluation metrics, 833–836
 Quantitative results, 809
 Quasi-conformal maps, 1415–1417, 1457, 1488–1491, 1746, 1748–1749, 1781–1782
 Quasi-conformal parameterization, 1498–1500, 1505–1510
 Quasi-conformal (QC) Teichmüller theory, 1414, 1432, 1433
 conformal mappings, 1414–1415
 quasi-conformal mappings, 1415–1417
 Teichmüller mappings, 1417–1418
 Quasi-conformal theory, 1485
 Quasiconvex envelope, 1828
 Quasiconvex problem (QP), 1471
 Quasi-interpolation operator, 639
 Quicksilver, 1311
 Quotient space, 1755
- R**
- Rada-Chen selective segmentation, 485
 Rademacher's theorem, 1832
 Radial basis function (RBF), 1487
 Radial kernels, 1800, 1802
 Radiative absorption coefficient, 1196
 Radon-Nikodym derivative, 1794
 Radon norms, 925
 Radon–Riesz property, 1073
 Radon transform, 508, 512, 559, 1125, 1127, 1190–1192, 1194, 1196
 matrix, 4
 Random decrement technique (RDT), 1047
 Randomized Kaczmarz algorithm, 1279
 Random sampling, 1722
 Random spectral reflectance vector, 1028
 Raw accuracy, 835
 Ray-driven (ray-casting) approach, 563
 Rayleigh distribution, 314
 Rayleigh noise, 318
 Rayleigh quotient, 1641
 Reaction-diffusion-convection, 1948
 Reconstruction operator, 1252
 Rectified linear unit (ReLU), 1189, 1299
 Recurrent inference machines, 757
 Recurrent neural networks (RNNs), 1302
 Refractive index n , 277
 Regional information, 1005–1007
 Regularization, 912, 1948
 functionals, 1067
 matrix, 6
 parameter, 6, 567, 997
 Regularization by denoising (RED), 177, 1135
 Regularized optimal transport, 1805–1815
 Regularized PIE (rPIE), 150
 Regularized variational methods, 6
 ReLU deep neural network, 1690–1691
 Remote sensing, 238, 239
 Reparametrization, 1377
 diffeomorphism group and gradient based methods, 1374–1375
 dynamic programming approach, 1372–1374
 iterative horizontalization method, 1375–1377
 Reproducing kernel Hilbert space (RKHS), 479, 1798, 1934
 ResBCU-Net, 494
 Residual connections, 1190
 Residual network (ResNet), 886
 Restoration models, 915
 Restriction, 439
 Retrospective approaches, 1205
 RGB color space, 849
 Ricci flow, 1752–1754, 1765
 Ridgelet coefficients, 317
 Riemannian Brownian motion, 1327, 1333, 1334, 1343
 Riemannian distance, 1758, 1931
 Riemannian geometry, 1590, 1740

- Riemannian manifold, 1350, 1356, 1367–1371, 1537, 1543, 1545, 1589, 1933
- Riemannian metrics, 1672, 1688, 1693, 1743, 1746, 1750, 1752, 1758, 1767, 1933, 1934
 general framework, 1351–1359
 SRV framework, 1359–1371
 tensor field, 1543, 1544, 1562
- Riemannian multi-shape gradient, 1600, 1607
- Riemannian shape gradient, 1586, 1593
- Riemannian shape manifold, 1592
- Riemannian structure, 1326, 1330
- Riemannian submersion, 1356
- Riemann mapping theorem, 1417, 1681, 1746, 1748
- Riemann-Roch theorem, 1749
- Riemann surface, 1491, 1742, 1743, 1746, 1749–1751
- Riesz' representation theorem, 1794, 1799
- Rigid motion group, 1773
- Risk minimization, 1135
- RLO-model, 318
- RNA polymerase (RNAP), 981–983, 991
 denoising without contamination, 986, 988
 robustness under contamination, 989
- Robin type, 625
- Robust denoising method, 977
 Huber contamination noise model, 976–977
 robust recovery, β -GAN, 978–980
 stabilized robust denoising, joint autoencoder and β -GAN, 981
- Robust principal components analysis, 124–127
 rotation, 286
- Rudin-Osher-Fatemi (ROF) model, 1386
- Rytov approximation, 280, 304
- S**
- Saddle point, 714
 problem, 520, 530
- SARAH method, 76
- Scaled gradient projection (SGP) method, 573
- Scattered data, 1828, 1860, 1873
- Scattered field u^{sca} , 278
- Scattering potential f , 277
- Schwartz function, 352, 353
- Schwarz' inequality, 1799
- Scribble-based image colorization, 852–853
- Second order elastic metrics, 1379
- Second order exponential curve, 1557–1559
- Second-order growth conditions, for
 block-adapted methods, 727–728
- Second-order linear elliptic differential operator, 624
- Second-order total generalized variation
 Gaussian denoising, 927
- Second order total variation model, 528–530
- Segmentation, 468, 710
 4D segmentation, 1206, 1222–1224, 1232
- Segmentation-based techniques, 601–602
- Seismic tomography, 239
- Semiconvex envelope, 1832–1833
- Semidefinite linear operators, 657
- Semidefinite programming (SDP), 157
- Semi-discrete optimal transport map, 1676–1680
- Semi-elliptic, 712
- Semi implicit method, 433
- Semi-implicit scheme, 1639
- Semi-proximal alternating direction method of multipliers (sPADMM), 1404
- Semismooth Newton (SSN), 627
- Separable trapezoid footprints algorithm, 563
- sGS-imABCD algorithm, 659–662
- Shallow learned regularizers
 bilevel learning, 1136–1137
 dictionary learning, 1137
- Shannon-Boltzmann entropy, 1796
- Shape analysis, 1326
- Shape calculus, 1591, 1592
- Shape interrogation, 1836, 1840, 1853
- Shape optimization, 1586
- Shape priors, 1474
- Shape spaces, 1773–1779, 1930–1932
 definitions, 1588
 diffeomorphic action, 1933–1936
 growth models, 1943–1950
 hybrid models, 1936–1942
- ShearLab3D, 1116, 1118
- Shearlet systems, 1099
 α -molecules, 1113–1114
 continuous, 1100–1105
 deep learning, 1124–1129
 digital, 1116–1119
 discrete, 1105–1110
 higher dimensions, 1111–1112
 sparse regularization, 1120–1124
 universal, 1114–1115
- Shear matrices, 1111
- Shepp-Logan brain phantom, 553
- Shepp-Logan digital phantom, 560
- Shepp-Logan phantom, 556, 1142
- Shoepoint analysis, 1157
- Shortest curve application, 1560–1563
- Shrinkage operator, 1220
- Signal-to-noise ratio (SNR), 46, 169
- Signed distance function (SDF), 1890

- Simplex-constrained vector functions, 1905
 - dual formulation, 1907–1910
 - primal-dual formulation, 1906
- Simply-connected open triangle meshes
 - conformal parameterization, 1501–1505
 - quasi-conformal parameterization, 1505–1507
- SinGAN, 778
- Single exponential curve fit, 1559
- Single multi-spectral texture factors, 1046
- Single penalty synthesis, 951
- Single-photon emission computed tomography (SPECT), 237
- Single time HJ PDEs, 213–214
- Singular value decomposition (SVD), 1237, 1239, 1248, 1253, 1557–1559
- Sinkhorn algorithm, 1811, 1818, 1820
- Sinkhorn divergence, 1793, 1815–1818, 1823
- Sinogram, 349–354, 356, 359, 559
 - completion, 350
 - corrected, 353, 355, 359, 360
 - corrupted, 356
 - flat, 353
 - image, 553
 - inconsistent, 353
 - interpolated, 353
 - normalized, 353, 355
 - polychromatic, 365
 - seamless surgery, 360
 - surgery, 349, 355, 356, 364, 374
 - surgery region, 365
 - uncorrected, 353
- Smooth truncated AGM (ST-AGM), 146
- Smoothed total variation, domain decomposition
 - direct splitting approach, 390
 - Euler-Lagrange equation, 391
- Smother, 439
- Smoothing and thresholding (SaT)
 - segmentation methodology, 1389–1392, 1405
 - rate, 457
 - SLaT method, for color images, 1396–1398
 - three-stage method for images, with intensity inhomogeneity, 1403–1405
 - tight-frame based method for images, with vascular structures, 1400–1401
 - T-ROF method, 1392–1393
 - two-stage method, for hyperspectral images, 1398–1399
 - two-stage method, for Poisson/Gamma noise, 1393–1396
 - wavelet-based segmentation method, for spherical images, 1401–1403
- Smoothing, lifting and thresholding (SLaT) method, 1396–1398
- Sobolev gradient, 449
 - of curve length, 449
- Sobolev metric of order, 1355, 1359
- Sobolev norms, 1931
- Sobolev regularity, 1358
- Sobolev space, 682, 928, 1417, 1490
- Softmax function, 1013
- Solitons, 1644
- Sparse approximation, 1109–1110
- Sparse PDE constrained optimization, numerical solution, 663–672
 - finite element approximation and error estimates, 632–641
 - ihADMM algorithm, 642–652
 - imABCD method, 652–662
- Sparse reconstruction, 8, 742
- Sparse regularization
 - dictionary learning plus logarithmic domain TV, 330
 - hybrid model, 328–330
- Sparse regularization, shearlets
 - image inpainting, 1122–1124
 - image separation, 1121–1122
- Sparse-tensor discretization, 1587
- Sparse tomography, 555
 - mathematics of, 551, 557
- Sparse-view CT image, 561
- Sparse-view full-angle tomography, 556
- Sparsity, 950
 - levels, 43
- Sparsity-inducing non-separable regularizers, 22–26
- Sparsity-inducing separable regularizers, 11–12
- Spatial 3D Gaussian mixture model, 1047–1049
- Spatial domain methods, 314
- Spatial interaction models, 1055
- Spatially-adaptive denormalization (SPADE), 783
- Spatially varying bidirectional reflectance distribution function (SVBRDF), 1026
- Spatiotemporal inverse problems, 1711
 - deformable template, reconstruction, 1715–1717
 - explicit temporal models, reconstruction without, 1712–1713
 - motion model, reconstruction, 1714–1715
- SPDNorm, 783
- SPECT-CT, 236
- SPECT-MR, 236

- Spectral CT, 238, 239
 Spectral decomposition, 1655
 Spectral transforms, 1634
 Speed comparison, 452
 Spherical conformal parameterization method, 1497
 Spherical harmonics (SPHARM), 1431
 Spherical marching scheme (SMS), 1433
 Split Bregman inpainting method, 1879, 1880
 Split-Bregman iterations, 448
 Split Bregman method, 463, 468
 Splitting structure, 181–183
 Square diagonal matrix, 18
 Squared L2 fidelity, 510
 Square root velocity (SRV) framework, 1379
 closed curves, 1363
 homogenous spaces, curves in, 1366–1367
 Lie groups, curves in, 1364–1365
 open curves, 1360
 optimal reparametrizations, 1361–1363
 \mathbb{R}^d , curves in, 1359–1364
 Riemannian manifolds, curves in, 1367–1371
 SRGAN model, 1664
 Stability, 1145, 1150
 Stable multiscale medial axis map, 1856–1859
 Stable ridge/edge transform, 1844
 basic transforms, 1844–1849
 extractable corner points, 1849–1851
 interior corners, 1852–1853
 Stair-casing effect, 948
 Standard Euclidean norm, 1238
 Standard generating function, 712
 Stationary velocity field (SVF), 1293, 1294
 Statistical analysis, 1026
 Statistical data-informed (DI) inverse framework, 1245–1250
 Statistical properties, DI regularization, 1254–1256
 Statistical property based models
 general SO-model, 318
 RLO model, 318
 Stejskal–Tanner equation, 709
 Steklov–Poincaré metric, 1588–1590, 1595, 1613
 Stiffness, 642
 matrix, 642
 Stochastic EPDiff model, 1344
 Stochastic Euler–Poincaré reduction, 1337–1339
 Stochastic Galerkin method, 1587
 Stochastic games theory, 679
 Stochastic gradient descent (SGD), 69, 358, 697, 890, 905
 Stochastic gradient descent with momentum (SGDM), 358
 Stochastic linesearch method, 82
 Stochastic methods, 741
 Stochastic model problem, 1611, 1614, 1618, 1619
 Stochastic variance reduction gradient (SVRG), 72
 Straight curve application, 1565–1569
 diabetes, biomarkers for, 1563–1565
 PDEs on \mathbb{M}_3 for denoising 3D X-ray data, 1571
 PDEs on \mathbb{M}_3 for denoising FODFs in DW-MRI, 1569
 Strain tensor, 1950
 Strong contamination model, 979, 980
 Structural similarity, 243–245, 1190
 edge sets, 243
 parallel level sets, 243
 Structural similarity index measure (SSIM), 867, 919, 983–987, 990, 991, 993
 Structured image reconstruction networks, 885–886
 ADMM-Net, 891–894
 ISTA-Net, 888–891
 LDA, 898–902
 PD-Net, 896–898
 proximal point network, 886–888
 variational network, 894–897
 Structured light technology, 1741
 Structure-promoting regularizers, 261
 algorithmic parameters, 252
 anisotropic models, 247–249
 data, 252
 isotropic models, 245–247
 software, 252
 super-resolution, 255
 x-ray, 253–255
 Structure tensor, 1556
 StyleGAN2 generator, 857
 Subdifferentiable, 713
 Subdifferential smoothness, 713
 Subdifferential theory, 513
 Subgradient, 1633
 Subgradient-based methods, 916
 Sub-Riemannian distance, 1937
 Sub-Riemannian geometry, 1543
 Sub-Riemannian metric, 1935
 tensor field, 1545
 Subspace method, 162–167
 Subspace minimization, 410–412, 416–419
 Sum of squared differences (SSD), 684–686, 1311

- Supervised learning, 1188
 problem, 1074
- Supervised methods, 1296, 1310–1314
- Support vector machine (SVM), 492, 1399, 1431, 1433
- Surface analysis, for medical applications
 3D surface registration, 1424–1427
 high dimensional shape deformation, 1427
 high-dimensional shape deformation, 1430
- Surface parameterization, 1484
 definition, 1484
 mesh parameterization, 1486
 point cloud parameterization, 1487
- Surface registration, 1780–1781
 dynamic surface tracking, 1785
 optimal transport map, 1783
 quasi-conformal map, 1781–1782
 registration framework, 1781
 Teichmüller map, 1782
- Surface Ricci flow, 1753
- Surgery based metal artifact reduction (SMAR), 350, 355, 357, 359, 361, 365, 367, 373, 374
 advantage, 374
 vs. CNN-MAR, 369–371
 convergence, 357
 extended version to 3D, 360
 iterative reconstruction step, 356–357
 vs. NMAR, 365–368
 performance, 368, 375
 preprocessing step, 355
 shape prior, 361–364, 375
 shape prior, performance of, 373
- Surgery region designation, 356
- SVF-Net, 1313
- Swendsen-Wang sampling method, 1037
- Symmetric Gauss-Seidel (sGS) decomposition, 631
- Symmetric kernels, 1803
- Symmetrised gradient, 913
- SYMNet, 1307
- Synthetic aperture radar (SAR), 315, 342
- Synthetic factor, 1047
- Synthetic single-particle data recovery
 experiment, 1281
- System-Aware Compression, 191–193, 196, 197
- T**
- Tangent vector field, 1354
- Target texture principal field, 1034, 1036
- Teichmüller coordinate, 1773
- Teichmüller distance, 1418
- Teichmüller map, 1417–1418, 1493–1494, 1750–1751, 1782–1783
- Teichmüller space, 1751, 1773
- Test function, 638
- Testing, 718, 719, 734, 737
- Texture compression, 1053
- Texture editing, 1053
- Texture modeling, 1025
- Texture synthesis algorithms, 1025
- Texture synthesis and enlargement, 1050–1053
- Three-point identity, 711, 712, 718, 719, 734, 736
- Thresholded-ROF (T-ROF) method, 1392–1393
- Tietze extension theorem, 1805, 1810, 1824
- Tight-frame based method, 1400–1401
- Tikhonov functional, 1068, 1073, 1088
- Tikhonov regularization, 509, 954, 1073, 1099, 1134, 1236–1240, 1243, 1244, 1256, 1258, 1263, 1271
- Time-continuous noise, 1340
- Tomographic image reconstruction
 analytic reconstruction methods, 1186
 iterative reconstruction methods, 1186–1187
- Tomography
 acoustic, 709
 computational, 709
 electrical impedance, 709
 optical, 709
 positron emission, 709
- Tomosynthesis, 556
- Topological annulus, 1773
- Topological poly-annulus, 1774–1777
- Topological prior knowledge, 1449–1450
 digital topology, 1451–1455
 geometric flows, regularisation, 1459–1462
 higher-order schemes, for level set-based segmentation models, 1462–1463
 topology prescription, 1451–1459
- Topological quadrilateral, 1773
- Torsion, 1539, 1540
- Total deep variation, 933, 1145, 1146, 1150, 1151
 architecture, 1147
 results, 1149
 training procedure, 1148
- Total field u^{tot} , 278
- Total generalized variation (TGV), 240, 256, 531–536, 913, 934, 935, 949
 directional total generalized variation, 248
 weighted total generalized variation, 246

- Total variation (TV), 240, 325, 565, 755, 934, 935, 937, 1134, 1187, 1206, 1633, 1794, 1892
 - directional total variation, 248
 - fractional order TV, 330–331
 - joint total variation, 245
 - modified TV, 326
 - non-local TV, 326
 - prior, 1236
 - TGV, 327
 - total nuclear variation, 249
 - vectorial total variation, 245
 - weberized TV, 326
 - weighted total variation, 246
 - Total variation based multiplicative intrinsic component optimization (TVMICO), 1227–1230
 - ADMM and numerical analysis, 1218–1220
 - formulation of proposed model, 1217–1218
 - p-subproblem, 1220
 - quantitative evaluation, 1228
 - solutions for subproblems c, w and bias field estimation b, 1221
 - u-subproblem, 1221
 - v-subproblem, 1220
 - Total variation flow (TVF), 1565, 1567–1569
 - Total-variation (TV), 880
 - regularization, 1007
 - Tracking algorithms, 1529
 - Traditional segmentation, 1204
 - Trainable Deep Active Contours (TDACs), 1017
 - Training data, 1188, 1189
 - Transform
 - domain methods, 314
 - Fourier, 709
 - Radon, 709
 - Translation operator, 1101
 - Transmission conditions, 382
 - Transport cost, 1666
 - Triangle inequality, 1756
 - Triangular inequality, 1945
 - Tri-diagonal matrix, 434
 - TRish method, 79, 90
 - Truncated SVD (TSVD), 1239, 1240, 1242, 1243, 1247, 1257–1260, 1262, 1265, 1266, 1271
 - Trust-region and adaptive regularization methods, 84
 - TV-based framework, 572
 - TV-L2 restoration, 8
 - Two-dimensional Fourier transform, 285
- U**
- U-Net, 1147, 1300–1301
 - UCTGAN, 787
 - Unconstrained Lagrangian optimizations, 178–180
 - Unconstrained nonsmooth convex optimization problem, 653
 - Uniform grid, 291
 - Uniformization, 1746, 1747
 - Uniqueness, 1639
 - analysis, 169
 - Universal shearlets, 1114–1115
 - Unsupervised learning, 1188
 - Unsupervised methods, 1295, 1296, 1303
 - CNN based methods, 1305–1307
 - loss function, 1303–1305
 - regularization for diffeomorphic mapping, 1304–1305
 - similarity metrics, 1303–1304
 - VAE based methods, 1307–1309
 - UTIA gonioreflectometer, 1029
- V**
- Validation set, 1190
 - Vanilla GAN, 826
 - Variance-Reduced Randomized Kaczmarz (VR-RK) method, 1277
 - Variance reduction, 72
 - Variational autoencoders (VAEs), 778, 780, 784–786, 856, 873, 1302, 1307–1309, 1663, 1697
 - Variational-based minimization approach, 1208
 - Variational inverse problems
 - image reconstruction, 912
 - optimality and duality, 915
 - regularizers, 912–914
 - restoration models, 915
 - solution methods, 916
 - Variational models
 - conventional algorithms, 1003–1010
 - deep learning, 1011–1017
 - Variational networks, 756, 1088
 - Variational principle, 1758–1759
 - Variational regularization, 240, 1138, 1146
 - Vector quantized variational autoencoder (VQ-VAE), 785, 788, 801
 - Vertex scaling, 1758–1764
 - Vese-Chan model, 460, 1892
 - Vessel-wall-plus-plaque thickness, 1505
 - Viscous Hamilton–Jacobi PDEs
 - log-concave models, posterior mean estimators for, 225–227
 - non log-concave priors, 227–229

- Visual texture, 1027–1030
Voronoi diagram, 1858, 1859
VoxelMorph, 1305, 1732, 1733
VoxelMorph-diff, 1306
Voxels, 571
VR-RK method, 1279
- W**
- Wasserstein distance, 1143, 1190, 1661, 1662, 1666, 1667, 1670, 1695, 1793, 1801, 1802
Wasserstein GAN (WGAN), 781, 827, 971, 976
 loss, 855
Wavefront set detection, 1126–1129
Wavelength, 277
Wavelet-based segmentation method, 1401–1403
Wavelets, 1098
Wave numbers k_0 , 277
Wave speed c , 277
Weak convergence, 730
Weak convergence of measures, 1794
Weberized TV, 326
Weight sharing, 1075
Weighted Least Squares (WLS) function, 565
- Weyl-Petersson metric, 1751
Wishart distribution density, 1044
Woodbury formula, 1251
- X**
- X-ray computed tomography, 4, 348, 1190, 1198–1199
 analytic reconstruction, 1191–1192
 limited angle computed tomography, 1192–1196
 low dose computed tomography, 1197–1198
 metal artefacts, reduction of, 1196–1197
X-ray crystallography, 1275
X-ray free electron lasers (XFEL), 1275
X-ray tomography, 1265–1268, 1270
- Y**
- Yamabe energy, 1760, 1764
Yamabe equations, 1753
YUV-L2 zebra, 868
YUV-LPIPS zebra, 868
- Z**
- Zelnik-Manor function, 1919