

Xue-Cheng Tai Egil Bae  
Tony F. Chan Marius Lysaker (Eds.)

LNCS 8932

# Energy Minimization Methods in Computer Vision and Pattern Recognition

10th International Conference, EMMCVPR 2015  
Hong Kong, China, January 13–16, 2015  
Proceedings



Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Germany*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbruecken, Germany*

Xue-Cheng Tai Egil Bae Tony F. Chan  
Marius Lysaker (Eds.)

# Energy Minimization Methods in Computer Vision and Pattern Recognition

10th International Conference, EMMCVPR 2015  
Hong Kong, China, January 13-16, 2015  
Proceedings



Springer

## Volume Editors

Xue-Cheng Tai  
University of Bergen, Department of Mathematics  
Bergen, Norway  
E-mail: tai@math.uib.no

Egil Bae  
University of California, Department of Mathematics  
Los Angeles, CA, USA  
E-mail: ebae@math.ucla.edu

Tony F. Chan  
The Hong Kong University of Science and Technology  
Clear Water Bay, Kowloon, Hong Kong, S.A.R.  
E-mail: tonyfchan@ust.hk

Marius Lysaker  
Telemark University College  
Porsgrunn, Norway  
E-mail: marius.lysaker@hit.no

ISSN 0302-9743 e-ISSN 1611-3349  
ISBN 978-3-319-14611-9 e-ISBN 978-3-319-14612-6  
DOI 10.1007/978-3-319-14612-6  
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014958022

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision,  
Pattern Recognition, and Graphics

© Springer International Publishing Switzerland 2015

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. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

Energy minimization has become an important paradigm for solving many challenging problems within computer vision and pattern recognition over the past few decades. Mathematical models that describe the desired solution as the minimizer of an energy potential arise through different schools of thought, including statistical approaches in the form of Markov random fields and geometrical approaches in the form of variational models or equivalent partial differential equations. Besides the challenge of formulating appropriate energy minimization models, a significant research topic is the design of computational methods for reliably and efficiently obtaining solutions of minimal energy.

This book contains 36 original research articles that cover the whole spectrum of energy minimization in computer vision and pattern recognition, including design and analysis of mathematical models and design of discrete and continuous optimization algorithms. Application areas include image segmentation and tracking, image restoration and inpainting, multiview reconstruction, shape optimization, and texture and color analysis. The articles have been carefully selected through a thorough double-blind peer-review process.

Furthermore, we were delighted that three internationally recognized experts in the fields of computer vision, pattern recognition, and optimization, namely, Andrea Bertozzi (UCLA), Ron Kimmel (Technion-IIT), and Long Quan (HKUST), agreed to further enrich the conference with inspiring keynote lectures.

We would like to express our gratitude to those who made this event possible and contributed to its success. In particular, our Program Committee of top international experts in the field provided excellent reviews. The administrative and financial support from the Hong Kong University of Science and Technology (HKUST), especially from HKUST Jockey Club Institute for Advanced Study (IAS), was crucial for the success of this event. We are grateful to Linus See (HKUST), Eric Lin (HKUST) and Shing Yu Leung (HKUST) for providing very helpful local administrative support. It is our belief that this conference helped to advance the field of energy minimization methods and to further establish the mathematical foundations of computer vision and pattern recognition.

November 2014

Xue-Cheng Tai  
Egil Bae  
Tony F. Chan  
Marius Lysaker

# Organization

EMMCVPR 2015 was organized by the HKUST Jockey Club Institute for Advanced Study (IAS).

## Executive Committee

### Conference Chair

Xue-Cheng Tai                              University of Bergen, Norway

### Organizers

Egil Bae                                      UCLA, USA  
Tony F. Chan                              HKUST, Hong Kong  
Marius Lysaker                              Telemark University College, Norway  
Shing Yu Leung                              HKUST, Hong Kong

### Invited Speakers

Andrea Bertozzi                              University of California at Los Angeles, USA  
Ron Kimmel                                  Technion-IIT, Israel  
Yi Ma    ShanghaiTech, China  
Long Quan                                      HKUST, Hong Kong

### Program Committee

J.-F. Aujol	B. Flach	C. Schnoerr
M. Björkman	D. Geiger	C.-B. Schonlieb
M. Blaschko	H. Ishikawa	A. Schwing
A. Bruhn	D. Jacobs	F. Sgallari
R. Chan	F. Kahl	A. Shekhovtsov
X. Chen	R. Kimmel	H. Talbot
J. Clark	I. Kokkinos	W. Tao
D. Cremers	A. S. Konushin	O. Veksler
J. Darbon	S. Li	J. Weickert
G. Doretto	H. Li	O. Woodford
P. Favaro	S. Maybank	X. Wu
M. Felsberg	M. Nikolova	C. Wu
M. Figueiredo	M. Pelillo	J. Yuan
A. Fix	T. Pock	J. Zerubia

### Sponsoring Institutions

HKUST Jockey Club Institute for Advanced Study

# Table of Contents

## Discrete and Continuous Optimization

Convex Envelopes for Low Rank Approximation . . . . .	1
<i>Viktor Larsson and Carl Olsson</i>	
Maximizing Flows with Message-Passing: Computing Spatially Continuous Min-Cuts . . . . .	15
<i>Egil Bae, Xue-Cheng Tai, and Jing Yuan</i>	
A Compact Linear Programming Relaxation for Binary Sub-modular MRF . . . . .	29
<i>Junyan Wang and Sai-Kit Yeung</i>	
On the Link between Gaussian Homotopy Continuation and Convex Envelopes . . . . .	43
<i>Hossein Mobahi and John W. Fisher III</i>	
How Hard Is the LP Relaxation of the Potts Min-Sum Labeling Problem? . . . . .	57
<i>Daniel Průša and Tomáš Werner</i>	
Coarse-to-Fine Minimization of Some Common Nonconvexities . . . . .	71
<i>Hossein Mobahi and John W. Fisher III</i>	

## Image Restoration and inpainting

Why Does Non-binary Mask Optimisation Work for Diffusion-Based Image Compression? . . . . .	85
<i>Laurent Hoeltgen and Joachim Weickert</i>	
Expected Patch Log Likelihood with a Sparse Prior . . . . .	99
<i>Jeremias Sulam and Michael Elad</i>	
Blind Deconvolution via Lower-Bounded Logarithmic Image Priors . . . . .	112
<i>Daniele Perrone, Remo Diethelm, and Paolo Favaro</i>	
Low Rank Priors for Color Image Regularization . . . . .	126
<i>Thomas Möllenhoff, Evgeny Strelakovsky, Michael Moeller, and Daniel Cremers</i>	
A Novel Framework for Nonlocal Vectorial Total Variation Based on $\ell^{p,q,r}$ -norms . . . . .	141
<i>Joan Duran, Michael Moeller, Catalina Sbert, and Daniel Cremers</i>	

Inpainting of Cyclic Data Using First and Second Order Differences . . . .	155
<i>Ronny Bergmann and Andreas Weinmann</i>	

Discrete Green's Functions for Harmonic and Biharmonic Inpainting with Sparse Atoms . . . . .	169
<i>Sebastian Hoffmann, Gerlind Plonka, and Joachim Weickert</i>	

## Segmentation

A Fast Projection Method for Connectivity Constraints in Image Segmentation . . . . .	183
<i>Jan Stühmer and Daniel Cremers</i>	

Two-Dimensional Variational Mode Decomposition . . . . .	197
<i>Konstantin Dragomiretskiy and Dominique Zosso</i>	

Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme . . . . .	209
<i>Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi</i>	

A Novel Active Contour Model for Texture Segmentation . . . . .	223
<i>Aditya Tatu and Sumukh Bansal</i>	

Segmentation Using SubMarkov Random Walk . . . . .	237
<i>Xingping Dong, Jianbing Shen, and Luc Van Gool</i>	

Automatic Shape Constraint Selection Based Object Segmentation . . . .	249
<i>Kunqian Li, Wenbing Tao, Xiangli Liao, and Liman Liu</i>	

## PDE and Variational Methods

Justifying Tensor-Driven Diffusion from Structure-Adaptive Statistics of Natural Images . . . . .	263
<i>Pascal Peter, Joachim Weickert, Axel Munk, Tatyana Krivobokova, and Housen Li</i>	

Variational Time-Implicit Multiphase Level-Sets: A Fast Convex Optimization-Based Solution . . . . .	278
<i>Martin Rajchl, John S.H. Baxter, Egil Bae, Xue-Cheng Tai, Aaron Fenster, Terry M. Peters, and Jing Yuan</i>	

An Efficient Curve Evolution Algorithm for Multiphase Image Segmentation . . . . .	292
<i>Günay Doğan</i>	

A Tensor Variational Formulation of Gradient Energy Total Variation . . . . .	307
<i>Freddie Åström, George Baravdish, and Michael Felsberg</i>	



Color Image Segmentation by Minimal Surface Smoothing . . . . .	321
<i>Zhi Li and Tieyong Zeng</i>	

Domain Decomposition Methods for Total Variation Minimization . . . . .	335
<i>Huibin Chang, Xue-Cheng Tai, and Danping Yang</i>	

## Motion, Tracking and Multiview Reconstruction

A Convex Solution to Disparity Estimation from Light Fields via the Primal-Dual Method . . . . .	350
<i>Mahdad Hosseini Kamal, Paolo Favaro, and Pierre Vanderghelynst</i>	

Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames . . . . .	364
<i>Ryan Kennedy and Camillo J. Taylor</i>	

Adaptive Dictionary-Based Spatio-temporal Flow Estimation for Echo PIV . . . . .	378
<i>Ecaterina Bodnariuc, Arati Gurung, Stefania Petra, and Christoph Schnörr</i>	

Point Sets Matching by Feature-Aware Mixture Point Matching Algorithm . . . . .	392
<i>Kun Sun, Peiran Li, Wenbing Tao, and Liman Liu</i>	

## Motion, Tracking and Multiview Reconstruction

Multi-utility Learning: Structured-Output Learning with Multiple Annotation-Specific Loss Functions . . . . .	406
<i>Roman Shapovalov, Dmitry Vetrov, Anton Osokin, and Pushmeet Kohli</i>	

Mapping the Energy Landscape of Non-convex Optimization Problems . . . . .	421
<i>Maira Pavlovskaja, Kewei Tu, and Song-Chun Zhu</i>	

Marked Point Process Model for Curvilinear Structures Extraction . . . . .	436
<i>Seong-Gyun Jeong, Yuliya Tarabalka, and Josiane Zerubia</i>	

Randomly Walking Can Get You Lost: Graph Segmentation with Unknown Edge Weights . . . . .	450
<i>Hanno Ackermann, Björn Scheuermann, Tat-Jun Chin, and Bodo Rosenhahn</i>	

## Medical Image Analysis

Training of Templates for Object Recognition in Invertible Orientation Scores: Application to Optic Nerve Head Detection in Retinal Images . . .	464
<i>Erik Bekkers, Remco Duits, and Marco Loog</i>	
A Technique for Lung Nodule Candidate Detection in CT Using Global Minimization Methods . . . . .	478
<i>Nóirín Duggan, Egil Bae, Shiwen Shen, William Hsu, Alex Bui, Edward Jones, Martin Glavin, and Luminita Vese</i>	
Hierarchical Planar Correlation Clustering for Cell Segmentation . . . . .	492
<i>Julian Yarkony, Chong Zhang, and Charless C. Fowlkes</i>	
<b>Author Index . . . . .</b>	<b>505</b>

# Convex Envelopes for Low Rank Approximation

Viktor Larsson and Carl Olsson

Centre for Mathematical Sciences  
Lund University, Sweden

**Abstract.** In this paper we consider the classical problem of finding a low rank approximation of a given matrix. In a least squares sense a closed form solution is available via factorization. However, with additional constraints, or in the presence of missing data, the problem becomes much more difficult. In this paper we show how to efficiently compute the convex envelopes of a class of rank minimization formulations. This opens up the possibility of adding additional convex constraints and functions to the minimization problem resulting in strong convex relaxations. We evaluate the framework on both real and synthetic data sets and demonstrate state-of-the-art performance.<sup>1</sup>

## 1 Introduction

The assumption that measurements consist of noisy observations from a low rank matrix has been proven useful in applications such as non-rigid and articulated structure from motion [1,2,3], photometric stereo [4] and optical flow [5]. The interpretation of the low rank assumption is that the observed data can be written as a linear combination of a few basis elements. The factorization approach, introduced to vision in [6], offers a simple way of determining both coefficients and basis elements. If the measurement matrix  $M$  is complete then the best approximation, in a least squares sense, can be computed in closed form [7] using the singular value decomposition (SVD). The main drawback is that the computation of a factorization requires a complete measurement matrix. In structure from motion this means that every point has to be visible in every image, something that rarely occurs in practice due to occlusions and tracking failures. In case there are missing entries and/or outliers the optimization problem is substantially more difficult.

The issue of outliers has received a lot of attention lately. In [8,9] the more robust  $L_1$ -norm is considered. These methods build on the so called Wiberg algorithm [10] which jointly optimizes a product  $UV^T$  of two fixed size  $U$  and  $V$  matrices. As a consequence the quality of the result is dependent on initialization. Another approach [11,3,12] tackles the problem of missing data by replacing the rank constraint with the weaker but convex nuclear norm penalty and solves

$$\min_X \mu \|X\|_* + \|W \odot (X - M)\|_F^2, \quad (1)$$

---

<sup>1</sup> This work has been funded by the Swedish Research Council (grant no. 2012-4213) and the Crafoord Foundation.

where  $W_{ij} = 0$  if the entry is missing and 1 otherwise. This approach is convex and therefore independent of initialization. In addition it can be shown that if the locations of the missing entries are random the approach gives the best low rank approximation [11]. The typical patterns of missing data in structure from motion still pose a problem for these approaches.

The motivation for using the nuclear norm in (1) is that it is the convex envelope of the rank function on the set  $\{X; \sigma_{max}(X) \leq 1\}$ . The constraint  $\sigma_{max}(X) \leq 1$  is however artificial and not present in (1). In [13] it is shown that the so called localized rank function

$$f(X) = \mu \text{rank}(X) + \|X - X_0\|_F^2, \quad (2)$$

has the convex envelope

$$f^{**}(X) = \sum_{i=1}^n \left( \mu - [\sqrt{\mu} - \sigma_i(X)]_+^2 \right) + \|X - X_0\|_F^2. \quad (3)$$

Note that the regularizer in (3) itself is not convex. The second term, enables a proportionally smaller penalty for large singular values, without losing convexity, giving a tighter convex envelope in the neighborhood of  $X_0$ . In fact, in contrast to the nuclear norm heuristic, minimizing (3) gives the same result as solving (2) with SVD. The advantage of using (3) is that it is convex and therefore can be combined with other convex constraints and functions. In [13] the missing data problem is solved by minimizing (3) on complete sub-blocks and enforcing agreement on the overlaps via linear constraints.

The formulation in [13] consists of a trade-off between matrix rank and data fit. In many cases it is of interest to search for a matrix of known fixed rank. For example for rigid structure from motion the measurement matrix is known to be of rank 4 (or 3 if the translation can be eliminated) [6]. In such cases the approach of solving (3) on sub-blocks requires determining an appropriate weight  $\mu$  for each sub-block that gives the correct rank. In this paper we show that we can incorporate such knowledge by replacing (2) with

$$f_g(X) = g(\text{rank}(X)) + \|X - X_0\|^2. \quad (4)$$

In particular we are interested in the case where

$$g(\text{rank}(X)) = \mu \max(r_0, \text{rank}(X)), \quad (5)$$

but our theory applies to a larger class of problems as well. The only requirement that we make is that  $g$  is a non-decreasing convex function.

The reason for considering (5) is that in case we know the rank of the sought matrix we can simply let  $\mu$  be large thus avoiding iteration over the parameters which is done in [13]. Consequently our approach is essentially parameter free. The max term also effectively reduces bias towards low rank solutions like the zero solution that are often uninteresting, giving a tighter convex relaxation. Our main contribution is the computation of the convex envelope of (4) and its proximal operator. While the formulation does not admit closed form solutions we give simple and fast algorithms for evaluations. In addition we present a way of strengthening the convex envelopes using a trust-region formulation.

**Notation.** Throughout the paper we use  $\sigma_i(X)$ ,  $i = 1, \dots, n$  to denote the  $i$ th singular value of a matrix  $X$ . Here  $n$  denotes the number of singular values and for notational convenience we will also define  $\sigma_0(X) = \infty$  and  $\sigma_{n+1}(X) = 0$ . The vector of all singular values is denoted  $\boldsymbol{\sigma}(X)$ . With some abuse of notation we write the SVD of  $X$  as  $U \text{diag}(\boldsymbol{\sigma}(X))V^T$ . For ease of notation we do not explicitly indicate the dependence of  $U$  and  $V$  on  $X$ . The scalar product is defined as  $\langle X, Y \rangle = \text{tr}(X^T Y)$ , where  $\text{tr}$  is the trace function, and the Frobenius norm  $\|X\|_F = \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i=1}^n \sigma_i^2(X)}$ . Truncation at zero is denoted  $[a]_+$ , that is,  $[a]_+ = 0$  if  $a < 0$  and  $a$  otherwise.

## 2 The Convex Envelope

In this section we compute the envelope of (4). We will assume that the function  $g$  can be written

$$g(k) = \begin{cases} g_0 & \text{if } k = 0 \\ g_0 + \sum_{i=1}^k g_i & \text{otherwise} \end{cases}, \quad (6)$$

where the sequence  $g_i$  is non-negative and non-decreasing for  $1 \leq i \leq n$ . It is easy to see that this is possible if  $g$  is convex and non decreasing on  $\mathbb{R}$ . Furthermore, we will assume that  $g_0 = 0$  since subtracting a constant from the objective function does not affect the minimizers (and only subtracts a constant from the convex envelope).

We will follow the approach of [13] which computes the bi-conjugate of (2) to find the convex envelope. In contrast to (2), we will not be able to find a closed form solution for the convex envelope of (4). Instead our approach will be to isolate a small set of singular value configurations that can possibly maximize the conjugate function. By numerically searching this solution set we are able to efficiently evaluate the convex envelope and compute its proximal operator.

### 2.1 The Conjugate Function

The convex envelope can be found by computing the second Fenchel conjugate  $f_g^{**} = (f_g^*)^*$ , where  $f_g^*$  is defined as

$$f_g^*(Y) = \sup_X \langle X, Y \rangle - f_g(X). \quad (7)$$

The calculations for the first conjugate roughly follows those of [13] and we only give the result here. We get that the first conjugate is given by

$$f_g^*(Y) = - \sum_{i=1}^n \min \left( g_i, \sigma_i^2 \left( X_0 + \frac{Y}{2} \right) \right) - \frac{1}{2} \|X_0\|_F^2 + \left\| X_0 + \frac{Y}{2} \right\|_F^2. \quad (8)$$

### 2.2 Evaluation of the Bi-conjugate

By completing squares and changing variables we get the bi-conjugate

$$f_g^{**}(X) = \mathcal{R}_g(X) + \|X - X_0\|_F^2, \quad (9)$$

where

$$\mathcal{R}_g(X) = \max_Z \left( \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \|Z - X\|_F^2 \right). \quad (10)$$

The next step in determining the convex envelope is to find the maximizing  $Z$  in (10). We first note that using von Neumann's trace theorem we can reduce the problem to a search over the singular values of  $Z$ . The norm term fulfills

$$-\|Z - X\|_F^2 \leq -\|Z\|_F^2 + 2 \sum_{i=1}^n \sigma_i(Z)\sigma_i(X) - \|X\|_F^2, \quad (11)$$

with equality if  $Z$  and  $X$  have the same  $U$  and  $V$  in their singular value decompositions. Since the sum in (10) does not depend on  $U$  or  $V$  the optimal  $Z$  has to be of the form  $Z = U \text{diag}(\boldsymbol{\sigma}(Z))V^T$  if  $X = U \text{diag}(\boldsymbol{\sigma}(X))V^T$ . This reduces the maximization in (10) to

$$\max_{\boldsymbol{\sigma}(Z)} \left( \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \sum_{i=1}^n (\sigma_i(Z) - \sigma_i(X))^2 \right). \quad (12)$$

Note that the elements of  $\boldsymbol{\sigma}(Z)$  have to fulfill  $\sigma_1(Z) \geq \sigma_2(Z) \geq \dots \geq \sigma_n(Z)$  since these are singular values.

**Properties of the Optimal  $\boldsymbol{\sigma}(Z)$ .** To limit the search space for maximization over  $\boldsymbol{\sigma}(Z)$  we will next derive some properties of the maximizer. Considering each singular value  $\sigma_k(Z)$  separately they should solve a program of the type

$$\max_s \min(g_k, s^2) - (s - \sigma_k(X))^2 \quad (13)$$

$$\text{s.t. } \sigma_{k+1}(Z) \leq s \leq \sigma_{k-1}(Z) \quad (14)$$

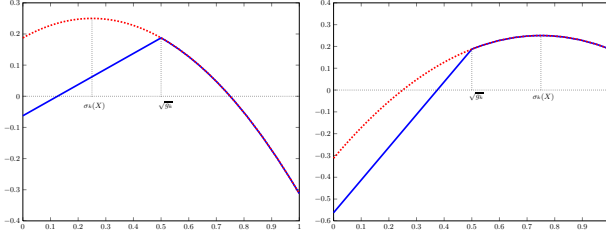
Note that for  $k = 1$  there is no upper bound on  $s$  and for  $k = n$  there is no positive lower bound since we use the convention that  $\sigma_0(Z) = \infty$  and  $\sigma_{n+1}(Z) = 0$ . We first consider the unconstrained objective function. This function is the point wise minimum of the two concave functions  $g_k - (s - \sigma_k(X))^2$  (for  $s \geq \sqrt{g_k}$ ) and  $s^2 - (s - \sigma_k(X))^2 = 2s\sigma_k(X) - \sigma_k^2(X)$ . The function is concave and attains its optimum in  $s = \sigma_k(X)$  if  $\sigma_k(X) \geq \sqrt{g_k}$  and in  $s = \sqrt{g_k}$  otherwise (see Figure 1). In case  $\sigma_k(X) = 0$  the optimum is not unique. For simplicity we will assume that  $\sigma_k(X) > 0$  in what follows. The solution we create will still be valid if  $\sigma_k(X) = 0$  but might not be unique. Let  $s_k$  be the individual unconstrained optimizers of (13), i.e.

$$s_k = \max(\sqrt{g_k}, \sigma_k(X)). \quad (15)$$

Note that this sequence is decreasing when  $\sigma_k(X)$  is larger than  $\sqrt{g_k}$ . We choose  $k_0$  such that  $s_{k_0}$  is the smallest value in the sequence  $s_k$ .

We now consider the constrained problem (13)-(14). Since  $\sigma_{k+1}(Z) \leq \sigma_{k-1}(Z)$  we see that the optimization over  $\sigma_k(Z)$  can be limited to three choices

$$\sigma_k(Z) = \begin{cases} s_k & \text{if } \sigma_{k+1}(Z) \leq s_k \leq \sigma_{k-1}(Z) \\ \sigma_{k-1}(Z) & \text{if } \sigma_{k-1}(Z) < s_k \\ \sigma_{k+1}(Z) & \text{if } s_k < \sigma_{k+1}(Z) \end{cases}. \quad (16)$$



**Fig. 1.** The objective function in (13) for  $\sigma_k(X) \leq \sqrt{g_k}$  and  $\sigma_k(X) \geq \sqrt{g_k}$

**Lemma 1.** *If  $Z$  is an optimal solution to (12) then there is a  $k \leq k_0$  such that*

$$\sigma_i(Z) = s_i, \quad \text{if } i < k, \quad (17)$$

$$\sigma_i(Z) = \sigma_k(Z), \quad \text{if } k \leq i \leq k_0. \quad (18)$$

*Proof.* Using induction we first prove the recursion

$$\sigma_i(Z) = \max(s_i, \sigma_{i+1}(Z)) \quad \text{for } i \leq k_0. \quad (19)$$

For  $i = 1$  we see from (16) that  $s_1$  is the optimal choice if  $s_1 > \sigma_2(Z)$  otherwise  $\sigma_2(Z)$  is optimal. Therefore  $\sigma_1(Z) = \max(s_1, \sigma_2(Z))$ . Next assume that  $\sigma_{i-1}(Z) = \max(s_{i-1}, \sigma_i(Z))$  for some  $i \leq k_0$ . Then

$$\sigma_{i-1}(Z) \geq s_{i-1} \geq s_i, \quad (20)$$

therefore we can ignore the second case in (16), which proves the recursion (19).

To prove the lemma assume  $\sigma_k(Z) \neq s_k$  for some  $k \leq k_0$ . From (19) it follows that

$$\sigma_k(Z) = \sigma_{k+1}(Z) > s_k. \quad (21)$$

But  $s_k$  is decreasing for  $k \leq k_0$  which implies that  $\sigma_{k+1}(Z) > s_{k+1}$ . By repeating the argument it follows that

$$\sigma_k(Z) = \sigma_{k+1}(Z) = \sigma_{k+2}(Z) = \dots = \sigma_{k_0}(Z). \quad (22)$$

□

**Lemma 2.** *If  $Z$  is an optimal solution to (12) then*

$$\sigma_i(Z) = \sigma_{i+1}(Z), \quad \text{if } i \geq k_0. \quad (23)$$

*Proof.* Consider  $\sigma_i(Z)$  for some  $i \geq k_0$ . If  $\sigma_i(Z) > s_i$  it must have been bounded from below in (16), i.e.  $\sigma_i(Z) = \sigma_{i+1}(Z)$ . If instead  $\sigma_i(Z) \leq s_i$  we have  $\sigma_{i+1}(Z) \leq \sigma_i(Z) \leq s_i \leq s_{i+1}$ . Then similarly  $\sigma_{i+1}(Z)$  is bounded from above in (16) which implies  $\sigma_{i+1}(Z) = \sigma_i(Z)$ .

□

**Algorithm.** We now summarize the properties derived in the previous section into an algorithm. Since we do not know which value the  $k$  of Lemma 1 will take the algorithm essentially consists of looping over  $k$  and testing the obtained solutions for feasibility. Furthermore the operations in each iteration are fast so that in practice the search for  $k$  is dominated by other steps such as computation of the singular value decomposition of  $X$ .

From the previous section it follows that the optimal solutions  $\sigma(Z)$  must have the form

$$\sigma_i(Z) = \begin{cases} \sigma_i(X) & i \leq k \\ s & i > k \end{cases}, \quad (24)$$

for some  $k \leq k_0$  and  $s \leq \sigma_k(X)$ . We can find the optimal  $k$  and  $s$  by considering the following optimization problem

$$\max_{k \leq k_0} \max_s \sum_{i=1}^k g_i + \sum_{i=k+1}^n \min(s^2, g_i) - \sum_{i=k+1}^n (s - \sigma_i(X))^2. \quad (25)$$

For a fixed  $k < k_0$  it follows from Lemma 1 that  $s^* = \sigma_{k+1}(Z)$  must satisfy

$$\sigma_{k+1}(X) \leq \sigma_{k+1}(Z) \leq \sigma_k(Z) = \sigma_k(X). \quad (26)$$

Thus for each  $k < k_0$  we only need to consider  $s$  in the interval  $[\sigma_{k+1}(X), \sigma_k(X)]$ . Since  $g_i$  are increasing we can further divide this interval into subintervals. We let  $I_l = [\sqrt{g_{k_l}}, \sqrt{g_{k_l+1}}]$ , where  $\sqrt{g_{k_l}}$ ,  $l = 1, \dots, m-1$  is the subsequence with terms in the (open) interval  $(\sigma_{k+1}(X), \sigma_k(X))$ . Furthermore, we let  $I_0 = [\sigma_{k+1}(X), \sqrt{g_{k_1}}]$  and  $I_m = [\sqrt{g_{k_m}}, \sigma_k(X)]$ . Note that on each of these subintervals the objective function can be written as a concave quadratic function

$$f_l^k(s) = \sum_{g_i \leq g_{k_l}} g_i + \sum_{g_i > g_{k_l}} s^2 - \sum_{i=k+1}^n (s - \sigma_i(X))^2, \quad s \in I_l \quad (27)$$

We can therefore rewrite the inner optimization in (25) as the piecewise smooth problem

$$\max_{0 \leq l \leq m} \max_{s \in I_l} f_l^k(s). \quad (28)$$

The optimum must lie at either a feasible stationary point of  $f_l^k$  or at one of the boundaries of  $I_l$  for some  $l$ . To find the optimal  $s$  we can simply enumerate all the possibilities and choose the maximizing one. Since each  $\sqrt{g_i}$  only lies in one of the intervals  $[\sigma_{k+1}(X), \sigma_k(X)]$  we only need to consider each  $g_i$  once. This makes the number of possible solutions depend linearly on the number of singular values.

The steps of the method are summarized in Algorithm 1.



**Data:**  $X, g$   
**Result:**  $\sigma(Z^*)$   
**for**  $k = 0 : k_0$  **do**  
    Compute  $s^*$  and  $l^*$  from (28);  
    **if**  $f_{l^*}^k(s^*) > f_{opt}$  **then**  
         $\sigma_i(Z^*) := \sigma_i(X), \quad \forall i < k;$   
         $\sigma_i(Z^*) := s^*, \quad \forall i \geq k;$   
         $f_{opt} := f_{l^*}^k(s^*);$   
    **end**  
**end**

**Algorithm 1:** Finding maximizing  $Z$  for (10)

### 2.3 The Proximal Operator of $f_g^{**}$

In order to optimize the convex envelope  $f_g^{**}(X)$  efficiently we need to be able to compute its proximal operator

$$\text{prox}_{f_g^{**}}(M) = \underset{X}{\text{argmin}} f_g^{**}(X) + \rho \|X - M\|_F^2. \quad (29)$$

The approach we will take is similar to how we evaluate  $f_g^{**}(X)$  itself but will require looping over two variables instead of one. The key observation is that switching the order of the minimization over  $X$  with maximization over  $Z$  enables us to characterize optimal solutions similarly to Section 2.2.<sup>2</sup> We therefore solve

$$\max_Z \min_X \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \|X - Z\|_F^2 + \|X - X_0\|_F^2 + \rho \|X - M\|_F^2. \quad (30)$$

The inner minimization in  $X$  is a simple least squares problem. By completing squares one sees that the optimal  $X$  is given by

$$X = M + \frac{X_0 - Z}{\rho}. \quad (31)$$

Inserting into (30) we get after some manipulations

$$\max_Z \sum_{i=1}^n \min(g_i, \sigma_i^2(Z)) - \frac{\rho + 1}{\rho} \|Z - Y\|_F^2 + C, \quad (32)$$

where  $C$  is a constant that does not depend on  $Z$  and

$$Y = \frac{X_0 + \rho M}{1 + \rho}. \quad (33)$$

<sup>2</sup> If  $\rho > 0$  the objective function is closed, proper convex-concave, continuous and the optimization can be restricted to a compact set. Switching optimization order is therefore justified by the existence of a saddle point, see [14].

Therefore we see that the singular value  $\sigma_k(Z)$  must solve the problem

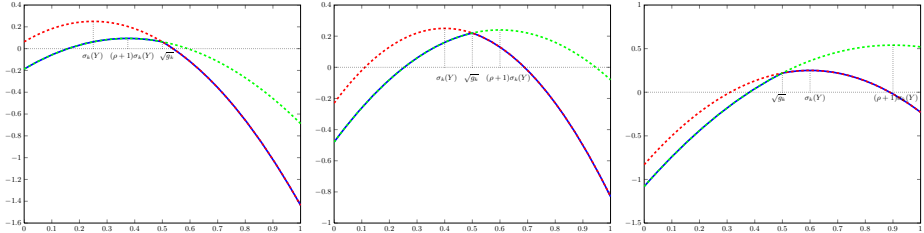
$$\max_s \min(g_i, s^2) - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2 \quad (34)$$

$$\text{s.t. } \sigma_{k+1}(Z) \leq s \leq \sigma_{k-1}(Z). \quad (35)$$

The objective function (34) is the pointwise minimum of the two quadratic strictly (assuming  $\rho > 0$ ) concave functions,

$$q_1(s) = g_i - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2, \quad q_2(s) = s^2 - \frac{\rho+1}{\rho}(s - \sigma_k(Y))^2. \quad (36)$$

The objective function is equal to  $q_1(s)$  for  $s \geq \sqrt{g_k}$  and  $q_2(s)$  otherwise. The functions  $q_1$  and  $q_2$  attain their maximum values at  $s = \sigma_k(Y)$  and  $s = (\rho+1)\sigma_k(Y)$  respectively. Note that since  $(\rho+1)\sigma_k(Y) > \sigma_k(Y)$  at most one of these can be feasible. It can also happen that neither is feasible, i.e.  $\sigma_k(Y) \leq \sqrt{g_k} \leq (\rho+1)\sigma_k(Y)$ . In this case the optimal  $s = \sqrt{g_k}$ . Figure 2 illustrates the shape of the objective function in the three possible cases.



**Fig. 2.** The objective function in (34) for left:  $(\rho+1)\sigma_k(Y) \leq \sqrt{g_k}$ , middle:  $\sigma_k(Y) \leq \sqrt{g_k}$  and  $(\rho+1)\sigma_k(Y) \geq \sqrt{g_k}$  and right:  $\sigma_k(Y) \geq \sqrt{g_k}$

Let  $s_k$  be the individual unconstrained maximizers of (34), i.e.

$$s_k = \begin{cases} \sigma_k(Y) & \text{if } \sigma_k(Y) \geq \sqrt{g_k} \\ \sqrt{g_k} & \text{if } \sigma_k(Y) \leq \sqrt{g_k} \leq (\rho+1)\sigma_k(Y) \\ (\rho+1)\sigma_k(Y) & \text{if } (\rho+1)\sigma_k(Y) \leq \sqrt{g_k} \end{cases}. \quad (37)$$

**Lemma 3.** *If  $Z$  is optimal in (32) then there is  $k_1$  and  $k_2$  such that*

$$\sigma_i(Z) = s_i, \quad \text{if } i < k_1 \quad (38)$$

$$\sigma_i(Z) = s^*, \quad \text{if } k_1 \leq i \leq k_2 \quad (39)$$

$$\sigma_i(Z) = s_i, \quad \text{if } i > k_2, \quad (40)$$

where  $s^*$  solves

$$\max_s \sum_{i=k_1}^{k_2} \min(g_i, s^2) - \frac{\rho+1}{\rho}(s - \sigma_i(Y))^2 \quad (41)$$

$$\text{s.t. } \sigma_{k_2+1}(Z) \leq s \leq \sigma_{k_1-1}(Z). \quad (42)$$

*Proof.* By construction there will exist  $p, q \in \mathbb{N}$  with  $p \leq q$  such that  $s_i$  is; decreasing for  $1 \leq i \leq p$ , increasing for  $p \leq i \leq q$  and decreasing for  $q \leq i \leq n$ . For  $1 \leq i \leq q$  we are in the same situation as in Lemma 1 and Lemma 2 with  $k_0 = p$ .

Consider now instead  $i \geq q$ . We will show that

$$\sigma_i(Z) = \min(s_i, \sigma_{i-1}(Z)) \quad \text{for } i \geq q. \quad (43)$$

It is clear from (16) that this holds for  $i = n$ . We continue using induction by assuming  $\sigma_{i+1}(Z) = \min(s_{i+1}, \sigma_i(Z))$  holds. Then

$$\sigma_{i+1}(Z) \leq s_{i+1} \leq s_i, \quad (44)$$

since  $s_i$  are decreasing for  $i \geq q$ . This means that for  $\sigma_i(Z)$  we can ignore the third case in (16). Thus it follows that  $\sigma_i(Z) = \min(s_i, \sigma_{i-1}(Z))$ . So (43) holds for all  $i \geq q$ .

Now assume that for some  $i \geq q$  we have  $\sigma_i(Z) \neq s_i$ . By (43) we must have that

$$\sigma_i(Z) = \sigma_{i-1}(Z) < s_i \leq s_{i-1}. \quad (45)$$

By repeating the argument we get

$$\sigma_i(Z) = \sigma_{i-1}(Z) = \sigma_{i-2}(Z) = \dots = \sigma_q(Z), \quad (46)$$

and the result follows.  $\square$

**Algorithm.** The properties listed in Lemma 3 allows us to find the optimal  $Z$  by searching over the two parameters  $k_1$  and  $k_2$ . The goal is to find all sequences  $\sigma_i(Z)$  of the type given in the lemma and determine which one gives the best objective value. For fixed  $k_1$  and  $k_2$  the problem in (41) is a piecewise smooth problem similar to (13) which we can solve in the same way by considering the feasible stationary points as well as the boundaries. Note that for feasible solutions we must have  $1 \leq k_1 \leq p$  and  $q \leq k_2 \leq n$ . We outline the steps in Algorithm 2.

### 3 Block Decomposition with ADMM

Next we consider the problem of missing data. The approach we take here follows [13] and we only give a very brief account of it here for completeness. The idea is to try to enforce low rank of sub-blocks of the matrix where no measurements are missing using our convex relaxation. We seek to minimize the non-convex function

$$f(X) = \sum_{i=1}^K g(\text{rank}(\mathcal{P}_i(X))) + \|\mathcal{P}_i(X) - \mathcal{P}_i(M)\|_F^2, \quad (47)$$

by replacing it with the convex relaxation

$$f_{\mathcal{R}}(X) = \sum_{i=1}^K \mathcal{R}_g(\mathcal{P}_i(X)) + \|\mathcal{P}_i(X) - \mathcal{P}_i(M)\|_F^2. \quad (48)$$

**Data:**  $X_0, \rho, \mu, M$   
**Result:** Set of possible solutions  $S$   
 $S := \emptyset$ ;  
 Define  $p, q$  as in proof of Lemma 3;  
**if**  $s_i$  *is decreasing with*  $i$  **then**  
      $S := \{s_i\}$ ;  
     return;  
**else**  
     **for**  $k_1 = 1 : p$  **do**  
         **for**  $k_2 = q : n$  **do**  
             Compute  $s^*$  from (41) and form  $\sigma(Z)$  as in Lemma 3;  
             **if**  $\sigma_i(Z)$  *is decreasing with*  $i$  **then**  
                  $S := S \cup \{\sigma(Z)\}$ ;  
             **end**  
         **end**  
     **end**  
**end**

**Algorithm 2:** Finding maximizing  $Z$  for the proximal operator (32)

Here the operator  $\mathcal{P}_i$  extracts elements corresponding to sub-block  $i$ . We do not explicitly penalize the rank of  $X$ , but instead accomplish this via the rank penalization of the sub-matrices.

To optimize (48) we use ADMM [15]. For each block  $\mathcal{P}_i(X)$  we introduce a separate set of variables  $X_i$  and enforce consistency via the linear constraints  $X_i - \mathcal{P}_i(X) = 0$ . We formulate an augmented Lagrangian of (48) as

$$\sum_{i=1}^K (\mathcal{R}_g(X_i) + \|X_i - \mathcal{P}_i(M)\|_F^2 + \rho \|X_i - \mathcal{P}_i(X) + \Lambda_i\|_F^2 - \rho \|\Lambda_i\|_F^2). \quad (49)$$

At each iteration  $t$  of ADMM we solve the subproblems

$$X_i^{t+1} = \underset{X_i}{\operatorname{argmin}} \mathcal{R}_g(X_i) + \|X_i - \mathcal{P}_i(M)\|_F^2 + \rho \|X_i - \mathcal{P}_i(X^t) + \Lambda_i^t\|_F^2, \quad (50)$$

for  $i = 1, \dots, K$  and

$$X^{t+1} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^K \rho \|X_i^{t+1} - \mathcal{P}_i(X) + \Lambda_i^t\|_F^2. \quad (51)$$

Here  $\Lambda_i^t$ ,  $i = 1, \dots, K$  are the scaled dual variables whose updates at iteration  $t$  are given by  $\Lambda_i^{t+1} = \Lambda_i^t + X_i^{t+1} - \mathcal{P}_i(X^{t+1})$ . The first problem (50) can be solved using the proximal operator derived in the previous section. The second subproblem (51) is a separable least squares problem with closed form solution.

### 3.1 Extending the Solution

To extend the solution beyond the blocks we employ a nullspace matching scheme which has previously been used in [16] and [17]. The goal is find a rank  $r$  factorization of the full solution  $X = UV^T$  given the solution on the blocks. Each

block  $\mathcal{P}_k(X)$  can be factorized as  $\mathcal{P}_k(X) = U_k V_k^T$ . Then  $\mathcal{P}_k(U)$ <sup>3</sup> must lie in the column space of  $U_k$  or equivalently it must be orthogonal to the complement, i.e.  $(U_k^\perp)^T \mathcal{P}_k(U) = 0$ . We can also write this as

$$A_k U = [0 \quad (U_k^\perp)^T \quad 0] U = 0. \quad (52)$$

Collecting these into matrix,  $AU = 0$ , we can find  $U$  by minimizing  $\|AU\|$ . Since the scale of  $U$  is arbitrary we can consider this as a homogeneous least squares problem which can be solved using SVD. For known  $U$  we can simply find  $V$  by minimizing  $\|W \odot (M - UV^T)\|$ .

## 4 Stronger Relaxations Using a Trust Region Formulation

In case of very large noise levels the regularizer  $\mathcal{R}_g$  may not be strong enough to enforce low rank of the solution. In this section we present an approach to strengthen it by restricting the algorithm to a local search close to a current solution estimate  $X_k$ . We consider minimization of

$$g(\text{rank}(X)) + \|X - X_0\|_F^2 + \lambda \|X - X_k\|_F^2 \quad (53)$$

The third term can be thought of as a restriction of the step-length of  $X$  to a region where our convex relaxation is accurate. By completing squares the expression above can be written

$$(1 + \lambda) \left( \frac{1}{1 + \lambda} g(\text{rank}(X)) + \left\| X - \frac{X_0 + \lambda X_k}{1 + \lambda} \right\|_F^2 + C \right), \quad (54)$$

where  $C$  is a constant that depends on  $\lambda$ ,  $X_0$  and  $X_k$ . Therefore we find that the convex envelope of (53) is

$$(1 + \lambda) \mathcal{R}_{\frac{g}{1+\lambda}}(X) + \|X - X_0\|_F^2 + \lambda \|X - X_k\|_F^2. \quad (55)$$

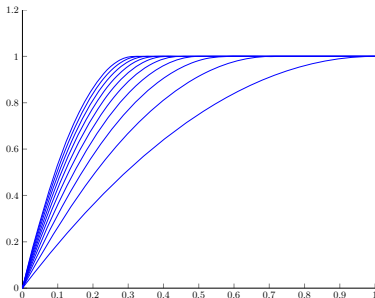
It can be shown that the term  $(1 + \lambda) \mathcal{R}_{\frac{g}{1+\lambda}}(X) \rightarrow g(\text{rank}(X))$  when  $\lambda \rightarrow \infty$ , that is, we have point wise convergence. Figure 3 shows a one-dimensional version of  $(1 + \lambda) \mathcal{R}_{\frac{g}{1+\lambda}}$  with  $g(k) = k$  for varying  $\lambda$ .

Our trust region approach consists of two steps. First we minimize (55) with respect to  $X$ . Then we update  $X_k$  and repeat the process. Note that at any fix point  $X = X_k$  we have a (possibly local) solution to

$$\min_X (1 + \lambda) \mathcal{R}_{\frac{g}{1+\lambda}}(X) + \|X - X_0\|_F^2. \quad (56)$$

In practice we make the  $X_k$  update at each step in the ADMM algorithm instead of running the ADMM until convergence before updating  $X_k$ . This greatly increases speed of convergence.

<sup>3</sup> Here  $\mathcal{P}_k(U)$  denotes the rows corresponding to block  $k$ .



**Fig. 3.** The regularizer  $r(\sigma) = 1 - [1 - \sqrt{1 + \lambda\sigma}]_+^2$  for different  $\lambda$

## 5 Implementation and Experiments

In the experiments we focus our attention to the function  $g(k) = \mu \max(r_0, k)$ . This choice allows us to penalize a rank higher than  $r_0$  while not being biased towards lower rank solutions.

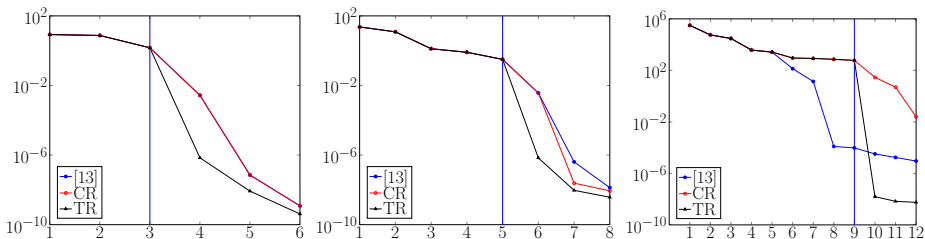
### 5.1 Comparison to [13]

We first compare the performance of the envelope of [13] and our convex relaxation (CR) in the block decomposition approach (48). We consider the same three image sequences (*book*, *hand* and *banner*) which was used in [13]. Since we are looking for fixed rank solutions we simply choose our weight  $\mu$  to be sufficiently large. This makes the approach essentially parameter free. In contrast [13] iterates over weights to find a correct rank solution. The difficulty of finding the optimal parameters is heavily depending on the amount of noise in the data. For problems with noisy data and many large blocks (such as the *banner* sequence) this may be computationally infeasible. We also compare to the trust region based iterative method (TR) described in section 4.

Figure 4 displays the singular values of a single block in the solutions for the three image sequences. Note the logarithmic scale. The methods perform very similarly for the *book* and *hand* sequence. This is due to these sequences having low levels of noise and the problem instance being small enough for it to be feasible to iteratively find a good  $\mu$ . The reconstruction error for the three sequences can be seen in Table 1.

**Table 1.** The errors  $\|W \odot (X - M)\|_F$  after extending the solution beyond the blocks as described in Section 3.1 (which ensures the correct rank)

	[13]	CR	TR
<i>book</i>	1.2731	1.2733	1.2678
<i>hand</i>	0.91386	0.9141	0.91508
<i>banner</i>	3950.2	3373.2	3373.2

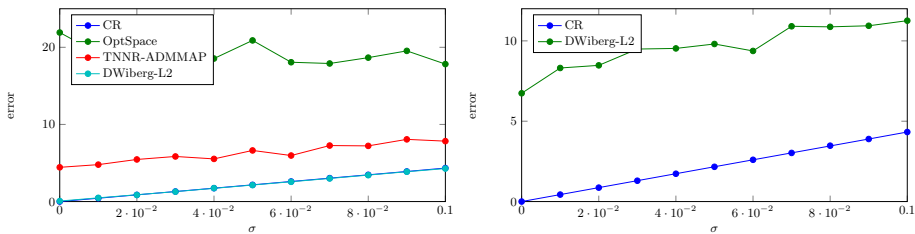


**Fig. 4.** Singular values for a single block in the *book*, *hand* and *banner* sequence. The vertical blue line indicates the desired rank.

## 5.2 Comparison to Non-convex Methods

Next we compare the performance of the proposed method to three state-of-the-art non-convex methods; OptSpace [18], Truncated Nuclear Norm Regularization [19] and Damped Wiberg-L2 [20].

The measurement matrix was chosen as  $M = UV^T + N$  where  $U, V \in \mathbb{R}^{100 \times 5}$ ,  $N \in \mathbb{R}^{100 \times 100}$  and  $U_{ij}, V_{ij} \sim \mathcal{N}(0, 1)$  and  $N_{ij} \sim \mathcal{N}(0, \sigma)$ . If  $\sigma$  is small then  $M$  will be approximately rank 5. The observation matrix  $W$  consisted of overlapping blocks along the diagonal and had 72% missing data. To the left in Figure 5 we can see the average of  $\|W \odot (X - M)\|_F$  over 100 instances. The performance of the proposed method and Damped Wiberg-L2 is very similar on this data. To illustrate the benefit of the proposed method we also performed an experiment on another family of instances generated by replacing the fifth column of  $V$  by  $10^3 \mathbf{1}$ . This essentially makes  $M$  have one very dominant singular value which is common in applications. The averaged result for these instances can be seen to the right in Figure 5.



**Fig. 5.** Comparison with non-convex methods. *Left:* Initial experiment. (Note that the errors for our approach and DWiberg-L2 are very similar). *Right:* Experiment with adjusted row-mean.

## References

1. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: IEEE Conference on Computer Vision and Pattern Recognition (2000)

2. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(5), 865–877 (2008)
3. Garg, R., Roussos, A., de Agapito, L.: Dense variational reconstruction of non-rigid surfaces from monocular video. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2013)
4. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. *Int. J. Comput. Vision* 72(3), 239–257 (2007)
5. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. *Int. J. Comput. Vision* 104(3), 286–314 (2013)
6. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. *Int. Journal on Computer Vision* 9(2), 137–154 (1992)
7. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. *Psychometrika* 1(3), 211–218 (1936)
8. Eriksson, A., Hengel, A.: Efficient computation of robust weighted low-rank matrix approximations using the  $L_1$  norm. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(9), 1681–1690 (2012)
9. Strelow, D.: General and nested Wiberg minimization. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2012)
10. Wiberg, T.: Computation of principal components when data are missing. In: *Proc. Symposium of Computational Statistics*, pp. 229–326 (1976)
11. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization* 20(4), 1956–1982 (2010)
12. Angst, R., Zach, C., Pollefeys, M.: The generalized trace-norm and its application to structure-from-motion problems. In: *International Conference on Computer Vision* (2011)
13. Larsson, V., Olsson, C., Bylow, E., Kahl, F.: Rank minimization with structured data patterns. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014, Part III. LNCS*, vol. 8691, pp. 250–265. Springer, Heidelberg (2014)
14. Rockafellar, R.T.: *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton (1970)
15. 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)
16. Olsen, S., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision* 31(2-3), 233–244 (2008)
17. Jacobs, D.: Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images. In: *IEEE Conference on Computer Vision and Pattern Recognition* (1997)
18. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. *IEEE Trans. Inf. Theory* 56(6), 2980–2998 (2010)
19. Hu, Y., Zhang, D., Ye, J., Li, X., He, X.: Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(9), 2117–2130 (2013)
20. Okatani, T., Yoshida, T., Deguchi, K.: Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In: *Proceedings of the International Conference on Computer Vision* (2011)



# Maximizing Flows with Message-Passing: Computing Spatially Continuous Min-Cuts

Egil Bae<sup>1</sup>, Xue-Cheng Tai<sup>3</sup>, and Jing Yuan<sup>2</sup>

<sup>1</sup> Department of Mathematics, University of California, Los Angeles, USA

`ebae@math.ucla.edu`

<sup>2</sup> Department of Medical Biophysics, Schulich Medical School, Western University,  
Canada

`cn.yuanjing@gmail.com`

<sup>3</sup> Department of Mathematics, University of Bergen, Norway

`tai@math.uib.no`

**Abstract.** In this work, we study the problems of computing spatially continuous cuts, which has many important applications of image processing and computer vision. We focus on the convex relaxed formulations and investigate the corresponding flow-maximization based dual formulations. We propose a series of novel continuous max-flow models based on evaluating different constraints of flow excess, where the classical pre-flow and pseudo-flow models over graphs are re-discovered in the continuous setting and re-interpreted in a new variational manner. We propose a new generalized proximal method, which is based on a specific entropic distance function, to compute the maximum flow. This leads to new algorithms exploring flow-maximization and message-passing simultaneously. We show the proposed algorithms are superior to state of art methods in terms of efficiency.

## 1 Introduction

Many problems in image processing and computer vision can be modeled and formulated by the theory of Markov Random Fields (MRF) over graphs, in terms of computing a maximum a posteriori probability (MAP) estimate, see [23] for reference. Graph-cuts and message-passing, e.g. [5,4,30,31,19] are two main categories of efficient algorithms for the combinatorial optimization problem. However, graph-based methods suffer from visible grid bias, and reducing such bias requires either adding more neighbors locally or considering high-order cliques, which inevitably leads to a more intensive computation and memory cost.

On the other hand, variational methods can be applied to solve the same class of optimization problems in the spatially continuous setting, while avoiding the metrication errors generated by combinatorial algorithms. In particular, convex relaxation methods [21,7,15,34,24,9,2,20] were recently developed by relaxing the discrete constraint to some convex set, which leads great advantages both in theory and numerics: the convex optimization theory is well-established, efficient and reliable solvers are available with provable convergence properties, and also

easy to handle large-scale computation and speed up by GPUs. In this regard, the proximal method plays the central element to build up a wide range of efficient first-order methods, see e.g. [11,10] for references.

## 1.1 Contributions

In this work, we propose a series of max-flow dual formulations, to compute minimum cuts in the continuous setting. In contrast to previous work on continuous max-flow [33,1], we formulate the flow excess constraints in different ways, which directly lead to new generalized proximal algorithms, where the Bregman divergence acts as the distance measurement for updating the labeling function. We propose primal-dual algorithmic schemes which combine both a flow-maximizing step and message-passing step in one unified numerical framework. This reveals close connections between the proposed flow-maximization methods and the classical methods, where 'cuts' over the graphs can be computed by maximizing flows or propagating messages. Finally, we compare the proposed algorithms with state-of-art continuous optimization methods: the Split-Bregman algorithm [15], the primal-dual algorithm [10] and the max-flow algorithm in [33] through experiments.

## 2 Revisit: Max-flow and Full-Flow Representation

Many discrete optimization problems in image processing and computer vision can be formulated as finding the minimum cut over appropriate graphs, as first observed by Greig et. al. [16]. The two most efficient combinatorial algorithms for computing the minimum cut solve the dual max-flow problem over the graph, and are called the Ford Fulkerson algorithm [13] and push-relabel algorithm [14]. More recently, continuous max-flow algorithms [33] have been proposed that are able to solve isotropic versions of the min-cut / max flow problem by convex optimization techniques. Both the continuous max-flow algorithm in [33] and the Ford Fulkerson algorithm solve a *full-flow representation* of the max-flow problem, in contrast to the pseudo-flow representation in the push-relabel algorithm and the algorithms in this paper.

### 2.1 Discrete Min-cut and Max-flow Models

A graph  $\mathcal{G}$  is a pair  $(\mathcal{V}, \mathcal{E})$  consisting of a vertex set  $\mathcal{V}$  and an edge set  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . We let  $C(v, w) \geq 0$  denote the cost / weight / capacity on edge  $(v, w)$  and use the convention  $C(v, w) = 0$  if there is no edge  $(v, w)$ . In the min-cut and max-flow problems, there are two special vertices in addition to  $\mathcal{V}$ , a source vertex  $s$  and a sink vertex  $t$ . The min-cut problem is to find a partition of  $\mathcal{V} \cup s \cup t$  into two sets  $V_s$  and  $V_t$ , such that  $s \in V_s$  and  $t \in V_t$  with smallest cost possible, i.e. to solve

$$\min_{V_s, V_t} \sum_{v \in V_s, w \in V_t} C(v, w), \quad \text{s.t. } s \in V_s, t \in V_t, \quad V_s \cup V_t = \mathcal{V}, \quad V_s \cap V_t = \emptyset \quad (1)$$

It is well known that the min-cut problem (1) is dual to the maximum flow problem over the same graph. We let  $p_s(v)$  denote the flow on the edge  $(s, v)$  and  $C_s(v)$  denote its capacity  $C(s, v)$ . Similarly,  $p_t(v)$  and  $C_t(v)$  are the flow and capacity on  $(v, t)$  and  $p(v, w)$  the flow on  $(v, w)$ . The maximum flow problem can be formulated as follows

$$\max_{p_s} \sum_{v \in \mathcal{V}} p_s(v) \quad (2)$$

$$\text{s.t. } |p(v, w)| \leq C(v, w) \quad p_s(v) \leq C_s(v) \quad p_t(v) \leq C_t(v) \quad \forall v, w \in V \quad (3)$$

$$\sum_{(w,v): w \in V} p((w, v)) - p_s(v) + p_t(v) = 0 \quad \forall v \in V \quad (4)$$

where the objective (2) is to push the maximum amount of flow from the source to the sink under flow capacity constraints (3). Additionally, the flow conservation constraint (4) should hold, which states that the total amount of incoming flow should be balanced by the amount of outgoing flow at each vertex.

The classical Ford-Fulkerson algorithm [13] solves the max-flow problem (2) by successively pushing flow from  $s$  to  $t$  along non-saturated paths, while maintaining the flow conservation constraint (4) each iteration. In this paper, we also call (2) subject to (3) and (4), the *full-flow representation* of max-flow.

## 2.2 Continuous Min-cut and Max-flow Models

In the spatially continuous setting, the min-cut problem (1), especially for image segmentation, can be similarly formulated in terms of finding the two segments  $S, \Omega \setminus S \subset \Omega$  such that

$$\min_S \int_S C_s(x) dx + \int_{\Omega \setminus S} C_t dx + \int_{\partial S} C(s) ds, \quad (5)$$

where  $C_s(x)$  and  $C_t(x)$  are pointwise costs for assigning any  $x$  to the foreground  $S$  and background  $\Omega \setminus S$  respectively. As proposed by [21,7], this problem can be solved globally and exactly by solving the *continuous min-cut* as follows

$$\min_{u(x) \in [0,1]} E(u) = \int_{\Omega} (1-u)C_s dx + \int_{\Omega} uC_t dx + \int_{\Omega} C(x) |\nabla u|_2 dx, \quad (6)$$

which results in a convex optimization problem. Further studies can be found in [22,15] etc.

**Continuous Max-flow: Full-Flow Representation.** An interesting study on the *continuous min-cut model* (6) was proposed in [32,33], which built up the duality connection between (6) to the so-called *continuous max-flow model*. It directly presents the analogue to the well-known duality between max-flow and min-cut [12] discussed above.

As the discrete graph configuration shown above, given the continuous image domain  $\Omega$  and two terminals, link the source  $s$  and the sink  $t$  to each pixel  $x \in \Omega$

respectively; define three flow fields around the pixel  $x$ :  $p_s(x) \in \mathbb{R}$  directed from the source  $s$  to  $x$ ,  $p_t(x) \in \mathbb{R}$  directed from  $x$  to the sink  $t$  and the spatial flow field  $p(x) \in \mathbb{R}^2$  around  $x$  within the image plain.

By the above spatially continuous setting, the *continuous max-flow model* tries to maximize the total flow passing from the source  $s$ :

$$\max_{p_s, p_t, p} \int_{\Omega} p_s dx \quad (7)$$

subject to the three flow capacity constraints:

$$p_s(x) \leq C_s(x), \quad p_t(x) \leq C_t(x), \quad |p(x)|_2 \leq C(x), \quad \forall x \in \Omega. \quad (8)$$

and the flow conservation condition:

$$p_t(x) - p_s(x) + \operatorname{div} p(x) = 0, \quad \forall x \in \Omega. \quad (9)$$

The authors [32,33] proved that the continuous max-flow model (7) is equivalent to the continuous min-cut problem (6) in terms of primal and dual, where the labeling function  $u(x)$  just works as a multiplier to the linear flow conservation condition (9). To see this, the equivalent primal-dual model

$$\min_u \max_{p_s, p_t, p} \int_{\Omega} p_s dx + \langle u, p_t - p_s + \operatorname{div} p \rangle, \quad (10)$$

subject to the flow capacity constraints (8) was considered. The flow conservation condition (9) played a central role in constructing the duality between the max-flow and min-cut models: (7) and (6).

We call (7) the *full-flow representation* of the continuous max-flow model in this paper. In the following sections, we will discuss the other two continuous max-flow models which are distinct from the full-flow representation model (7). We will see that different continuous max-flow models can be constructed through variants of flow preservation (9), while the full-flow representation model (7) just corresponds to the balance of in-flow and out-flow.

To compute a solution to (6) or (7), discretization of the domain  $\Omega$  is necessary. One fundamental difference to the discrete max-flow and min-cut models is the rotationally invariant 2-norm in (6) and (8), which corresponds to the Euclidean perimeter in (5). In this paper we assume a general discretized image domain and differential operators when deriving the duality theory, but we keep the continuous notation  $\nabla$ ,  $\operatorname{div}$ ,  $\int$  to ease readability. To derive rigorous existence proofs for infinite dimensional spaces is quite involved and out of the scope of this conference paper.

### 3 Continuous Max-flow Models Represented by Pre-flows and Pseudo-flows

In this section, we propose and study two other continuous max-flow models in terms of the representations of *pre-flows* and *pseudo-flows*. Both models are dual to the continuous min-cut model (6).

### 3.1 Continuous Max-flow: Pre-flow Representation

Now we partially optimize the max-flow model (7) by maximizing over the source flow  $p_s(x) \leq C_s(x)$ . By simple computation, we can prove that

**Proposition 1.** *The continuous max-flow model (7) is equivalent to the following flow-maximization problem:*

$$\max_{p_t, p} \int_{\Omega} p_t dx \quad (11)$$

$$\text{s.t. } C_s(x) - \operatorname{div} p(x) - p_t(x) \geq 0, \quad \forall x \in \Omega \quad (12)$$

$$p_t(x) \leq C_t(x), \quad |p(x)| \leq C(x), \quad \forall x \in \Omega. \quad (13)$$

*Proof.* We first observe that the max-flow model (7) can be equivalently formulated as

$$\max_{p_t, p} \int_{\Omega} p_t dx \quad (14)$$

$$\text{s.t. } p_s(x) + \operatorname{div} p(x) - p_t(x) = 0, \quad \forall x \in \Omega \quad (15)$$

$$p_s(x) \leq C_s(x), \quad p_t(x) \leq C_t(x), \quad |p(x)| \leq C(x), \quad \forall x \in \Omega. \quad (16)$$

This just comes from the fact that the total source flow  $\int p_s dx$  equals to the total sink flow  $\int p_t dx$ , due to the flow balance condition (9). Changing the positive direction of flows  $p_s$  and  $p_t$  in (7), we then have (14).

Therefore, by the same procedures as in [32], optimizing (14) over the constraint  $p_s(x) \leq C_s(x)$ , we see that (14) can be equivalently expressed as

$$\min_{u \geq 0} \max_{p_t, p} \int_{\Omega} p_t dx + \langle u, C_s + \operatorname{div} p - p_t \rangle \quad (17)$$

$$\text{s.t. } p_t(x) \leq C_t(x), \quad |p(x)| \leq C(x) \quad \forall x \in \Omega.$$

where  $u$  is a Lagrange multiplier for  $C_s + \operatorname{div} p - p_t \geq 0$ . Clearly, (17) is just the primal-dual formulation of (11). Hence, we have:

$$(7) \iff (14) \iff (17) \iff (11).$$

The equivalence between (7) and (11) is proved.

Obviously, (11) gives another continuous max-flow model which tries to maximize the total flow streaming out to the sink  $t$  while keeping the maximum source flow  $p_s(x) = C_s(x)$ . We see that the excess of flows at each pixel is no longer constrained to vanish, but to be non-negative (12), i.e. the flow conservation condition (9) is not kept.

Moreover, we will show that (11) results in a novel max-flow algorithm, in the continuous context, which has similar steps as the well-known *push-relabel* algorithm proposed in [14]. With this perspective, the constraint (12) recovers the *pre-flow* condition. We call (11) the *pre-flow representation* of the continuous max-flow model. In view of (17), we have that

**Proposition 2.** *The pre-flow based max-flow model (11) is dual to the continuous min-cut problem (6), and also equivalent to its primal-dual model*

$$\begin{aligned} \min_{u \geq 0} \max_{p_t, p} \int_{\Omega} p_t dx + \langle u, C_s + \operatorname{div} p - p_t \rangle \quad (18) \\ \text{s.t. } p_t(x) \leq C_t(x), |p(x)| \leq C(x) \quad \forall x \in \Omega. \end{aligned}$$

The proof follows by (17).

### 3.2 Continuous Max-flow: Pseudo-flow Representation

By maximizing the continuous max-flow model (7) over the flows  $p_s(x) \leq C_s(x)$  and  $p_t(x) \leq C_t(x)$  simultaneously, we have that

**Proposition 3.** *The continuous max-flow model (7) is equivalent to the following flow-maximization problem:*

$$\max_{|p(x)| \leq C(x)} \int_{\Omega} \min(0, C_t + \operatorname{div} p - C_s) dx, \quad (19)$$

The flow excess at each point  $(C_t + \operatorname{div} p - C_s)(x) \neq 0$  is neither balanced nor non-negative, i.e. the pseudo-flow condition. Problem (19) is also related to the dual formulation of multi-region partitions proposed in [2].

*Proof.* Following the same steps in [32], optimizing the continuous max-flow model (7) over  $p_s(x) \leq C_s(x)$  and  $p_t(x) \leq C_t(x)$  results in

$$\min_{u(x) \in [0,1]} \max_{|p(x)| \leq C(x)} \int_{\Omega} u (C_t + \operatorname{div} p - C_s) dx \quad (20)$$

The min and max operators are interchangeable, by the minimax theorem. Then, by minimizing the above functional over  $u(x) \in [0, 1]$  at each pixel  $x \in \Omega$ , we obtain the optimization problem (19).

The formulation (19) emphasizes: first, the flow excess at each pixel  $x$  is neither balanced nor non-negative (pre-flow condition); actually, the flow excess can be either positive or negative; second, the object is to find the spatial flow field  $p(x)$  which maximizes the total negative flow excess, i.e.  $(C_t + \operatorname{div} p - C_s)(x) \leq 0$ . Observe that we find the third equivalent max-flow model in terms of the *pseudo-flow* condition, proposed in [17]. In this regard, we call (19) the *pseudo-flow representation* of the continuous max-flow model. In the following sections, we propose a new algorithm associated to the pseudo-flow based max-flow model (19).

## 4 Entropic Proximal Max-flow Algorithms

In this section, we consider the generalized proximal method to solve the newly proposed continuous max-flow models: (11) and (19) which are dual to the continuous min-cut problem (6). We will see that such proximal method based on

the generalized entropic distance functions leads to the generalized augmented Lagrangian method [28,18], and builds up a class of novel continuous max-flow algorithms which explores flow-maximization joint with message-passing simultaneously.

We first introduce the entropic proximal method using the generalized Bregman distance as its mapping kernel. Then we build up the new entropic-proximal based algorithms to the proposed continuous max-flow models (11) and (19). We also discuss their essential links to the *push-relabel* and *pseudoflow* algorithms over graphs.

#### 4.1 Proximal Methods with Bregman Distance

Given the closed proper convex function  $f(x)$ , the proximal mapping of any point  $z$  is defined by [26]:

$$\text{prox}_f(z) = (I + \lambda \partial f)^{-1}(z) = \arg \min_x \left\{ \frac{1}{2\lambda} \|x - z\|^2 + f(x) \right\}. \quad (21)$$

Then the classical proximal method [8] to minimize the function  $f(x)$  can be formulated as computing a sequence of proximal mappings iteratively:

$$x^{k+1} = (I + \lambda \partial f)^{-1}(x^k) = \arg \min_x \left\{ \frac{1}{2\lambda_k} \|x - x^k\|^2 + f(x) \right\}. \quad (22)$$

Convergence properties of the proximal method was studied in [27]. Its close connections to the augmented Lagrangian method were demonstrated in [25,28] by computing the iterative proximal mappings of the dual sequence.

The proximal method is one of important elements to design most the efficient first-order primal-dual algorithms [10]. One of its interesting extensions is to incorporate the generalized Bregman distance or divergence functions  $D_g(x, y)$  [6] as the proximity measurement, which results in the entropic proximal method:

$$x^{k+1} = \arg \min_x \left\{ D_g(x, x^k) + f(x) \right\} \quad (23)$$

where

$$D_g(x, y) = g(x) - g(y) - \langle \partial g, x - y \rangle, \quad (24)$$

$g(x)$  is a differentiable and strictly convex function.

Clearly, the Bregman distance (24) provides a quite general conception on the proximity measurement: for example, the function  $g(x) = \frac{1}{2} \|\cdot\|^2$  just gives the common squared Euclidean distance  $\frac{1}{2} \|x - y\|^2$ ; the entropy function for the vector  $x := (x_1, \dots, x_n) \in (\mathbb{R}^+)^n$

$$g(x) = \sum_i (x_i \log x_i - x_i)$$

results in the generalized *Kullback-Leibler* divergence of two vectors  $x, y \in (\mathbb{R}^+)^n$  such that

$$D_g(x, y) = \sum_{i=1}^n \left( x_i \log(x_i/y_i) - x_i + y_i \right), \quad (25)$$

see also [3] for the definition of more Bregman distances.

**Generalized Augmented Lagrangian Method.** [28] showed the entropic proximal method (23) over the dual sequence just amounts to the *generalized augmented Lagrangian method*, which incorporates the classical augmented Lagrangian method as its special case with the quadratic Euclidean distance.

Now we consider the generalized optimization problem associated to the continuous max-flow models:

$$\min_{u \in C_u} \max_{p \in C_p} L(p, u) = f(p) + \langle u, G(p) \rangle \quad (26)$$

where  $C_u$  and  $C_p$  are the constraint sets on  $u$  and  $p$  respectively.

Let the dual function  $D(u)$  be

$$D(u) := \max_{p \in C_p} L(p, u).$$

As in [28], the entropic proximal method (23) to the dual function  $D(u)$  gives the generalized augmented Lagrangian method

$$u^{k+1} = \arg \min_{u \in C_u} \left\{ c D_g(u, u^k) + D(u) \right\}$$

where  $c$  is some positive constant. Therefore, we have the corresponding augmented Lagrangian function as follows:

$$L_c(p, v) = \min_{u \in C_u} \left\{ L(p, u) + c D_g(u, v) \right\}.$$

The generalized augmented Lagrangian method contains the following two steps at each iteration:

$$p^{k+1} = \arg \max_{p \in C_p} L_{c^k}(p, u^k), \quad (27)$$

$$u^{k+1} = \arg \min_{u \in C_u} \langle u, G(p^{k+1}) \rangle + c^k D_g(u, u^k). \quad (28)$$

It is important to notice that the function  $L_c(x, v)$  is the smoothed approximation to  $L(x, u)$ , hence better properties in numerics. In particular, when the quadratic  $L_2$ -norm is used as the distance function, then the classical augmented Lagrangian method is recovered.

In the following part, we propose and discuss a class of new continuous max-flow algorithms based on the entropic proximal method, especially the generalized augmented Lagrangian method. We will also show its close connections to the existing max-flow algorithms over graphs.

## 4.2 Entropic Proximal Max-flow Algorithm to (11)

For the pre-flow represented max-flow model (11), its corresponding primal-dual model (18) gives the common Lagrangian function:

$$\begin{aligned} \max_{p_t, p} \min_{u(x) \geq 0} L(p_t, p, u) &:= \int_{\Omega} p_t dx + \langle u, C_s + \operatorname{div} p - p_t \rangle \\ \text{s.t. } p_t(x) &\leq C_t(x), |p(x)| \leq C(x) \quad \forall x \in \Omega. \end{aligned} \quad (29)$$



In this work, we consider the Kullback-Leibler distance (25) as the proximal function, i.e.

$$D(u, v) = \int_{\Omega} \left\{ u(x) \log(u(x)/v(x)) - u(x) + v(x) \right\} dx.$$

This results in the augmented Lagrangian function

$$L_c(p_t, p, v) = \min_{u(x) \geq 0} L(p_t, p, u) + cD(u, v) \quad (30)$$

From the first order optimality condition, we obtain the explicit expression for the minimizer  $u = v \exp[-(C_s + \operatorname{div} p - p_t)/c]$  for  $v \geq 0$ , which leads to

$$L_c(p_t, p, v) = \int_{\Omega} \left\{ p_t - c \left[ v \exp \left\{ -\frac{C_s + \operatorname{div} p - p_t}{c} \right\} + 1 \right] \right\} dx, \quad v \geq 0. \quad (31)$$

In view of the step (27) of the generalized augmented Lagrangian method, the augmented Lagrangian function (30) can then be expressed, in terms of  $u^k$  at each iteration, as:

$$L_c(p_t, p, u^k) = \int_{\Omega} \left\{ p_t - c \left[ u^k \exp \left\{ -\frac{C_s + \operatorname{div} p - p_t}{c} \right\} + 1 \right] \right\} dx, \quad u^k \geq 0.$$

By means of (30), we have the new continuous max-flow algorithm corresponding to the pre-flow model (11):

**Algorithm 4.** Initialize  $u^0(x) \in (0, 1) \forall x \in \Omega$ ,  $p_t^0, p^0$ . For  $k=0, 1, \dots$  until convergence, perform the following two steps (flow maximization and message passing):

– Maximize over the flows  $p_t$  and  $p$  by

$$p_t^{k+1} := \arg \max_{p_t(x) \leq C_t(x)} L_c(p_t, p^k, u^k); \quad (32)$$

$$p^{k+1} := \arg \max_{|p(x)| \leq C(x)} L_c(p_t^k, p, u^k); \quad (33)$$

where the step (32) can be solved explicitly through simple variational computation and the step (33) can be solved iteratively, as shown below.

– Update the message function  $u$  by

$$u^{k+1} := u^k \exp \left\{ -\frac{C_s + \operatorname{div} p^{k+1} - p_t^{k+1}}{c} \right\} \quad (34)$$

For the flow-maximization step (32), it's easy to solve the given maximization problem explicitly by

$$p_t^{k+1} = \min \left\{ C_s + \operatorname{div} p^k - c \log u^k, C_t \right\}, \quad (35)$$

since  $\frac{\partial L_c(p_t, p^k, u^k)}{\partial p_t} = 0$  if  $p_t = C_s + \operatorname{div} p^k - c \log u^k$ .

For the flow-maximization step (33), we apply one iteration of the projected-gradient method

$$p^{k+1} = \operatorname{Proj}_{|\cdot| \leq C(x)} \left\{ p^k + \gamma \nabla \left( u^k \exp \left\{ - \frac{C_s + \operatorname{div} p^k - p_t^k}{c} \right\} \right) \right\}, \quad (36)$$

where  $\gamma > 0$  is the step-size.

### 4.3 Entropic Proximal Max-flow Algorithm to (19)

Likewise, for the pseudo-flow represented max-flow model (19), its corresponding primal-dual formulation (20) expresses the common Lagrangian function

$$\min_{u(x) \in [0,1]} \max_{|p(x)| \leq C(x)} L(p, u) := \int_{\Omega} u (C_t + \operatorname{div} p - C_s) dx.$$

Consider the function  $u(x) \in [0, 1]$ , we apply the following Bregman distance as the proximal function

$$D(u, v) = \int_{\Omega} \left\{ u \log \left( \frac{u}{v} \right) + (1 - u) \log \left( \frac{1 - u}{1 - v} \right) \right\} dx.$$

The resulting augmented Lagrangian function is

$$L_c(p, v) = \min_{u(x) \in [0,1]} L(p, u) + c D(u, v) \quad (37)$$

$$= -c \int_{\Omega} \log \left\{ (1 - v) + v \exp \left( - \frac{C_t + \operatorname{div} p - C_s}{c} \right) \right\} dx, \quad (38)$$

where  $c > 0$  works as the step-size.

Considering  $\exp(0/c) = 1$ , it is easy to see that

$$L_c(p, v) = -c \int_{\Omega} \log \left\{ (1 - v) \exp \left( \frac{0}{c} \right) + v \exp \left( - \frac{C_t + \operatorname{div} p - C_s}{c} \right) \right\} dx.$$

As  $c \rightarrow 0^+$ , we have the limit function [29]

$$\lim_{c \rightarrow 0^+} L_c(p, v) = \int_{\Omega} c \min(0, C_t + \operatorname{div} p - C_s) dx$$

which is just the original pseudo-flow represented max-flow model (19). To this end, we see that the augmented Lagrangian function (37) just works as the smoothed version of the energy function (19).

Following the step (27) of the generalized augmented Lagrangian method, the augmented Lagrangian function (37) can then be expressed, in terms of  $u^k$  at each iteration, as:

$$L_c(p, u^k) = -c \int_{\Omega} \log \left\{ (1 - u^k) + u^k \exp \left( - \frac{C_t + \operatorname{div} p - C_s}{c} \right) \right\} dx.$$

By means of (30), we have the new continuous max-flow algorithm to its pseudo-flow represented model (19):

**Algorithm 5.** Initialize  $u^0(x) \in (0, 1) \forall x \in \Omega$ ,  $p_t^0, p^0$ . For  $k=0, 1, \dots$  until convergence, perform the following two steps (flow maximization and message passing):

- Maximize over the flows  $p$  by

$$p^{k+1} := \arg \max_{|p(x)| \leq C(x)} L_c(p, u^k); \quad (39)$$

which can be solved approximately by one iteration of projected gradient.

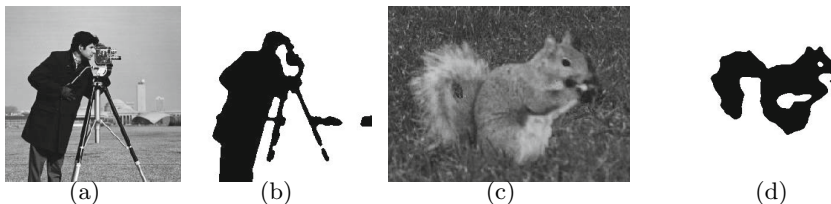
- Update the message function  $u$  by

$$u^{k+1} := \frac{u^k \exp(-G^{k+1}/c)}{1 - u^k + u^k \exp(-G^{k+1}/c)}, \quad (40)$$

where for  $\forall x \in \Omega$

$$G^{k+1}(x) = (C_t + \operatorname{div} p^{k+1} - C_s)(x).$$

Algorithm 5 is similar to the smoothing dual algorithm proposed in [2] for multiphase partition problems. One crucial difference is that algorithm 5 solves the problem exactly without any smoothing approximation.



**Fig. 1.** Segmentation with data term (41): (b) result on image (a) with  $C(x) = \alpha = 0.5$ ,  $c_1 = 0.15$  and  $c_2 = 0.6$ ; (d) result on image (c) with  $C(x) = \alpha = 0.25$ ,  $c_1 = 0.16$  and  $c_2 = 0.5$

## 5 Experiments

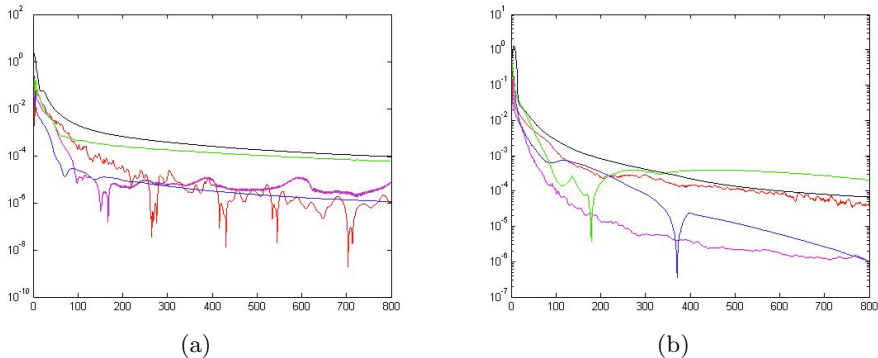
This section validates the convergence of the algorithms 4 and 5 on some image segmentation examples and comparisons are given to the previous max-flow algorithm [33], the Split-Bregman algorithm [15] and the primal-dual algorithm [10]. They are regarded as the state of the art algorithms for solving the convex partition problem. We choose the fidelity term

$$C_s(x) = |I(x) - c_1|^2, \quad C_t(x) = |I(x) - c_2|^2, \quad (41)$$

where  $I$  is the input image and  $c_1$  and  $c_2$  are two scalar gray values approximating the mean image intensities within each region. Results are shown in Figure 1. Figure 2 shows plots of the relative energy error

$$\frac{|E(u^k) - E(u^*)|}{E(u^*)}$$

where  $E$  is the energy (6),  $u^k$  is the solution at iteration  $k$  and  $u^*$  is the ground truth solution computed by 100000 iterations for each method. It can be observed that the two new variants of the max-flow algorithm converge at a similar rate as the old max-flow algorithm on example figure 1 (a), while on figure 1 (c) algorithm 5 is faster and algorithm 4 is slower. In both images all the max-flow algorithms converge considerably faster than the Split-Bregman and primal-dual algorithm. We speculate the reason for the faster convergence is that the max-flow algorithms avoid the projection step for incorporating the constraint  $u(x) \in [0, 1], \forall x \in \Omega$ . The CPU times are



**Fig. 2.** Convergence of relative energy error  $\frac{|E(u^k) - E(u^*)|}{E(u^*)}$  for iterations  $k = 1, \dots, 800$ : (a) image 1(a); (b) image 1(c). The function  $u^*$  is the ground truth solutions computed by 100000 iterations of each method. Red is the new max-flow algorithm 5, magenta is new max-flow algorithm 4, blue is the old max-flow algorithm [33], green is Split-Bregman [15] and black is the primal-dual algorithm [10].

## 6 Conclusions

In this paper, we propose a series of novel flow-maximization models dual to the continuous min-cut problem by formulating the flow excess conditions in different ways. In theory, the proposed dual formulations discover and re-interpret the conventional pre-flow and pseudo-flow models over discrete graphs in the spatially continuous setting under a new variational perspective. In addition, the new dual formulations, i.e. the continuous max-flow models, directly lead to new generalized proximal dual optimization based algorithms, which embed

both flow maximization and message-passing in a single algorithmic framework. Moreover, we show the proposed algorithms numerically outperform the state-of-art methods by experiments.

**Acknowledgements.** This research has been supported by the Norwegian Research Council eVita project 214889 and eVita project 166075 and Natural Sciences and Engineering Research Council of Canada (NSERC) Accelerator Grant R3584A04.

## References

1. Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on PAMI* 28, 106–118 (2006)
2. Bae, E., Yuan, J., Tai, X.-C.: Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision* 92(1), 112–129 (2011)
3. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with bregman divergences. *Journal of Machine Learning Research* 6, 1705–1749 (2005)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on PAMI* 26, 359–374 (2001)
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on PAMI* 23, 1222 (2001)
6. 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 Computational Mathematics and Mathematical Physics* 7, 200–217 (1967)
7. Bresson, X., Esedoglu, S., Vanderghenst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision* 28(2), 151–167 (2007)
8. Censor, Y.A., Zenios, S.A.: *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press (1997)
9. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Technical Report TR-2008-05, University of Bonn (November 2008)
10. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40(1), 120–145 (2011)
11. 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 Sciences* 3(4), 1015–1046 (2010)
12. Ford, L.R., Fulkerson, D.R.: *Flows in Networks*. Princeton University Press, Princeton (1962)
13. Ford Jr., L.R., Fulkerson, D.R.: Maximal flow through a network. *Canad. J. Math.* 8, 399–404 (1956)
14. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum-flow problem. *J. ACM* 35(4), 921–940 (1988)
15. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split bregman method: Segmentation and surface reconstruction. *J. Sci. Comput.* 45(1-3), 272–293 (2010)

16. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. *J. Royal Stat. Soc., Series B*, 271–279 (1989)
17. Hochbaum, D.S.: The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research* 56(4), 992–1009 (2008)
18. Iusem, A.N., Svaiter, B.F., Teboulle, M.: Entropy-like proximal methods in convex programming. *Mathematics of Operations Research* 19(4), 790–814 (1994)
19. Kolmogorov, V., Wainwright, M.J.: On the optimality of tree-reweighted max-product message-passing. In: *UAI*, pp. 316–323 (2005)
20. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained total variation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) *SSVM 2009*. LNCS, vol. 5567, pp. 150–162. Springer, Heidelberg (2009)
21. Nikolova, M., Esedoglu, S., Chan, T.F.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. App. Math.* 66(5), 1632–1648 (2006)
22. Olsson, C., Byröd, M., Overgaard, N.C., Kahl, F.: Extending continuous cuts: Anisotropic metrics and expansion moves. In: *ICCV*, pp. 405–412 (2009)
23. Paragios, N., Chen, Y., Faugeras, O.: *Handbook of Mathematical Models in Computer Vision*. Springer-Verlag New York, Inc., Secaucus (2005)
24. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 792–805. Springer, Heidelberg (2008)
25. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* 1(2), 97–116 (1976)
26. Rockafellar, R.T.: *Convex analysis*. Princeton Mathematical Series, vol. 28. Princeton University Press, Princeton (1970)
27. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization* 14(5), 877–898 (1976)
28. Teboulle, M.: Entropic proximal mappings with applications to nonlinear programming. *Math. Oper. Res.* 17(3), 670–690 (1992)
29. Teboulle, M.: A unified continuous optimization framework for center-based clustering methods. *J. Mach. Learn. Res.* 8, 65–102 (2007)
30. Wainwright, M., Jaakkola, T., Willsky, A.: Map estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. *IEEE Transactions on Information Theory* 51, 3697–3717 (2002)
31. Wainwright, M.J., Jaakkola, T., Willsky, A.S.: Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory* 51(11), 3697–3717 (2005)
32. Yuan, J., Bae, E., Tai, X.C.: A study on continuous max-flow and min-cut approaches. In: *CVPR, USA, San Francisco* (2010)
33. 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, 559–587 (2013)
34. Zach, C., Gallup, D., Frahm, J.-M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: *VMV 2008* (2008)

# A Compact Linear Programming Relaxation for Binary Sub-modular MRF

Junyan Wang and Sai-Kit Yeung

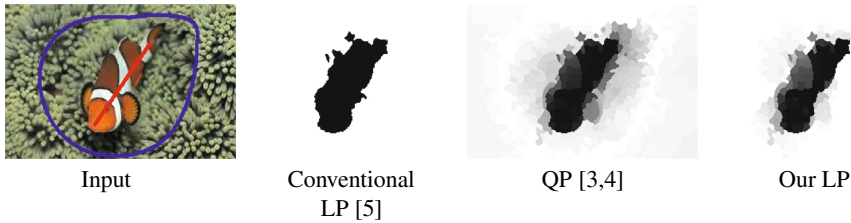
Singapore University of Technology and Design  
{junyan\_wang, saikit}@sutd.edu.sg

**Abstract.** Direct linear programming (LP) solution to binary sub-modular MRF energy has recently been promoted because i) the solution is identical to the solution by graph cuts, ii) LP is naturally parallelizable and iii) it is flexible in incorporation of constraints. Nevertheless, the conventional LP relaxation for MRF incurs a large number of auxiliary variables and constraints, resulting in expensive consumption in memory and computation. In this work, we propose to approximate the solution of the conventional LP at a significantly smaller complexity by solving a novel compact LP model. We further establish a tightenable approximation bound between our LP model and the conventional LP model. Our LP model is obtained by linearizing a novel  $l_1$ -norm energy derived from the Cholesky factorization of the quadratic form of the MRF energy, and it contains significantly fewer variables and constraints compared to the conventional LP relaxation. We also show that our model is closely related to the total-variation minimization problem, and it can therefore preserve the discontinuities in the labels. The latter property is very desirable in most of the imaging and vision applications. In the experiments, our method achieves similarly satisfactory results compared to the conventional LP, yet it requires significantly smaller computation cost.

## 1 Introduction

Markov Random Field (MRF) has become one of the most popular models for fundamental computer vision tasks. In an MRF model, an MRF energy is minimized in order to find an optimal solution to the task. Minimizing general MRF energies is NP-hard [1], while certain types of the MRF energies can be minimized efficiently and exactly by using, for example, graph cuts [2].

**Conventional LP Relaxation.** Recently, Bhusnurmath and Taylor [6] promoted the direct continuous linear programming (LP) solution to the binary sub-modular MRF. The LP model was obtained by linearizing the  $l_1$ -norm pairwise potential in the binary sub-modular MRF using auxiliary variables. Bhusnurmath and Taylor proved that the solution to the continuous LP model is identical to the graph-cuts solution given the same binary MRF energy. Their work was motivated by the fact that LP algorithms, e.g. the interior point method, can be easily parallelized. This is natural, since the interior point method is based on elementary matrix operations. The GPU version of all common matrix operations can easily be found in many toolboxes, such as MATLAB and



**Fig. 1.** Conventional LP is computationally demanding but it preserves discontinuity of the labels at the true boundary. QP [3,4] has much lower computational complexity but often produces over-smooth labels at the boundary. Our method provides a solution sharper at object boundary at an affordable computational cost.

CULA<sup>1</sup>. On the contrary, the parallel implementation of graph cuts is very challenging, on which consensus has not yet been reached [7,8]. Furthermore, incorporating linear constraints into an LP model is straightforward, while this is not the case for graph cuts. Lempitsky et al. [9] also showed linear constraints can be useful to segmentation.

**Motivations.** As reported in [5,6], the conventional LP relaxation contains a large number of auxiliary variables and constraints, which would cause large consumption in memory and computation. Consequently, the computation upon shared with multiple computing units may still remain expensive.

In contrast to the LP model, the computational complexity of the quadratic programming (QP) relaxation for the binary sub-modular MRF energy proposed in [3,4] is much smaller than that required by the conventional LP model. This is largely because no auxiliary variables or constraints are required in the model. However, the QP model may produce over-smooth ambiguous labels at the desired discontinuities in the solution. For instance in object segmentation in images, this may cause incorrect segmentation. As shown in Fig. 1, the solution by conventional LP is clean and more desirable than that from QP.

**Our Contributions.** To gain high quality solution similar to that from LP, at a computational cost similar to that of QP models, we propose a novel LP relaxation for binary sub-modular MRF to leverage both the compactness of the QP relaxation and the edge preservability of LP relaxation. Our LP relaxation is obtained by linearizing a novel  $l_1$ -norm minimization problem that is derived from the Cholesky factorization of the QP relaxation model. We further establish a tightenable approximation bound between our LP relaxation and the conventional LP relaxation. The complexity of the resultant algorithm for solving the proposed LP problem is of the same order of the corresponding QP model, and it is significantly smaller than that of the conventional LP. In addition, the derived novel  $l_1$ -norm minimization is strongly related to the total-variation minimization problem according to our theoretical analysis. Thus, it is able to preserve discontinuities in labels.

<sup>1</sup> <http://www.culatools.com/>



## 2 Background

### 2.1 The Binary Submodular MRF Model in Computer Vision

In the generic MRF model for the labeling problems in computer vision, the labels in the image are formulated as an Markov random field, and the corresponding distribution is in the form of Gibbs distribution according to the HammersleyClifford theorem. The labeling task is therefore cast into an Maximum *a posterior* (MAP) problem. Due to the Gibbs distribution form, the MAP problem becomes an energy minimization problem, and the energy is often written in the following standard form:

$$E(\mathbf{x}) = \sum_{p \in \mathcal{P}} D_p(x_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(x_p, x_q),$$

where  $\mathbf{x}$  is a label vector corresponding to all elements in the image,  $D_p(\cdot)$  is known as the unary term, or data fidelity term, and  $V_{pq}(\cdot, \cdot)$  is a pairwise potential.

Due to the fundamental works by Boykov, Olga and Zabih [2] and Komogorov and Zabih [10], it is well-known that the above energy, especially for binary label, can be solved exactly by graph cuts, as long as  $E$  is submodular. One of the most successful applications of this formulation is object segmentation [11].

More recently, approximate solution to general MRF models attracts much attention from the energy minimization community [12,13]. We argue that a more generalizable approach for solving the binary submodular problem can make approximations to general problems easier.

### 2.2 Conventional LP Relaxation for Binary Submodular MRF

In the binary submodular MRF energy, the unary term is often formulated as a term linear in the label vector. The complexity of the optimization for the MRF model only lies in the pairwise potential. The pairwise potential can be written as:

$$V_{pq}(x_p, x_q) = w_{pq}|x_p - x_q|^o, \quad (1)$$

where  $p$  and  $q$  are the indices of image elements,  $\mathcal{E}$  is a neighborhood system and  $o$  is either 1 or 2 in this paper. We will elaborate on the choice of the value of  $o$  in this paper. In the context of segmentation,  $w_{pq}$  can be defined as  $w_{pq} = \frac{1}{1 + \{\|I_p - I_q\|^2\}} + c$ , where  $I_p, I_q$  are the image values at the  $p$ - and  $q$ -th pixel/superpixel in the image. The first component in  $w_{pq}$  encourages discontinuous labeling at image edges, and the constant  $c$  that imposes smoothness to the resultant boundary. The constant weight in the latter part is related to the curve-shortening flow in the active contour models [14,15].

It has been pointed out that when  $o = 1$ , the minimization of the binary submodular MRF energy with the above pairwise potential term in Eq. (1) is equivalent to an  $l_1$ -norm minimization problem in [6].

Formally, we may rewrite the total pairwise potential as

$$\sum_{\{p,q\} \in \mathcal{E}} w_{pq}|x_p - x_q| = \sum_{i,j} w_{ij}^e |x_i - x_j| = \|\text{diag}(\mathbf{w}^e)\mathbf{D}\mathbf{x}\|_{l_1}, \quad (2)$$

where  $w_{ij}^e = w_{pq}$  if  $i = p, j = q$ , and  $w_{ij}^e = 0$  if  $\{i, j\} \notin \mathcal{E}$ ,  $[\text{diag}(\mathbf{w}^e)]^{N^2 \times N^2}$  is the diagonal matrix composed of  $\mathbf{w}^e$ , where  $\mathbf{w}^e$  is the vectorized  $\{w_{ij}^e\}$ .  $\mathbf{D}$  is an incidence matrix defined as follow:

$$[D]_{ij}^{N^2 \times N} = \begin{cases} 1, & \text{if } j = (i \bmod N) \\ -1, & \text{if } (i \bmod N, j) \in \mathcal{E} \end{cases} \quad (3)$$

The LP model of the full MRF energy can be rewritten as follows:

$$\begin{aligned} \min_x \quad & \mathbf{v}^T \mathbf{x} + \mathbf{1}^T \mathbf{y} \\ \text{s.t.} \quad & -\mathbf{y} \leq \text{diag}(\mathbf{w}^e) \mathbf{D} \mathbf{x} \leq \mathbf{y} \\ & 0 \leq \mathbf{x} \leq 1, 0 \leq \mathbf{y}. \end{aligned} \quad (4)$$

where  $\mathbf{v}$  is the weights in unary term, and the variable  $x_{pq}$  is an auxiliary variable induced by the linearization process. It is further shown in [5] and [6] that the  $l_1$ -norm minimization problem can be solved by LP, and it is proven in [6] that the solution to the LP problem in [6] converges to either 0 or 1 without any external prodding.

A drawback of this LP formulation is that it requires a large number of auxiliary variables and constraints. Suppose that there are  $N$  elements to be labeled, then there can be *as many as*  $N + N \times N$  variables and  $N + 2N \times N$  linear constraints, which is the worst case. The computational complexity of LP is known as  $O(n^3)$  [16] where  $n$  is the number of variables, and when  $n$  is fixed the complexity is  $O(m)$  [17] where  $m$  is the number of constraints. As a result, the computational complexity for solving the above LP problem is  $O(N^6)$ , and the computational cost can be high, which has been witnessed in [5].

### 2.3 Comparing $l_1$ -Norm Minimization with $l_2$ -Norm Minimization

Two decades ago, it was observed that the minimization of square of image gradients will result in blurry edges. This leads to the invention of the celebrated ROF total-variation minimization model for denoising [18]. It has already been pointed out that the  $l_1$ -norm minimization in our context corresponds to total variation minimization [19]. Likewise, the  $l_2$ -norm minimization corresponds to the problem of minimization of square of gradients in the context of denoising.

In segmentation, the solution from  $l_2$ -norm minimization may also become over-smooth and therefore ambiguous at the boundaries. This can affect the accuracy of boundary locating in the segmentation, as shown in Fig. 1. Accordingly, we also expect the solution of our model to contain sharp discontinuities, and the  $l_1$ -norm minimization seems promising.

## 3 A Compact LP Relaxation for Binary Submodular MRF

### 3.1 Deriving a Compact LP Relaxation via Cholesky Factorization of $l_2$ -Norm

Since the conventional  $l_1$ -norm minimization is computationally expensive, we propose to seek alternatives to it. In the following, we will show that a new  $l_1$ -norm, which is

induced by factorizing the  $l_2$ -norm form of the boundary term in Eq.(1), can lead to a more compact LP problem with significantly less computational complexity compared to the original LP problem.

First, we rewrite the  $l_2$ -norm in quadratic form:

$$\sum_{i,j} w_{ij}^e \cdot 2(\mathbf{x}_i - \mathbf{x}_j)^2 = \mathbf{x}^T \widetilde{\mathbf{W}} \mathbf{x} \quad (5)$$

where  $\widetilde{\mathbf{W}} = \text{diag}(\widehat{\mathbf{w}}) + \text{diag}(\widehat{\mathbf{w}}) - 2\mathbf{W}$ ,  $\widehat{w}_i = \sum_j w_{ij}^e$ ,  $\widehat{w}_j = \sum_i w_{ij}^e$  and  $\mathbf{W} = [w_{ij}^e]$ . The full derivation of the above is included in the Appendix.

A quadratic continuous optimization problem is NP-hard if the matrix in the quadratic term is non-definite, i.e. the optimization is non-convex. In fact, having even single negative eigenvalue leads to NP-hard problem [20]. Regarding the convexity of the formulation, we have the following proposition.

**Proposition 1.** *The matrix  $\widetilde{\mathbf{W}}$  in Eq.(5) is positive semi-definite.*

The proof is included in the Appendix. Since  $\widetilde{\mathbf{W}}$  is positive semi-definite, the formulation is convex. It is also possible to ensure the matrix to be positive definite by adding a small positive value to the diagonals. In addition to the well-posedness of this formulation, we show that positive definiteness of the matrix  $\widetilde{\mathbf{W}}$  allows the problem to be linearized.

Our linear relaxation is based on the following facts:

$$\mathbf{x}^T \widetilde{\mathbf{W}} \mathbf{x} = \mathbf{x}^T \mathbf{U}^T \mathbf{U} \mathbf{x} = \|\mathbf{U} \mathbf{x}\|_{l_2}^2,$$

where  $\mathbf{U}$  is an upper triangular matrix of the same dimension of  $\widetilde{\mathbf{W}}$  and  $\widetilde{\mathbf{W}} = \mathbf{U}^T \mathbf{U}$  is known as the Cholesky factorization/decomposition. The squared matrix  $\mathbf{U}$  is *unique* for symmetric positive definite matrix  $\widetilde{\mathbf{W}}$ . The Cholesky factorization of it generally uses  $n^3/3$  FLOPs, where  $n$  is the rank of the matrix, and it is instantaneous for very large matrix on modern processors.

We observe that the matrix  $[\text{diag}(\mathbf{w}^e)\mathbf{D}]$  operating on  $\mathbf{x}$  in the conventional  $l_1$ -norm can also be thought of as being factorized from the matrix  $\widetilde{\mathbf{W}}$ . To see this, we can rewrite Eq. (1) as follows:

$$\|\text{diag}(\mathbf{w}^e)\mathbf{D}\mathbf{x}\|_{l_2}^2 = \mathbf{x}^T [\text{diag}(\mathbf{w}^e)\mathbf{D}]^T [\text{diag}(\mathbf{w}^e)\mathbf{D}] \mathbf{x} = \mathbf{x}^T \widetilde{\mathbf{W}} \mathbf{x}.$$

This motivates us to have the following new reformulation of the pairwise potential as:

$$E_{l_1^+}^2(\mathbf{x}) = \|\mathbf{U} \mathbf{x}\|_{l_1} \quad (6)$$

Here, we call the above norm to be minimized as the Cholesky  $l_1$ -norm.

A major difference between the conventional  $l_1$ -norm and our Cholesky  $l_1$ -norm is that the linear operator  $\mathbf{U}$  has much smaller dimension than  $[\text{diag}(\mathbf{w})\mathbf{D}]$ , giving rise to a LP relaxation with significantly fewer variables and constraints.

$$\begin{aligned} \min_{\mathbf{x}, \delta^+} \mathbf{v}^T \mathbf{x} + \mathbf{1}^T \delta^+ \\ \text{s. t. : } -\delta^+ \preceq \mathbf{U} \mathbf{x} \preceq \delta^+ \\ 0 \leq \mathbf{x}_i \leq 1, \delta_i^+ \geq 0, \end{aligned} \quad (7)$$

where the first term is the same as in Eq. (4) and  $\delta^+$  is an additional vector of auxiliary variables used for the linear relaxation and its dimension is  $N$ , as the same as  $\mathbf{x}$ . Essentially, Eq.(7) tries to reduce the bounding values of  $\mathbf{U}\mathbf{x}$ . The above LP is obtained by applying the equivalence between  $l_1$ -norm minimization and linear programming.

Compared with the conventional LP model in Eq.(4), our model in Eq.(7) has a significantly smaller number of variables and constraints. Specifically, for the image containing  $N$  superpixels, there are  $N + N \times N$  variables and  $N + 2N \times N$  linear constraints for the worst case in the original model [6,5], whereas there are only  $2N$  variables and  $2N$  linear constraints in our model. The complexity of our model is therefore  $O(N^3)$  which is the same as QP according to Eq.(5). The number of variables and constraints does not change when increasing the number of edges in the graph. We will compare the performance of the two formulations experimentally. The matrices were all set to sparse mode in the implementation.

### 3.2 Mathematical Relationship between $l_1$ -Norm and Cholesky $l_1$ -Norm

In this subsection, we are particularly interested in how tightly the proposed Cholesky  $l_1$ -norm can be related to the conventional  $l_1$ -norm energy, and we are interested in the relationship between the Cholesky  $l_1^+$ -norm and total variation.

Let us consider the reduced QR factorization of the rectangular matrix  $[\text{diag}(\mathbf{w})\mathbf{D}]$  in the  $l_1$ -norm boundary term, i.e.  $[\text{diag}(\mathbf{w})\mathbf{D}] = \mathbf{Q}^{N^2 \times N} \mathbf{R}^{N \times N}$ , where  $\mathbf{Q}$  is an orthogonal matrix, such that  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}^{N \times N}$ , and  $\mathbf{R}$  is an upper triangular matrix. The following fact will relate our Cholesky  $l_1$  relaxation to the original  $l_1$ -norm minimization.

**Theorem 1.** *The upper triangular matrix  $\mathbf{U}$  in the Cholesky  $l_1$ -norm minimization model in Eq.(6) is identical to the upper triangular matrix  $\mathbf{R}$  in the QR factorization of  $[\text{diag}(\mathbf{w})\mathbf{D}]$  in the  $l_1$ -norm minimization model in Eq.(2)*

The proof of this theorem is presented in the Appendix. This theorem implies several additional relationships between the  $l_1$ -norm and the Cholesky  $l_1$ -norm.

**Corollary 1.**  $\mathbf{U}\mathbf{x} = \mathbf{Q}^T \mathbf{Q} \mathbf{U} \mathbf{x} = \mathbf{Q}^T [\text{diag}(\mathbf{w})\mathbf{D}\mathbf{x}]$ .

The above equality implies that the Cholesky  $l_1$ -norm is the  $l_1$ -norm of the linearly transformed weighted gradients, and the transformation matrix is  $\mathbf{Q}$ . The weighted variations in  $\mathbf{x}$  are projected on the subspace of  $\mathbf{Q}$  before calculating the total. Hence, we may also view our Cholesky  $l_1$ -norm as a total subspace-variation. This observation implies that the quasi-total variation minimization may share the discontinuity preservability of the total variation minimization.

Besides, Theorem 1 offers us a stronger relationship between the two formulations in terms of a tight equivalence-of-norm bound.

**Theorem 2.** *The difference between Cholesky  $l_1$ -norm and  $l_1$ -norm satisfies the following inequalities:*

$$(\|\text{diag}(\mathbf{w})\mathbf{D}\mathbf{x}\| / \|\mathbf{Q}\|) \leq \|\mathbf{U}\mathbf{x}\| \leq \|\mathbf{Q}^T\| \|\text{diag}(\mathbf{w})\mathbf{D}\mathbf{x}\|$$

where the norms are all  $l_1$ -norm, and they are either the  $l_1$ -norm of vector or the induced  $l_1$ -norm of matrix.

The proof of this theorem is included in the appendix.

**Remarks.** From the above, we can observe that the difference between the Cholesky  $l_1$ -norm and the  $l_1$ -norm is determined by  $\|\mathbf{Q}\|_{l_1}$  and  $\|\mathbf{Q}^T\|_{l_1}$  which are variable and hopefully reducible by selecting a proper  $\widetilde{\mathbf{W}}$  at the beginning. For example, the weight matrix  $\widetilde{\mathbf{W}}$  may be chosen such that its unique Cholesky factor  $\mathbf{Q}$  gives  $\|\mathbf{Q}\|_{l_1} \approx \|\mathbf{Q}^T\|_{l_1} \approx 1$ , without any loss of accuracy in modeling. This means the above bound is *tightenable* in principle. This result encourages us to further explore the useful subspaces in the Cholesky  $l_1$ -norm to approximate the total variation norm.

## 4 Experiments

In the experiment, we will evaluate our method in the context of interactive object segmentation, in which the unary term encodes the seeding information [11] and the pairwise potential is defined as under Eq. (1). We compare our method with the original graph cuts (GC) [2], the  $l_1$ -norm minimization via LP [5,6], and the  $l_2$ -norm minimization by QP [3,4].

### 4.1 Experimental Settings

*Data and Performance Measure.* To evaluate the performance gain in terms of computation. We perform the conventional LP and our proposed LP on GPU for synthetic data. In this experiment, we randomly generate the model parameters and apply the interior point method to solving the LP.

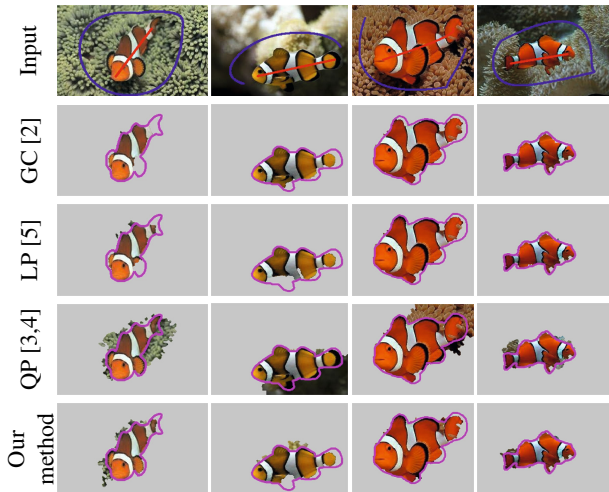
To evaluate the effectiveness of our method, we evaluate on a clownfish dataset and the Oxford interactive segmentation benchmark dataset [21]. Ground truth results and user input seeds on objects and backgrounds are provided in both datasets. The performances of the methods measured by the overlapping ratio between the labeled region and the ground truth object region: 
$$F = \frac{\text{size}(\text{Result Region} \cap \text{True Region})}{\text{size}(\text{Result Region} \cup \text{True Region})}.$$

*Implementation Issues.* We adopt superpixelization [22] as a preprocessing to reduce the computational cost. The number of superpixels is around 800 for all test images. We choose the average color of each superpixel to represent the superpixel. We implement all the methods in MATLAB. We used the `linprog` function and `quadprog` function. We use default option settings of the functions. The graph cuts is based on Michael Rubinstein’s implementation<sup>2</sup>. There are some parameters in the model for segmentation. We used  $c = 0.00001$ ,  $\lambda = 10$  in all the experiments. The threshold value for converting the continuous labels to binary labels is empirically chosen as 0.08. We also experiment on the effect of different threshold values. We perform the experiments on a PC with Intel Core i5-450M (2.4GHz) processor and 4GB memory.

<sup>2</sup> <http://www.mathworks.com/matlabcentral/fileexchange/21310-maxflow>

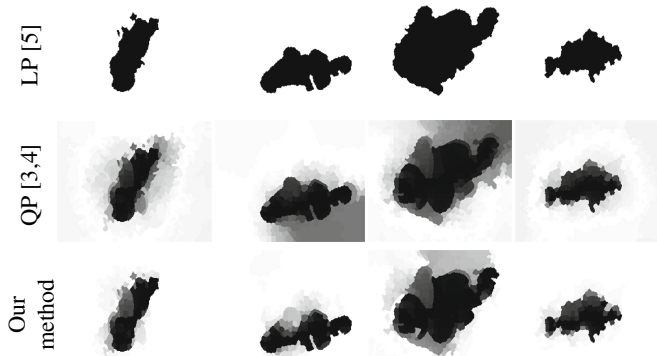
## 4.2 Results

*The Clownfish Dataset.* We first present and analyze the experimental results for the clownfish dataset which contains 62 images. See Fig. 2 for example segmentation results and input seeds. In addition to the manually drawn background seeds, we include the points at the image border as the background seeds in this experiment. As expected, we can see that the results of the conventional LP is very similar to those by graph cuts. A characteristic of them is that they suffer from the small-cut problem. In contrast, QP may produce larger regions due to the possible diffusion of labels at the boundaries. Thus, the resultant regions can be larger than the desired region. Our method compromises the two types of methods and the overall results may outperform the others, e.g., when LP suffers from small-cut problem and/or QP suffers from large-cut problem. We also visualize the continuous labels of conventional LP, QP and our method in Fig. 3. The solutions of LP are binary without thresholding, and the solutions of QP can be over-smooth. The boundaries in the solutions of our LP are clearer than QP, and the solutions are smoother than LP. Quantitative segmentation results of the clownfish dataset are shown in Fig. 4. The results show that QP slightly outperforms the conventional LP on this dataset, and our method slightly outperforms the others. From Table. 1, we can see that the computational cost of our compact LP model is comparable to QP and requires significantly less computational expenses compared to conventional LP. We also note that there is some minor difference between the results by graph cuts and those by conventional LP. We conjecture that the difference is a result of early termination of the interior point method for solving the LP.

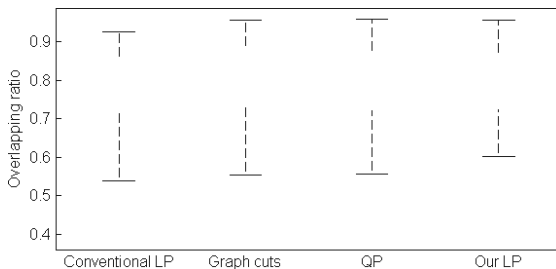


**Fig. 2.** Example results of the seed-initialized interactive segmentation on clownfish dataset. The results are shown as extracted image regions against the ground truth shape contours in purple.

*The Oxford Dataset.* We mainly evaluate our method on the Oxford dataset which contains 151 images. The user input seeds provided in this dataset are generally insufficient for producing a satisfactory segmentation. We adopt the robotuser [21] to simulate the



**Fig. 3.** Example labels of segmentation results in Fig. 2

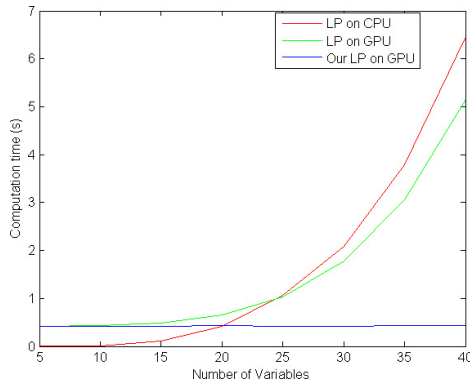


**Fig. 4.** Quantitative results on clownfish dataset

additional user interactions. By increasing the number of interactions, the segmentation results can finally become satisfactory. The maximum number of user interactions is set to 20 in our experiments. See Fig. 6 for example results. We can observe that GC and LP performs quite alike, while QP may produce larger regions. In most of the situations our methods produce more accurate segmentation results than QP. We present the solutions of QP and our method before thresholding in Fig. 7. The LP produces binary labels as expected, the QP produces smooth labels near the object boundaries and our method produces piecewise smooth labels with relatively clear discontinuities at the boundaries. The quantitative results are shown as red boxes in Fig. 8(a).

To quantitatively reveal the effect of the discontinuity preservability of our method, we further consider the robustness of the segmentation to threshold values. We hypothesize that the continuous labels with clear discontinuities at the boundaries will be robust to different threshold values. Therefore, we generate a vector of 100 threshold values equally spaced in  $[0, 1]$  for the evaluation. We apply all these threshold values to the continuous labels of QP and our method. Surprisingly, we observe that our method overwhelmingly outperforms the QP for almost all the threshold values in the sense of average overlapping ratio. See Fig. 8(b) for the plots of mean performance with standard deviation.

*Computational Costs.* We propose to compare the computational costs for solving conventional LP and our method using the same implementation of interior point method



**Fig. 5.** Comparison of computational times on GPU. As a reference, in [6], the average computation time was 0.66 sec. for GC on CPU and the 0.76 sec. for LP on GPU.

on CPU and GPU. The GPU implementation is realized by simply using `gpuarray` in MATLAB. We used small number of variables because MATLAB does not support sparse matrix in GPU. The results are shown in Fig. 5. From the plot we can observe that the computational cost of our method is almost unchanged but slightly oscillating when increasing the number of variables.

The statistics of the computational costs for our experiment on Oxford dataset are shown in Table 1. Very recently, a fast optimization approach has been proposed for solving a similar segmentation model [23]. However, the computational cost of their approach for 760 superpixels is 23.7 sec. on a machine with 2.7GHz Intel CPU.

**Table 1.** Comparison of computational costs

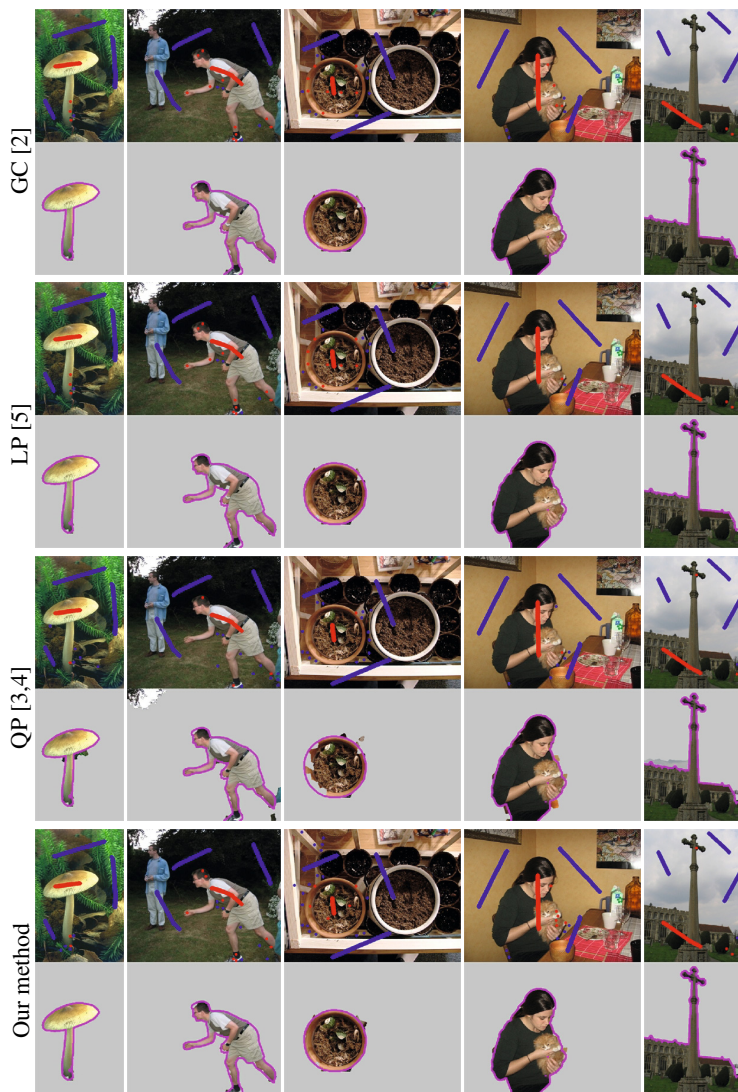
	CPU-LP[6,5]	CPU-QP [3,4]	CPU-Our method
Worst-case complexity	$O(N^6)$	$O(N^3)$	$O(N^3)$
Average time (s)	72.35	1.13	12.9

## 5 Conclusion and Future Work

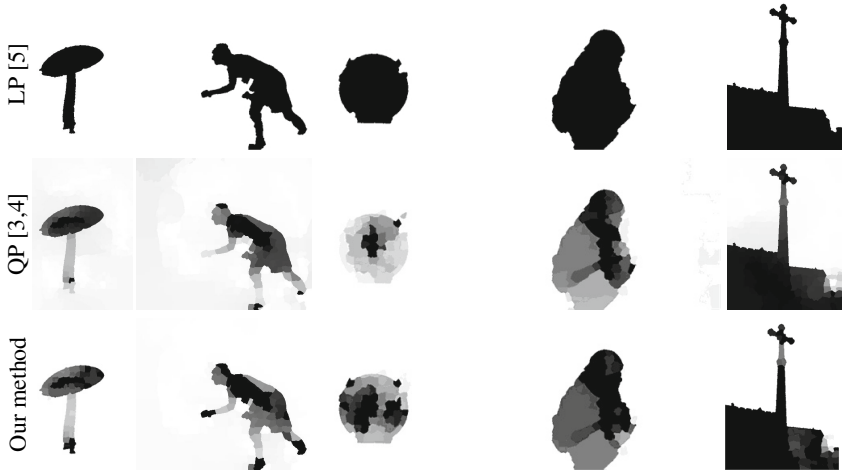
In this paper, we proposed a novel LP relaxation for the binary sub-modular MRF model. Our LP relaxation contains significantly fewer variables and constraints compared to the conventional LP. We also showed that our  $l_1$ -norm minimization is tightly related to the total variation minimization through mathematical analysis. Experimental results show that our method is significantly faster than the conventional LP, and it uniformly outperforms QP when converting the continuous labels to binary labels. Our model may be of use to other MRF models, e.g. the TV-MRF [24], as well as many applications, such as shape estimation [25,26,27].

**Acknowledgement.** Junyan Wang is supported by SUTD-ZJU Collaboration Research Grant 2012 SUTD-ZJU/RES/03/2012. Sai-Kit Yeung is supported by SUTD Start-Up Grant ISTD 2011 016, SUTD-MIT International Design Center Grant IDG31300106, and Singapore MOE Academic Research Fund MOE2013-T2-1-159.

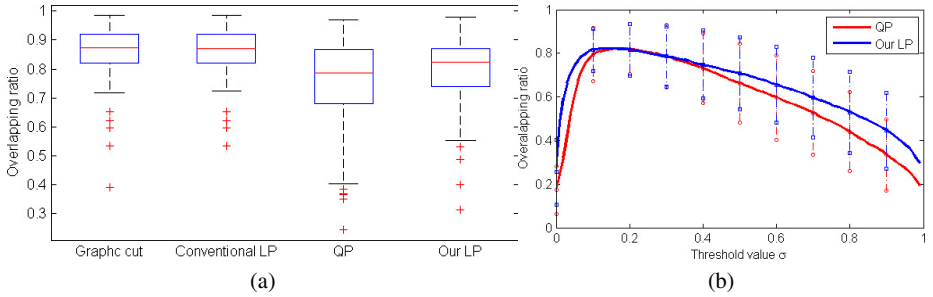




**Fig. 6.** Comparison of segmentation performance on Oxford dataset. The upper images in each row show the input images overlaid with input seeds. The lower images in each rows show extracted image regions against the ground truth shape contours in purple.



**Fig. 7.** Continuous labels before thresholding from LP, QP and our method on example inputs in Fig. 6.



**Fig. 8.** Quantitative results of the experiments on Oxford dataset. a) Comparison of segmentation accuracy. b) Comparison of QP and our LP for all threshold values.

## A Appendix

### A.1 Derivation of Eq. (5)

$$\begin{aligned} \sum_{ij} w_{ij}^e (x_i^2 + x_j^2 - 2x_i x_j) &= \sum_i x_i^2 \sum_j w_{ij}^e + \sum_j x_j^2 \sum_i w_{ij}^e - 2 \sum_{ij} w_{ij}^e x_i x_j \\ &= \sum_i x_i^2 \bar{w}_i + \sum_j x_j^2 \hat{w}_j - 2 \sum_{ij} w_{ij}^e x_i x_j = \mathbf{x}^T \widetilde{\mathbf{W}} \mathbf{x} \end{aligned}$$

where  $\widetilde{\mathbf{W}} = \text{diag}(\bar{w}) + \text{diag}(\hat{w}) - 2\mathbf{W}$ ,  $\mathbf{W} = [w_{ij}^e]$ .

### A.2 Proof of Proposition 1

*Proof.* The definition of  $\widetilde{\mathbf{W}}$  is as follows.

$$\widetilde{\mathbf{W}} = \text{diag}(\bar{w}) + \text{diag}(\hat{w}) - 2\mathbf{W}$$

where  $\bar{w}_j = \sum_k w_{jl} = \sum_k w_{lj'} = \hat{w}_{j'}$ , if  $j = j'$ . In short  $\text{diag}(\bar{w}) = \text{diag}(\hat{w})$ . Note that  $w_{jj'} = 0$  for  $j = j'$ . Hence, we have the following.

$$\widetilde{\mathbf{W}}_{jj'} = \begin{cases} 2\bar{w}_j, & \text{for } j = j' \\ -2w_{jj'}, & \text{otherwise} \end{cases}$$

Therefore, matrix  $\widetilde{\mathbf{W}}$  is a symmetric diagonal dominant matrix, and the diagonal elements are nonnegative. Such matrix is a positive semi-definite matrix.  $\square$

### A.3 Proof of Theorem 1

*Proof.* Substituting  $[\text{diag}(\mathbf{w})\mathbf{D}] = \mathbf{Q}^{N^2 \times N} \mathbf{R}^{N \times N}$  into Eq. (2), we obtain the following form of the boundary term.

$$\mathbf{B}_{l_1}(\mathbf{x}) = \|\mathbf{QRx}\|_{l_1}$$

where we applied the QR factorization. The  $l_2$  relaxation of this form will lead to

$$\mathbf{B}_{l_2}(\mathbf{x}) = (\mathbf{x}^T \mathbf{R}^T \mathbf{Q}^T \mathbf{QRx})^{1/2} = (\mathbf{x}^T \mathbf{R}^T \mathbf{Rx})^{1/2} = \|\mathbf{Rx}\|_{l_2}$$

The corresponding  $l_1^+$ -norm minimization is therefore the following

$$\mathbf{B}_{l_1^+}(\mathbf{x}) = \|\mathbf{Rx}\|_{l_1} = \|\mathbf{QRx}\|_{l_1}$$

Note that the Cholesky decomposition is unique and  $\mathbf{R}$  is upper-triangular. We can conclude that  $\mathbf{U} = \mathbf{R}$ .  $\square$

### A.4 Proof of Theorem 2

*Proof.* We prove the left hand side first.

$$\begin{aligned} \|\text{diag}(\mathbf{w}^e)\mathbf{Dx}\|_{l_1} &= \|\mathbf{QUx}\|_{l_1} \leq \|\mathbf{Q}\|_{l_1} \|\mathbf{Ux}\|_{l_1} \\ &\Leftrightarrow \frac{1}{\|\mathbf{Q}\|_{l_1}} \|\text{diag}(\mathbf{w}^e)\mathbf{Dx}\|_{l_1} \leq \|\mathbf{Ux}\|_{l_1} \end{aligned}$$

where we have replaced  $\mathbf{R}$  with  $\mathbf{U}$ . The right hand side is likewise.

$$\|\mathbf{Ux}\|_{l_1} = \|\mathbf{Q}^T \mathbf{QUx}\|_{l_1} \leq \|\mathbf{Q}^T\|_{l_1} \|\mathbf{QUx}\|_{l_1} = \|\mathbf{Q}^T\|_{l_1} \|\text{diag}(\mathbf{w}^e)\mathbf{Dx}\|_{l_1},$$

which completes the proof.  $\square$

## References

1. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts—a review. TPAMI 29, 1274–1279 (2007)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI 23, 1222–1239 (2001)

3. Grady, L.: Random walks for image segmentation. *TPAMI* 28, 1768–1783 (2006)
4. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: *CVPR*. IEEE (2007)
5. Li, H., Shen, C.: Interactive color image segmentation with linear programming. *Machine Vision and Applications* 21, 403–412 (2010)
6. Bhusnurmath, A., Taylor, C.J.: Graph cuts via  $L_1$  norm minimization. *TPAMI* 30, 1866–1871 (2008)
7. Derigs, U., Meier, W.: Implementing goldberg’s max-flow-algorithm — a computational investigation. *Zeitschrift für Operations Research* 33, 383–403 (1989)
8. Jamriska, O., Sykora, D., Hornung, A.: Cache-efficient graph cuts on structured grids. In: *IEEE CVPR*, pp. 3673–3680 (2012)
9. Lempitsky, V.S., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: *ICCV* (2009)
10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *TPAMI* 26, 147–159 (2004)
11. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: *ICCV* (2001)
12. Komodakis, N., Paragios, N., Tziritas, G.: Mrf energy minimization and beyond via dual decomposition. *TPAMI* 33, 531–552 (2011)
13. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnorr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Lellmann, J., Komodakis, N.: et al.: A comparative study of modern inference techniques for discrete energy minimization problems. In: *CVPR*, pp. 1328–1335 (2013)
14. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *IJCV* 1, 321–331 (1988)
15. Chan, T., Vese, L.: Active contours without edges. *TIP* 10, 266–277 (2001)
16. Ye, Y.: An  $o(n^3l)$  potential reduction algorithm for linear programming. *Mathematical Programming* 50, 239–258 (1991)
17. Megiddo, N.: Linear programming in linear time when the dimension is fixed. *Journal of the ACM (JACM)* 31, 114–127 (1984)
18. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 259–268 (1992)
19. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) *EMMCVPR 2005*. LNCS, vol. 3757, pp. 136–152. Springer, Heidelberg (2005)
20. Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization* 1, 15–22 (1991)
21. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: *CVPR* (2010)
22. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. *TPAMI* 31, 2290–2297 (2009)
23. Wang, P., Shen, C., van den Hengel, A.: A fast semidefinite approach to solving binary quadratic problems. In: *CVPR* (2013)
24. Wu, T.P., Yeung, S.K., Jia, J., Tang, C.K., Medioni, G.G.: A closed-form solution to tensor voting: Theory and applications. *TPAMI* 34, 1482–1495 (2012)
25. Yeung, S.K., Wu, T.P., Tang, C.K., Chan, T.F., Osher, S.J.: Normal estimation of a transparent object using a video. In: *TPAMI* (2014)
26. Yeung, S.K., Wu, T.P., Tang, C.K.: Extracting smooth and transparent layers from a single image. In: *CVPR* (2008)
27. Yeung, S.K., Wu, T.P., Tang, C.K., Chan, T.F., Osher, S.: Adequate reconstruction of transparent objects on a shoestring budget. In: *CVPR* (2011)

# On the Link between Gaussian Homotopy Continuation and Convex Envelopes

Hossein Mobahi and John W. Fisher III

Computer Science and Artificial Intelligence Lab. (CSAIL)  
Massachusetts Institute of Technology (MIT)  
`{hmobahi, fisher}@csail.mit.edu`

**Abstract.** The continuation method is a popular heuristic in computer vision for nonconvex optimization. The idea is to start from a simplified problem and gradually deform it to the actual task while tracking the solution. It was first used in computer vision under the name of graduated nonconvexity. Since then, it has been utilized explicitly or implicitly in various applications. In fact, state-of-the-art optical flow and shape estimation rely on a form of continuation. Despite its empirical success, there is little theoretical understanding of this method. This work provides some novel insights into this technique. Specifically, there are many ways to choose the initial problem and many ways to progressively deform it to the original task. However, here we show that when this process is constructed by Gaussian smoothing, it is optimal in a specific sense. In fact, we prove that Gaussian smoothing emerges from the best affine approximation to Vese’s nonlinear PDE. The latter PDE evolves any function to its convex envelope, hence providing the optimal convexification.

**Keywords:** Continuation Method, Diffusion Equation, Nonconvex Optimization, Vese’s PDE.

## 1 Introduction

Minimization of nonconvex energy functions arises frequently in computer vision. Examples include image segmentation [49], image alignment [67], image completion [46], dictionary learning [44], part-based models [25], and optical flow [62]. Unfortunately, a severe limitation of nonconvex problems is that finding their global minimum is generally intractable.

Some possible options for handling nonconvex tasks include<sup>1</sup> local optimization methods (e.g. gradient descent), convex surrogates, and the continuation method. Each of these ideas has its own merit and is preferred in certain settings. For example, local methods are useful when most local minima produce reasonably good solutions; otherwise the algorithm may get stuck in poor local minima. Convex surrogates are helpful when the nonconvexity of the task is mild, so that little structure is lost by the convex approximation. For example, it has been observed

---

<sup>1</sup> In this paper we only discuss deterministic schemes.

that for face recognition problem, the nonconvex sparsity encouraging  $\ell_0$  norm can be replaced by the convex  $\ell_1$  and yet produce impressive result [69]. Recently [23] proposed an a surrogate construction with bounded discrepancy between the solution of the convexified and original task.

The third idea is to utilize the continuation method. It solves a sequence of subproblems, starting from a convex (hence easy) task and progressively changing it to the actual problem while tracing the solution. Such complexity progression is in contrast to convex surrogates that produce a one-shot relaxation. Here, the solution of each subproblem guides solving the next one. This approach is often useful when the nonconvexity of the problem is so severe that convex surrogates cannot provide any meaningful approximation.

In this paper, we focus on optimization by the continuation method. The idea has been known to the computer vision community for at least three decades. This dates back to the works of Terzopoulos [63], Blake and Zisserman [6], and Yuille [72,73,74,75,76]. Since then, this technique has been used with growing interest to solve some difficult optimization problems. In particular, it is a key component in several state-of-the-art solutions for computer vision and machine learning problems as we discuss in Section 2.

Despite its long history and empirical success, there is little understanding about the fundamental aspects of this method. For example, it is known that the continuation method cannot always find the global minimizer of all nonconvex tasks. In fact, the quality of the solution attained by this approach heavily depends on the choice of the subproblems. However, there are endless choices for the initial convex problem, and endless ways to progressively change it to the original nonconvex task. Obviously, some of these choices should work better than the others. However, to date, there is no known principle for preferring one construction versus another.

For example, a possible way to construct the subproblem sequence is by Gaussian smoothing [50,47]. The idea is to convolve the original nonconvex function with an isotropic Gaussian kernel at various bandwidth values. This generates a sequence of functions varying from a highly smoothed (large bandwidth) to the actual nonconvex function (zero bandwidth). In fact, it can be proved that under certain conditions, enough smoothing can lead to a convex function [43]. The convexity implies that finding the minimizer of the smoothed function is easy. This minimizer is used to initialize the next subproblem, with slightly smaller bandwidth. The process repeats until reaching the last subproblem, which is the actual task. Since this type of progression goes from low-frequency toward fully detailed, it is also called *coarse-to-fine optimization*.

In this paper, we provide original insights into the choice of subproblems for the continuation method. Specifically, we prove that constructing the subproblems by Gaussian smoothing of the nonconvex function is optimal in a specific sense. Recall that the continuation method starts from an already convex objective and progressively maps it to the actual nonconvex function. Among infinite choices for the initial convex task, the *convex envelope* of the nonconvex problem is (in many senses) the best choice. Unfortunately, the convex envelope of an arbitrary function

is nontrivial and generally expensive to compute. Vese has shown that the convex envelope of a function can be generated by an evolutionary PDE [66]. However, this PDE does not have an analytical solution. Our contribution is to prove that the best affine approximation to Vese’s PDE results in the *heat equation*. The solution of the latter is known; it is the Gaussian convolution of the nonconvex function. Hence, Gaussian smoothing is the outcome of the best affine approximation of the (nonlinear) convex envelope generating PDE.

## 2 Related Works

Here we review some remarkable works that rely on the concept of optimization by the continuation method.

In computer vision, the early works around this concept were Blake and Zisserman’s *Graduated Non-Convexity* (GNC) [6] as well as works by Terzopoulos’ [63], both on surface reconstruction problems. Shortly afterward, Geiger and Giroi [29] as well as Yuille [72] used similar concepts from a statistical physics viewpoint. The latter method is known as *Mean Field Annealing* (MFA). Motivated by problems in stereo and template matching, Yuille popularized MFA in a series of works [30,72,73,74,75,76]. MFA is a *deterministic* variant of simulated annealing<sup>2</sup>, where the stochastic behavior is approximated by the mean state. This model starts from high temperature (smoother energy and hence fewer extrema) and gradually cools down toward the desired optimization task.

Since then, the concept of optimization by the continuation method has been successfully utilized in various vision applications such as image segmentation [9], shape matching [64], image deblurring [8], image denoising [54,51], template matching [22], pixel correspondence [40], active contours [18], Hough transform [39], edge detection [78], early vision [5], robot navigation [52], and image matting [53]. In fact, many computer vision methods that rely on multiscale image representation within the optimization loop are implicitly performing the continuation method, e.g. for image alignment [47].

The growing interest in this method within computer vision community has made it one of the most popular solutions for the contemporary nonconvex minimization problems. Just within the past few years, it has been utilized for low-rank matrix recovery [45], error correction [48], super resolution [19], photometric stereo [70], image segmentation [35], face alignment [57], 3D surface estimation [1], motion estimation in videos [61], optical flow [10,62], shape and illumination recovery [2], and dense correspondence of images [36]. The last three are in fact *state of the art* solutions for their associated problems.

Independently, the machine learning community has been using similar ideas for optimization. Notably, Rose popularized the method of *Deterministic Annealing* (DA) for clustering problems [55]. This method starts from the *maximum entropy* solution (the simple task), and gradually reduces the entropy

<sup>2</sup> There is some conceptual similarity between simulated annealing (SA) and some of the continuation methods. However, SA is an MCMC method and is known for its very slow convergence. The continuation methods studied here are deterministic and converge much faster [7,40].

to only leave the actual objective function. Variants of DA have been recently used for learning occluding objects [20], object tracking [33], image deblurring [41], clustering boolean data [26], graph clustering [56], unsupervised language learning [60]. Chapelle has utilized continuation in various applications such as semi-supervised learning [12,13,59], semi-supervised structured output [21], multiple instance learning [28], and ranking [14]. Bengio argues that some recent breakthroughs in the training of deep architectures [34,24], has been made by algorithms that use some form of continuation for learning [4].

Other examples that utilize continuation for optimization are clustering [32], graph matching [31,77,42], multiple instance learning [37], language modeling [3], manifold sampling [58], and  $\ell_0$  norm minimization [65]. One of the most interesting applications, however, has been recently introduced by [16,17]. The goal is to find optimal parameters in computer programs. The authors define a smoothing operator acting on programs to construct smooth interpretations. They then seek the optimal parameters by starting from highly smoothed interpretations and gradually reducing the smoothing level. The idea is further extended to smoothing the space of proofs and seeking the optimal proof to a problem by the continuation method [15].

Throughout this paper, we use  $x$  for scalars,  $\mathbf{x}$  for vectors,  $\mathbf{X}$  for matrices, and  $\mathcal{X}$  for sets. Here  $\|\mathbf{x}\|$  means  $\|\mathbf{x}\|_2$  and  $\triangleq$  means equality by definition. When a function is denoted as  $g(\mathbf{x}; t)$ , the gradient  $\nabla$ , Hessian  $\nabla^2$  and Laplacian  $\Delta$  operators are only applied to the vector  $\mathbf{x}$  and not  $t$ . The convolution operator is denoted by  $\star$ . The isotropic Gaussian kernel with standard deviation  $\sigma$  is shown by  $k_\sigma$ ,

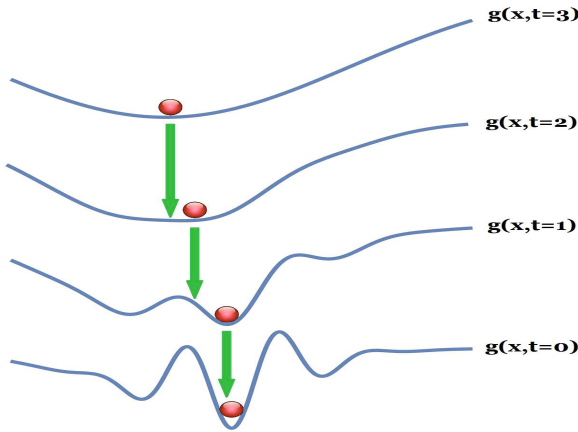
$$k_\sigma(\mathbf{x}) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^{\dim(\mathbf{x})}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}.$$

### 3 Optimization by Continuation

Given an (possible nonconvex) objective function  $f : \mathcal{X} \rightarrow \mathbb{R}$ , where  $\mathcal{X} = \mathbb{R}^n$ . Consider an embedding of  $f$  into a family of functions  $g : \mathcal{X} \times \mathcal{T}$ , where  $\mathcal{T} \triangleq [0, \infty)$ , with the following properties. First,  $g(\mathbf{x}, 0) = f(\mathbf{x})$ . Second, when  $t \rightarrow \infty$ , then  $g(\mathbf{x}, t)$  is strictly convex and has a unique minimizer (denoted by  $\mathbf{x}_\infty$ ). Third,  $g(\mathbf{x}, t)$  is continuously differentiable in  $\mathbf{x}$  and  $t$ . Such embedding  $g$  is sometimes called a *homotopy*, as it continuously transforms one function to another.

Define the curve  $\mathbf{x}(t)$  for  $t \geq 0$  as one with the following properties. First,  $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}_\infty$ . Second,  $\forall t \geq 0 \quad ; \quad \nabla g(\mathbf{x}(t), t) = \mathbf{0}$ . Third,  $\mathbf{x}(t)$  is *continuous* in  $t$ . This curve simply sweeps a specific stationary path of  $g$  originated at  $\mathbf{x}_\infty$ , as the parameter  $t$  progresses backward (See Figure 1). In general, such curve neither needs to exist, nor needs to be unique. However, with some additional assumptions on  $g$ , it is possible to guarantee *existence* and *uniqueness* of  $\mathbf{x}(t)$ , e.g. by Theorem 3 of [71].





**Fig. 1.** Plots show  $g$  versus  $x$  for each fixed time  $t$ . The marble indicates the location for  $x(t)$ .

---

**Algorithm 1.** Algorithm for Optimization by the Continuation Method

---

- 1: Input:  $f : \mathcal{X} \rightarrow \mathbb{R}$ , Sequence  $t_0 > t_1 > \dots > t_m = 0$ .
  - 2:  $\mathbf{x}_0 =$  global minimizer of  $g(\mathbf{x}; t_0)$ .
  - 3: **for**  $k = 1$  **to**  $m$  **do**
  - 4:    $\mathbf{x}_k =$  Local minimizer of  $g(\mathbf{x}; t_k)$ , initialized at  $\mathbf{x}_{k-1}$ .
  - 5: **end for**
  - 6: Output:  $\mathbf{x}_m$
- 

In practice, the continuation method is realized as follows. First,  $\mathbf{x}_\infty$  is either derived analytically<sup>3</sup> or approximated numerically as  $\arg\min_{\mathbf{x}} g(\mathbf{x}; t)$  for large enough  $t$ . The latter can use standard convex optimization tools as  $g(\mathbf{x}; t)$  approaches a convex function in  $\mathbf{x}$  for large  $t$ . Then, the stationary path  $\mathbf{x}(t)$  is numerically tracked until  $t = 0$  (See Algorithm 1). As discussed in Section 2, for a wide range of applications, the continuation solution  $\mathbf{x}(0)$  often provides a good local minimizer of  $f(\mathbf{x})$ , if not the global minimizer.

## 4 Motivation for Gaussian Homotopy

There are some limited number of studies on very specific problems which guarantee the continuation method can discover the global minimum of the problem. An example of this kind is the work by Yuille and Kosowsky [38] on assignment problem. However, in general, there is no guarantee for the continuation method to reach the global minimizer of  $f(\mathbf{x})$ .

In fact, the quality of the solution attained by the continuation method depends on the choice of the homotopy map  $g(\mathbf{x}; t)$ . It is therefore crucial to choose

---

<sup>3</sup> For functions whose tails vanish fast enough, this point is simply the center of mass of the function [43].

$g(\mathbf{x}; t)$  in the most sensible way. Currently, there is no pointer in the literature to justify one homotopy versus others. For example, Fua and Leclerc [27] use  $g(\mathbf{x}, t) = f(\mathbf{x}) + t\mathbf{x}^T \mathbf{A}\mathbf{x}$ , where  $\mathbf{A} \succ \mathbf{O}$ . Blake and Zisserman [6] utilize a task-tailored polynomial map. Methods based on deterministic annealing use negative entropy  $g(\mathbf{x}, t) = f(\mathbf{x}) + t\mathbf{x}^T \log(\mathbf{x})$  (applicable only to nonnegative variables  $\mathbf{x}$ ) [55,56]. Nielson [50] and Mobahi [47] use Gaussian homotopy by convolving  $f$  with the Gaussian kernel, i.e.  $g(\cdot, t) = f \star k_t$ . When Gaussian homotopy is used for optimization, it is sometimes called *coarse-to-fine optimization*<sup>4</sup>.

In this section, we claim that Gaussian homotopy is optimal in a specific sense; it solves the best affine approximation (around the origin of the function space, i.e. the function  $f(\mathbf{x}) = 0$ ) to a nonlinear PDE that generates convex envelopes. We will postpone the proof to the next section.

By definition, a homotopy for optimizing  $f(\mathbf{x}) = g(\mathbf{x}; 0)$  must continuously convexify it to  $g(\mathbf{x}, \infty)$ . Among all convex choices  $g(\mathbf{x}, \infty)$ , the *convex envelope* is the optimal convexifier of  $f$  in many senses. For example, it provides the best (largest) possible convex underestimator of the  $f$ . Furthermore, geometrically, the convex envelope is precisely the function whose epigraph coincides with the convex hull of the epigraph of  $f$ .

The convex envelope, however, is often unknown itself and its computation is generally very expensive. Interestingly, Vese [66] has characterized an elegant PDE that if its initial condition is set to  $f(\mathbf{x})$ , it evolves toward the convex envelope of  $f$  and reaches there in the limit  $t \rightarrow \infty$ . More precisely, this is a nonlinear PDE that evolves a function  $v(\mathbf{x}; t)$  for  $v: \mathcal{X} \times \mathcal{T}$  as the following,

$$\frac{\partial}{\partial t} v = \sqrt{1 + \|\nabla v\|^2} \min\{0, \lambda_{\min}(\nabla^2 v)\} \quad , \quad \text{s.t. } v(\cdot; 0) = f(\cdot), \quad (1)$$

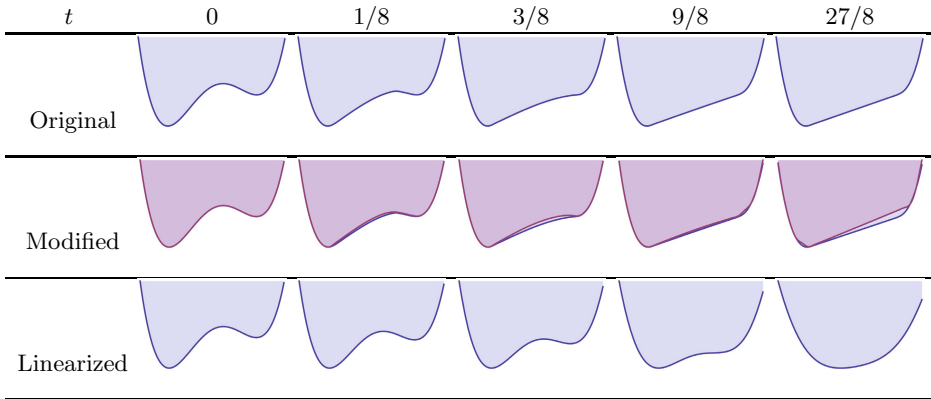
where  $\lambda_{\min}(\nabla^2 v)$  is the smallest (sign considered) eigenvalue of the Hessian of  $v$ . Intuitively, this PDE acts like a conditional diffusion process. At any evolution moment  $t$ ,  $v(\mathbf{x}; t)$  is spatially diffused at points  $\mathbf{x}$  where  $v(\mathbf{x}; t)$  is nonconvex and is left as is at points  $\mathbf{x}$  where  $v(\mathbf{x}; t)$  is convex (nonconvexity and convexity of  $v$  here are w.r.t. to the variable  $\mathbf{x}$ ). Consequently, throughout the evolution, nonconvex structures diminish by diffusion while convex structures sustain.

Vese's PDE involves the nonsmooth function  $\min$ , which complicates its treatment for the purpose of this paper<sup>5</sup>. Hence, we introduce the *modified Vese's PDE* by replacing  $\min$  with a smooth approximation,

$$\begin{aligned} \frac{\partial}{\partial t} u &= \sqrt{1 + \|\nabla u\|^2} m(\boldsymbol{\lambda}(\nabla^2 u)) \quad , \quad \text{s.t. } u(\cdot; 0) = f(\cdot) \quad (2) \\ m(\boldsymbol{\lambda}) &\triangleq \frac{\sum_{k=1}^n \lambda_k e^{-\frac{\lambda_k}{\phi}}}{1 + \sum_{k=1}^n e^{-\frac{\lambda_k}{\phi}}} \end{aligned}$$

<sup>4</sup> This is because moving from large to small  $t$  reveals coarse to fine structure of the optimization landscape.

<sup>5</sup> The difficulty arises later in Section 5, where we need to differentiate the r.h.s. of (1), but  $\min$  is not differentiable.



**Fig. 2.** Evolution of the function  $x^4 + 2x^3 - 12x^2 - 2x$  by Vese’s original PDE (1) (top), versus its modified (middle) and linearized (3) (bottom) forms. Since the difference between the original and modified version of Vese’s PDE is very subtle, in the middle row the modified solution (magenta) is superimposed on the original solution (blue). The modified version uses  $\delta = 10$  to make the the two visually distinct (with  $\delta = 1$  these plots already become indistinguishable). While all three evolutions convexify the initial function, the original and modified Vese’s equations respectively generate the perfect and close approximate to the convex envelope.

where  $\delta > 0$ , and  $\boldsymbol{\lambda} \triangleq (\lambda_1, \dots, \lambda_n)$  is a  $n \times 1$  vector. Observe that  $\lim_{\delta \rightarrow 0^+} m(\boldsymbol{\lambda}) = \min\{0, \lambda_1, \dots, \lambda_n\}$ . Hence, we can construct an arbitrarily close approximation to  $\min\{0, \lambda_1, \dots, \lambda_n\}$  by choosing a small enough  $\delta > 0$ . Although Vese’s PDE and its modified form are not identical, from practical viewpoint their difference is often negligible (See Figure 2, the top and middle rows). Hence, we proceed with the modified Vese’s PDE for our analysis in Section 5, solely for technical reasons.

Neither the original nor the modified versions of Vese’s PDE can be solved analytically due to their highly *nonlinear* nature. However, in Section 5 we will prove that the best affine approximation of the modified Vese’s operator around the origin of the function space (i.e. the function  $f(\mathbf{x}) = 0$ ) is the *Laplace* operator, hence the following approximation (See Figure 2 for an illustrative example),

$$\frac{\partial}{\partial t} \hat{u} = \frac{1}{n+1} \Delta \hat{u} \quad , \quad \hat{u}(\cdot; 0) = f(\cdot). \quad (3)$$

The resulted PDE (3) is essentially the *heat equation* [68] on the domain  $\mathcal{X} = \mathbb{R}^n$  with the initial condition  $\hat{u}(\mathbf{x}, 0) = f(\mathbf{x})$ . The solution of the heat equation in (3) is known to have the following form,

$$\hat{u}(\mathbf{x}; t) = \left(\frac{n+1}{4\pi t}\right)^{\frac{n}{2}} [f(\cdot) \star e^{-\frac{\|\cdot\|^2 (n+1)}{4t}}](\mathbf{x}).$$

The function  $\hat{u}$  can be reparameterized in its scale parameter via  $\sigma^2 = \frac{2t}{n+1}$ . This only changes the speed of progression, which is not crucial for our conclusion here. Hence, the homotopy can be expressed as the convolution of  $f$  with the Gaussian kernel  $k_\sigma$  as below,

$$\hat{h}(\mathbf{x}; \sigma) = [f \star k_\sigma](\mathbf{x}).$$

This approximation buys us a significant benefit in practice, for the following reason. While the nonlinear operator appearing in the original PDE (1) or its modified version (2) does not allow for a closed form solution, the linear PDE (3) makes this possible, provided that the integral for the Gaussian convolution of  $f$  in (4) has a closed form expression. The latter is true for some important classes of functions including polynomials and Gaussian bumps. Both of these classes are rich enough to represent almost any function, respectively through Taylor series and Gaussian Radial-Basis-Functions (RBF). For example, [47] uses these function spaces in order to formulate the image alignment problem and then solves it by Gaussian homotopy continuation.

Note that unlike Vese's equation that always evolves the nonconvex function to a convex one (in fact, to its convex envelope), heat equation does not necessarily produce a convex function. However, it does so for functions that on average (across all points) are convex<sup>6</sup>. There exist sufficient conditions<sup>7</sup> on the nonconvex functions to guarantee their convexity after enough smoothing [43].

## 5 Affine Approximation of Modified Vese's Operator

Here we prove our earlier claim that the best affine approximation to the modified Vese's PDE around the origin of function space (i.e. the function  $f(\mathbf{x}) = 0$ ) leads to the Laplace operator. We first need a few definitions. In the sequel, let  $\mathcal{H}$  be the space of twice differentiable functions  $h : \mathcal{X} \rightarrow \mathbb{R}$ , where  $\mathcal{X} \triangleq \mathbb{R}^n$ . We consider linear and nonlinear operators that have the form  $\mathcal{H} \rightarrow \mathcal{H}$  and denote them by  $\mathcal{L}$  and  $\mathcal{N}$  respectively. We say an operator is linear if and only if it obeys  $\forall h_1 \in \mathcal{H}, h_2 \in \mathcal{H}, a \in \mathbb{R}, b \in \mathbb{R}; \mathcal{L}\{ah_1 + bh_2\} = a\mathcal{L}\{h_1\} + b\mathcal{L}\{h_2\}$ .

**Definition 1 (Affine Operator).** *An affine operator is the form  $\mathcal{L}\{h\} + c$  where  $\mathcal{L}$  is a linear operator in  $h$  and  $c$  is constant in  $h$ .*

**Definition 2 (Modified Vese's Operator).** *The modified Vese's operator is defined as the operator acting on the function  $h \in \mathcal{H}$  to return  $\sqrt{1 + \|\nabla h\|^2} m(\lambda(\nabla^2 h))$ ,*

$$\text{where } m(\lambda) \triangleq \frac{\sum_{k=1}^n \lambda_k e^{-\frac{\lambda_k}{\delta}}}{1 + \sum_{k=1}^n e^{-\frac{\lambda_k}{\delta}}}.$$

<sup>6</sup> For example, in univariate functions  $f(x)$ , the Gaussian smoothed function is  $g(x; \sigma) \triangleq [f \star k_\sigma](x)$  and hence  $g''(x; \sigma) \triangleq [f'' \star k_\sigma](x)$ . When  $\sigma \rightarrow \infty$ , convolution with  $k_\sigma(x)$  acts an averaging operator. Hence if  $\int_{\mathcal{X}} f''(x) dx > 0$ , i.e.  $f$  is on average convex, then  $\forall x; g''(x; \sigma) > 0$  (i.e.  $g(x; \sigma)$  is convex everywhere) a large enough  $\sigma$ .

<sup>7</sup> For example, if the tails vanish fast enough and the  $-\infty < \int_{\mathcal{X}} f(\mathbf{x}) d\mathbf{x} < \infty$ , then it is guaranteed that for a large enough  $\sigma$ ,  $g(\mathbf{x}; \sigma)$  is convex. See [43] for details.

**Definition 3 (Best Affine Approximation of a Nonlinear Operator).** Consider a  $h \in \mathcal{H}$  and suppose it is resulted by perturbing some function  $h^* \in \mathcal{H}$  by the function  $\epsilon\phi$ , that is,

$$h = h^* + \epsilon\phi, \quad (4)$$

where  $\phi \in \mathcal{H}$  and  $\epsilon \in \mathbb{R}$ . Suppose  $\mathcal{N}\{h^* + \epsilon\phi\}$  is differentiable in  $\epsilon$  around  $\epsilon = 0$  so that its first order expansion w.r.t.  $\epsilon$  obeys  $\mathcal{N}\{h\} = \mathcal{N}\{h^* + \epsilon\phi\} = \mathcal{N}\{h^*\} + \epsilon(\frac{d}{d\epsilon}\mathcal{N}\{h^* + \epsilon\phi\})|_{\epsilon=0} + o(\epsilon)$ . The “best affine approximation” to  $\mathcal{N}\{h\}$  around the fixed function  $h^*$  is defined as discarding the term  $o(\epsilon)$  from the above, so that,

$$c_{opt} \triangleq \mathcal{N}\{h^*\} \quad , \quad \mathcal{L}_{opt}\{h\} \triangleq \epsilon(\frac{d}{d\epsilon}\mathcal{N}\{h^* + \epsilon\phi\})|_{\epsilon=0}. \quad (5)$$

**Theorem 1.** The best affine approximation of the modified Vese’s operator, acting on functions close to the zero function ( $h^*(\mathbf{x}) = 0$ ) and with bounded zeroth, first and second order derivatives, is equal to  $\frac{1}{n+1} \Delta$ .

*Proof.* The nonlinear operator of interest here is the modified Vese’s operator,

$$\mathcal{N}\{h\} \triangleq \sqrt{1 + \|\nabla h\|^2} m(\boldsymbol{\lambda}(\nabla^2 h)). \quad (6)$$

Observe that for this operator,  $\mathcal{N}\{h^* + \epsilon\phi\}$  is differentiable in  $\epsilon$ . Since  $h^*(\mathbf{x}) = 0$ , (5) implies that  $c_{opt} = \mathcal{N}(0) = 0$ . Note that we exploited the fact that  $\phi$ ,  $\nabla\phi$  and  $\boldsymbol{\lambda}(\nabla^2\phi)$  are bounded at any  $\mathbf{x} \in \mathcal{X}$  so that by  $\epsilon = 0$  one can conclude  $h = \epsilon\phi = 0$ ,  $\|\nabla h\|^2 = \epsilon^2\|\nabla\phi\|^2 = 0$ , and  $\boldsymbol{\lambda}(\nabla^2 h) = \epsilon\boldsymbol{\lambda}(\nabla^2\phi) = 0$ .

We now focus on computing  $\mathcal{L}_{opt}\{h\}$  using (5), which amounts to finding,

$$\epsilon \left( \frac{d}{d\epsilon} \sqrt{1 + \|\nabla\epsilon\phi\|^2} m(\boldsymbol{\lambda}(\nabla^2\epsilon\phi)) \right) |_{\epsilon=0}. \quad (7)$$

We proceed by first computing  $\left( \frac{d}{d\epsilon} \sqrt{1 + \epsilon^2\|\nabla\phi\|^2} m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0}$ . By chain rule, this is equivalent to,

$$\begin{aligned} & \left( \frac{d}{d\epsilon} \sqrt{1 + \epsilon^2\|\nabla\phi\|^2} \right) |_{\epsilon=0} \left( m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0} \\ & + \left( \frac{d}{d\epsilon} m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0} \left( \sqrt{1 + \epsilon^2\|\nabla\phi\|^2} \right) |_{\epsilon=0}. \end{aligned}$$

Since  $\nabla\phi$  and  $\boldsymbol{\lambda}(\nabla^2\phi)$  are assumed to be bounded, at  $\epsilon = 0$ , the above expression can be written as

$$\left( \frac{d}{d\epsilon} \sqrt{1 + \epsilon^2\|\nabla\phi\|^2} \right) |_{\epsilon=0} m(\mathbf{0}) + \left( \frac{d}{d\epsilon} m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0} \sqrt{1 + 0}. \quad (8)$$

Hence the above sum simplifies to  $\left( \frac{d}{d\epsilon} m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0}$ . Applying chain rule again, this becomes  $\left( \nabla m(\epsilon\boldsymbol{\lambda}(\nabla^2\phi)) \right) |_{\epsilon=0} \bullet \left( \frac{d}{d\epsilon} \epsilon\boldsymbol{\lambda}(\nabla^2\phi) \right) |_{\epsilon=0}$ , where  $\bullet$  denotes the inner product between two  $n \times 1$  vectors. Evaluating it at  $\epsilon = 0$  yields

$\nabla m(\mathbf{0}) \bullet \boldsymbol{\lambda}(\nabla^2 \phi)$ . Since  $\nabla m(\mathbf{0}) = \frac{1}{n+1} \mathbf{1}$ , where  $\mathbf{1}$  is a  $n \times 1$  vector with all entries equal to 1, the expression becomes  $\frac{1}{n+1} \mathbf{1} \bullet \boldsymbol{\lambda}(\nabla^2 \phi)$ . However,  $\mathbf{1} \bullet \boldsymbol{\lambda}(\nabla^2 \phi)$  is simply the sum of the eigenvalues, thus it is  $\text{Trace}(\nabla^2 \phi)$ . Finally, since  $\text{Trace}(\nabla^2 \phi)$  is sum of the diagonals of the Hessian matrix for  $\phi$ , it is equivalent to the Laplacian  $\Delta \phi$ . In summary, we just derived that,

$$\left( \frac{d}{d\epsilon} \sqrt{1 + \epsilon^2 \|\nabla \phi\|^2} m(\epsilon \boldsymbol{\lambda}(\nabla^2 \phi)) \right)_{|\epsilon=0} = \frac{1}{n+1} \Delta \phi, \quad (9)$$

Going back to the definition of  $\mathcal{L}_{\text{opt}}\{h\}$  in (7), it follows that,

$$\mathcal{L}_{\text{opt}}\{h\} \triangleq \epsilon \left( \frac{d}{d\epsilon} \sqrt{1 + \|\nabla \epsilon \phi\|^2} m(\boldsymbol{\lambda}(\nabla^2 \epsilon \phi)) \right)_{|\epsilon=0} \quad (10)$$

$$= \epsilon \frac{1}{n+1} \Delta \phi. \quad (11)$$

We now manipulate  $\epsilon \frac{1}{n+1} \Delta \phi$ . Moving  $\epsilon$  inside, it can be equivalently be written as  $\frac{1}{n+1} \Delta(\epsilon \phi)$ . However, by (4),  $\epsilon \phi$  is just the definition of  $h - h^*$ . Using that fact that  $h^* = 0$ , we obtain,

$$\mathcal{L}_{\text{opt}}\{h\} = \frac{1}{n+1} \Delta h. \quad (12)$$

□

## 6 Discussion and Future Works

This work provided new insights into the optimization by homotopy continuation. We showed that constructing the homotopy by Gaussian convolution is optimal in a specific sense. That is, the Gaussian homotopy is the result of the *best affine approximation* to the modified Vese's PDE. Vese's PDE is interesting for homotopy construction because it evolves the nonconvex function to its convex envelope. The convex envelope provides optimal convexification for nonconvex functions. However, Vese's PDE does not have any closed form solution due to its nonlinearity, hence cannot be used in practice. In contrast, Gaussian smoothing can be computed in closed form for a large family of functions, including those represented by polynomials or Gaussian radial basis functions.

Recall that the optimality of the Gaussian homotopy is proved here in a certain setting; when the modified Vese's PDE is *linearized* around the *origin* of the function space  $h^*(\mathbf{x}) = 0$ . Such linearization severely degrades the fidelity of the approximation. An important question is whether linearity or working around the origin could be relaxed without losing the advantage of closed form solution to the PDE. Such extension is a clear direction for future studies.

A possibility might be exploiting the conditional diffusion property of Vese's PDE. Remember this PDE only diffuses nonconvex regions throughout the evolution, and is insensitive to convex regions. If the nonconvex and convex parts of an objective function could be separated, applying Gaussian smoothing only

to the nonconvex part might produce a better approximation to Vese's PDE, as opposed to smoothing the entire objective function. This is obviously a non-linear evolution because it requires a switching behavior between convex and nonconvex regions.

Another direction for improving the approximation quality is to manipulate the objective function. For example, transforming the objective function  $f(\mathbf{x})$  to  $-\exp(-M f(\mathbf{x}))$ , where  $M > 0$  is a large constant, does not alter the global minimizers. However, the latter form may lead to a better agreement between the linearized and original PDE, when used as their initial condition. The intuition is that, the transformed function is very close to zero almost everywhere (recall that our linearization is around  $h^*(\mathbf{x}) = 0$ ). Smoothing the exponentially transformed function is also pursued by [23], but for one-shot convexification. Note that the exponential transform followed by the diffusion process is related to the *Burgers'* PDE [11]. This connection might be of value, but does not completely answer all questions. That is because while the solution of Burgers' equation has a known form, it involves Gaussian convolution of  $\exp(-M f(\mathbf{x}))$ , which may not have an analytical form for interesting choices of  $f(\mathbf{x})$ , e.g. polynomials. This integral also arises in [23] and is approximated by sampling based methods.

**Acknowledgment.** This work is partially funded by the Shell Research. First author is thankful to Alan L. Yuille (UCLA) and Steven G. Johnson (MIT) and Vadim Zharnitsky (UIUC) for discussions, and grateful to William T. Freeman (MIT) for supporting this work.

## References

1. Balzer, J., Mörwald, T.: Isogeometric finite-elements methods and variational reconstruction tasks in vision - a perfect match. In: CVPR (2012)
2. Barron, J.: Shapes, Paint, and Light. Ph.D. thesis, EECS Department, University of California, Berkeley (August 2013)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: ICML (2009)
4. Bengio, Y.: Learning Deep Architectures for AI. Now Publishers Inc. (2009)
5. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision* 19(1), 57–91 (1996)
6. Blake, A., Zisserman, A.: *Visual Reconstruction*. MIT Press (1987)
7. Blake, A.: Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE PAMI* 11(1), 2–12 (1989)
8. Boccuto, A., Discepoli, M., Gerace, I., Pucci, P.: A gnc algorithm for deblurring images with interacting discontinuities (2002)
9. Brox, T.: From pixels to regions: partial differential equations in image analysis. Ph.D. thesis, Saarland University, Germany (April 2005)
10. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(3), 500–513 (2011)

11. Burgers, J.M.: The nonlinear diffusion equation: asymptotic solutions and statistical problems. D. Reidel Pub. Co. (1974)
12. Chapelle, O., Chi, M., Zien, A.: A continuation method for semi-supervised svms. pp. 185–192. ICML 2006 (2006)
13. Chapelle, O., Sindhvani, V., Keerthi, S.S.: Optimization techniques for semi-supervised support vector machines. *J. Mach. Learn. Res.* 9, 203–233 (2008)
14. Chapelle, O., Wu, M.: Gradient descent optimization of smoothed information retrieval metrics. *Inf. Retr.* 13(3), 216–235 (2010)
15. Chaudhuri, S., Clochard, M., Solar-Lezama, A.: Bridging boolean and quantitative synthesis using smoothed proof search. *SIGPLAN Not* 49(1), 207–220 (2014)
16. Chaudhuri, S., Solar-Lezama, A.: Smooth interpretation. In: *PLDI*. pp. 279–291. ACM (2010)
17. Chaudhuri, S., Solar-Lezama, A.: Smoothing a program soundly and robustly. In: Gopalakrishnan, G., Qadeer, S. (eds.) *CAV 2011*. LNCS, vol. 6806, pp. 277–292. Springer, Heidelberg (2011)
18. Cohen, L.D., Gorre, A.: Snakes: Sur la convexite de la fonctionnelle denergie (1995)
19. Coupé, P., Manjón, J.V., Chamberland, M., Descoteaux, M., Hiba, B.: Collaborative patch-based super-resolution for diffusion-weighted images. *NeuroImage* 83, 245–261 (2013)
20. Dai, Z., Lücke, J.: Unsupervised learning of translation invariant occlusive components. In: *CVPR*, pp. 2400–2407 (2012)
21. Dhillon, P.S., Keerthi, S.S., Bellare, K., Chapelle, O., Sundararajan, S.: Deterministic annealing for semi-supervised structured output learning. In: *AISTATS 2012*, vol. 15 (2012)
22. Dufour, R.M., Miller, E.L., Galatsanos, N.P.: Template matching based object recognition with unknown geometric parameters. *IEEE Transactions on Image Processing* 11(12), 1385–1396 (2002)
23. Dvijotham, K., Fazel, M., Todorov, E.: Universal convexification via risk-aversion. In: *UAI* (2014)
24. Erhan, D., Manzagol, P.A., Bengio, Y., Bengio, S., Vincent, P.: The difficulty of training deep architectures and the effect of unsupervised pre-training. In: *AISTATS*, pp. 153–160 (2009)
25. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(9), 1627–1645 (2010)
26. Frank, M., Streich, A.P., Basin, D., Buhmann, J.M.: Multi-assignment clustering for boolean data. *J. Mach. Learn. Res.* 13(1), 459–489 (2012)
27. Fua, P., Leclerc, Y.: Object-centered surface reconstruction: combining multi-image stereo shading. *International Journal on Computer Vision* 16(1), 35–56 (1995)
28. Gehler, P., Chapelle, O.: Deterministic annealing for multiple-instance learning. In: *AISTATS 2007*, pp. 123–130. Microtome, Brookline (2007)
29. Geiger, D., Girosi, F.: Coupled markov random fields and mean field theory. In: *NIPS*, pp. 660–667. Morgan Kaufmann (1989)
30. Geiger, D., Yuille, A.L.: A common framework for image segmentation. *International Journal of Computer Vision* 6(3), 227–243 (1991)
31. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *IEEE PAMI* 18, 377–388 (1996)
32. Gold, S., Rangarajan, A., Mjolsness, E.: Learning with preknowledge: Clustering with point and graph matching distance measures. In: *NIPS*, pp. 713–720 (1994)
33. Held, D., Levinson, J., Thrun, S., Savarese, S.: Combining 3d shape, color, and motion for robust anytime tracking. In: *RSS, Berkeley, USA* (July 2014)



34. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief networks. *Neural Computation* 18(7), 1527–1554 (2006)
35. Hong, B.W., Lu, Z., Sundaramoorthi, G.: A new model and simple algorithms for multi-label mumford-shah problems. In: *CVPR* (June 2013)
36. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: *CVPR*, pp. 2307–2314. *IEEE* (2013)
37. Kim, M., Torre, F.D.: Gaussian processes multiple instance learning, pp. 535–542 (2010)
38. Kosowsky, J.J., Yuille, A.L.: The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks* 7(3), 477–490 (1994)
39. Leich, A., Junghans, M., Jentschel, H.J.: Hough transform with GNC. 12th European Signal Processing Conference (EUSIPCO, 2004)
40. Leordeanu, M., Hebert, M.: Smoothing-based optimization. In: *CVPR* (2008)
41. Li, X.: Fine-granularity and spatially-adaptive regularization for projection-based image deblurring. *IEEE Transactions on Image Processing* 20(4), 971–983 (2011)
42. Liu, Z., Qiao, H., Xu, L.: An extended path following algorithm for graph-matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(7), 1451–1456 (2012)
43. Loog, M., Duistermaat, J.J., Florack, L.M.J.: On the behavior of spatial critical points under gaussian blurring (A folklore theorem and scale-space constraints). In: Kerckhove, M. (ed.) *Scale-Space 2001*. LNCS, vol. 2106, pp. 183–192. Springer, Heidelberg (2001)
44. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: *ICML*, vol. 382, p. 87. *ACM* (2009)
45. Malek-Mohammadi, M., Babaie-Zadeh, M., Amini, A., Jutten, C.: Recovery of low-rank matrices under affine constraints via a smoothed rank function. *IEEE Transactions on Signal Processing* 62(4), 981–992 (2014)
46. Mobahi, H., Rao, S., Ma, Y.: Data-driven image completion by image patch subspaces. In: *Picture Coding Symposium* (2009)
47. Mobahi, H., Ma, Y., Zitnick, L.: Seeing through the Blur. In: *CVPR* (2012)
48. Mohimani, G.H., Babaie-Zadeh, M., Gorodnitsky, I., Jutten, C.: Sparse recovery using smoothed  $\ell^0$  (sl0): Convergence analysis. *CoRR* abs/1001.5073 (2010)
49. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42(5), 577–685 (1989)
50. Nielsen, M.: Graduated non-convexity by smoothness focusing. In: *Proceedings of the British Machine Vision Conference*, pp. 60.1–60.10. *BMVA Press* (1993)
51. Nikolova, M., Ng, M.K., Tam, C.P.: Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. *Trans. Img. Proc.* 19(12), 3073–3088 (2010)
52. Pretto, A., Soatto, S., Menegatti, E.: Scalable dense large-scale mapping and navigation. In: *Proc. of: Workshop on Omnidirectional Robot Vision, ICRA* (2010)
53. Price, B.L., Morse, B.S., Cohen, S.: Simultaneous foreground, background, and alpha estimation for image matting. In: *CVPR*, pp. 2157–2164. *IEEE* (2010)
54. Rangarajan, A., Chellappa, R.: Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation, pp. II:127–II:133 (1990)
55. Rose, K., Gurewitz, E., Fox, G.: A deterministic annealing approach to clustering. *Pattern Recognition Letters* 11(9), 589–594 (1990)
56. Rossi, F., Villa-Vialaneix, N.: Optimizing an organized modularity measure for topographic graph clustering: A deterministic annealing approach. *Neurocomputing* 73(7-9), 1142–1163 (2010)
57. Saragih, J.: Deformable face alignment via local measurements and global constraints, pp. 187–207. Springer, Heidelberg (2013)

58. Shroff, N., Turaga, P.K., Chellappa, R.: Manifold precis: An annealing technique for diverse sampling of manifolds. In: NIPS, pp. 154–162 (2011)
59. Sindhvani, V., Keerthi, S.S., Chapelle, O.: Deterministic annealing for semi-supervised kernel machines. In: ICML 2006, pp. 841–848. ACM, New York (2006)
60. Smith, N.A., Eisner, J.: Annealing techniques for unsupervised statistical language learning. In: ACL, Barcelona, Spain, pp. 486–493 (July 2004)
61. Stoll, M., Volz, S., Bruhn, A.: Joint trilateral filtering for multiframe optical flow. In: ICIP, pp. 3845–3849 (2013)
62. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: CVPR, pp. 2432–2439. IEEE (2010)
63. Terzopoulos, D.: The computation of visible-surface representations. IEEE Trans. Pattern Anal. Mach. Intell. 10(4), 417–438 (1988)
64. Tirthapura, S., Sharvit, D., Klein, P., Kimia, B.: Indexing based on edit-distance matching of shape graphs. In: SPIE International Symposium on Voice, Video, and Data Communications, pp. 25–36 (1998)
65. Trzasko, J., Manduca, A.: Highly undersampled magnetic resonance image reconstruction via homotopic  $\ell_0$ -minimization. IEEE Trans. Med. Imaging 28(1), 106–121 (2009)
66. Vese, L.: A method to convexify functions via curve evolution. Commun. Partial Differ. Equations 24(9-10), 1573–1591 (1999)
67. Vural, E., Frossard, P.: Analysis of descent-based image registration. SIAM J. Imaging Sciences 6(4), 2310–2349 (2013)
68. Widder, D.V.: The Heat Equation. Academic Press (1975)
69. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. 31(2), 210–227 (2009)
70. Wu, Z., Tan, P.: Calibrating photometric stereo by holistic reflectance symmetry analysis, pp. 1498–1505. IEEE (2013)
71. Wu, Z.: The Effective Energy Transformation Scheme as a Special Continuation Approach to Global Optimization with Application to Molecular Conformation. SIAM J. on Optimization 6, 748–768 (1996)
72. Yuille, A.: Energy Functions for Early Vision and Analog Networks. A.I. memo, Defense Technical Information Center (1987)
73. Yuille, A.L.: Generalized deformable models, statistical physics, and matching problems. Neural Computation 2, 1–24 (1990)
74. Yuille, A., Geiger, D., Bulthoff, H.: Stereo integration, mean field theory and psychophysics. In: Faugeras, O. (ed.) ECCV 1990. LNCS, vol. 427, pp. 71–82. Springer, Heidelberg (1990)
75. Yuille, A.L., Peterson, C., Honda, K.: Deformable templates, robust statistics, and hough transforms, San Diego, CA, pp. 166–174. International Society for Optics and Photonics (1991)
76. Yuille, A.L., Stolorz, P.E., Utans, J.: Statistical physics, mixtures of distributions, and the em algorithm. Neural Computation 6(2), 334–340 (1994)
77. Zaslavskiy, M., Bach, F., Vert, J.P.: A path following algorithm for the graph matching problem. IEEE PAMI (2009)
78. Zerubia, J., Chellappa, R.: Mean field annealing using compound gauss-markov random fields for edge detection and image estimation. IEEE Transactions on Neural Networks 4(4), 703–709 (1993)

# How Hard Is the LP Relaxation of the Potts Min-Sum Labeling Problem?

Daniel Průša and Tomáš Werner

Czech Technical University, Faculty of Electrical Engineering  
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic  
{prusapa1,werner}@cmp.felk.cvut.cz

**Abstract.** An important subclass of the min-sum labeling problem (also known as discrete energy minimization or valued constraint satisfaction) is the pairwise min-sum problem with arbitrary unary costs and attractive Potts pairwise costs (also known as the uniform metric labeling problem). In analogy with our recent result, we show that solving the LP relaxation of the Potts min-sum problem is not significantly easier than that of the general min-sum problem and thus, in turn, the general linear program. This suggests that trying to find an efficient algorithm to solve the LP relaxation of the Potts min-sum problem has a fundamental limitation. Our constructions apply also to integral solutions, yielding novel reductions of the (non-relaxed) general min-sum problem to the Potts min-sum problem.

**Keywords:** Markov random field, discrete energy minimization, valued constraint satisfaction, linear programming relaxation, uniform metric labeling problem, Potts model.

## 1 Introduction

The *min-sum (labeling) problem*, also known as discrete energy minimization [15,5] or valued constraint satisfaction [16], has numerous applications in machine learning and computer vision and other fields, in particular as MAP inference in graphical models [17]. The problem has a natural LP relaxation [13,18,7,3,17], which underlies many algorithms to approximately solve the problem (see [5] and references therein). It is therefore of great practical importance to have efficient algorithms to solve this LP. Unfortunately, the simplex and interior point methods solving general LP are prohibitively inefficient for large-scale vision instances.

It is known that the LP relaxation of the pairwise min-sum problem with 2 labels reduces in linear time to max-flow [1,11]. Therefore, this problem can be solved very efficiently because the worst-case complexity of best known algorithms for max-flow is much better than for the general LP (though both are in the P complexity class). Our recent paper [10] showed that solving the LP relaxation of the pairwise min-sum problem with 3 or more labels (with some costs possibly infinite) is as hard as solving the general LP, precisely, the latter reduces to the former in linear time. This suggests that trying to find a very efficient algorithm to solve the LP relaxation may be futile.

This negative result raises the question whether there are any other useful subclasses of the min-sum problem for which the LP relaxation is significantly easier than the general linear program and therefore there is hope for efficient algorithms. In this paper,

we show that this is unlikely for the class of pairwise min-sum problems with attractive Potts costs, which is also known as the uniform metric labeling problem [2,6,3,4].

We present two efficient reductions of the general pairwise min-sum problem to the Potts min-sum problem that preserve the LP relaxation. The first one (§4, §5) reduces the general min-sum problem with some costs possibly infinite to the Potts min-sum problem with 3 labels (the complexity of this reduction is given by Theorems 5 and 8). Combined with [10], this implies that solving the general system of linear inequalities reduces in linear time to the LP relaxation of the Potts min-sum problem with 3 labels (Corollary 6, our most surprising result) and that the general linear program reduces in better than quadratic time to the LP relaxation of the Potts min-sum problem with 3 labels (Corollary 9). The second one (§6) reduces the general min-sum problem with  $k$  labels and finite costs to the Potts min-sum problem with  $k$  labels (Theorem 11). The output costs in this reduction are typically much smaller than in the first reduction.

Though these results are somewhat weaker than for the general min-sum problem [10], they are far from obvious. They show that finding an efficient algorithm to solve the LP relaxation of the Potts min-sum problem is unlikely because this might mean improving the complexity of the best known algorithms for the general LP. An example of an algorithm specialized to the LP relaxation of the Potts min-sum problem is [9].

Our reductions straightforwardly apply also to the original non-relaxed min-sum problems, thus we obtain as side-results novel reductions from the general min-sum problem to the Potts one (Theorems 4, 7, and 10). These results can be seen analogical to, e.g., the construction [12] which reduces the general pairwise min-sum problem with finite costs to the pairwise min-sum problem with 2 labels.

## 2 Min-sum Problem and Its LP Relaxation

We denote  $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$  and  $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{\infty\}$ . Let  $(V, E)$  be a connected undirected graph, with *objects*  $V$  and *object pairs*  $E \subseteq \binom{V}{2}$ . Let  $K$  be a finite set of *labels*. Let  $g_u: K \rightarrow \overline{\mathbb{Q}}$  and  $g_{uv}: K \times K \rightarrow \overline{\mathbb{Q}}$  be unary and pairwise *cost functions*, where we adopt that  $g_{uv}(k, \ell) = g_{vu}(\ell, k)$ . The pairwise *min-sum problem* is defined as

$$\min_{\mathbf{k} \in K^V} \left( \sum_{v \in V} g_u(k_u) + \sum_{\{u, v\} \in E} g_{uv}(k_u, k_v) \right). \quad (1)$$

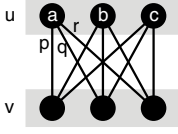
All the costs  $g_u(k)$  and  $g_{uv}(k, \ell)$  together will be understood as a vector  $\mathbf{g} \in \overline{\mathbb{Q}}^I$  with  $I = (V \times K) \cup \{ \{ (u, k), (v, \ell) \} \mid \{u, v\} \in E; k, \ell \in K \}$ .

The *local marginal polytope* [17] is the set  $\Lambda$  of vectors  $\boldsymbol{\mu} \in \mathbb{R}_+^I$  satisfying

$$\sum_{k \in K} \mu_u(k) = 1, \quad u \in V \quad (2a)$$

$$\sum_{\ell \in K} \mu_{uv}(k, \ell) = \mu_u(k), \quad u \in V, v \in N_u, k \in K \quad (2b)$$

where  $N_u = \{v \mid \{u, v\} \in E\}$  are the neighbors of object  $u$  and we again adopt that  $\mu_{uv}(k, \ell) = \mu_{vu}(\ell, k)$ . The numbers  $\mu_u(k), \mu_{uv}(k, \ell)$  are known as *pseudomarginals* [17]. Figure 1 illustrates the meaning of constraints (2) for one object pair.



**Fig. 1.** Two objects forming an object pair  $\{u, v\} \in E$ . Objects  $u \in V$  are depicted as boxes, labels  $(u, k)$  as nodes, and label pairs  $\{(u, k), (v, \ell)\}$  as edges. Note the meaning of constraints (2): for unary pseudomarginals  $a, b, c$  and pairwise pseudomarginals  $p, q, r$ , (2b) enforces  $a = p + q + r$  and (2a) enforces  $a + b + c = 1$ .

The LP relaxation of problem (1) reads

$$\min\{\mathbf{g}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \Lambda\} \quad (3)$$

where, if some costs (components of  $\mathbf{g}$ ) are infinite, we define  $0 \cdot \infty = 0$  in the scalar product  $\mathbf{g}^\top \boldsymbol{\mu}$ . If  $\boldsymbol{\mu} \in \{0, 1\}^I$  then (3) solves (1) exactly.

Reparameterizations of a vector  $\mathbf{g} \in \overline{\mathbb{Q}}^I$  is a vector  $\mathbf{g}' \in \overline{\mathbb{Q}}^I$  given by

$$g'_u(k) = g_u(k) - \sum_{v \in N_u} \varphi_{uv}(k) \quad (4a)$$

$$g'_{uv}(k, \ell) = g_{uv}(k, \ell) + \varphi_{uv}(k) + \varphi_{vu}(\ell) \quad (4b)$$

where  $\varphi = (\varphi_{uv}(k) \in \mathbb{R} : u \in V, v \in N_u, k \in K)$ . We have  $\mathbf{g}^\top \boldsymbol{\mu} = \mathbf{g}'^\top \boldsymbol{\mu}$  for every  $\varphi$  and every  $\boldsymbol{\mu}$  satisfying (2), thus reparameterizations preserve the objective of (1) and its LP relaxation. Consider a *lower bound*

$$L(\mathbf{g}) = \sum_{u \in V} \min_{k \in K} g_u(k) + \sum_{\{u, v\} \in E} \min_{k, \ell \in K} g_{uv}(k, \ell) \quad (5)$$

on the true optimal value (1). The dual to the LP (3) can be expressed [18] as maximizing the lower bound over reparameterizations, i.e., maximizing  $L(\mathbf{g}')$  over  $\varphi$ .

If the pairwise cost functions  $g_{uv}$  in (1) are metric while the unary cost functions  $g_u$  remains arbitrary, the problem (1) has been called the *metric labeling problem* [2,6,3,4]. Its special case is the *uniform metric* or the *attractive Potts interaction*

$$g_{uv}(k, \ell) = h_{uv} \llbracket k \neq \ell \rrbracket \quad (6)$$

where  $h_{uv} \geq 0$  and  $\llbracket k \neq \ell \rrbracket$  equals 1 if  $k \neq \ell$  and 0 otherwise. We will refer to problem (1) with pairwise costs (6) as the *Potts min-sum problem*.

### 3 Summary of Results

This section gives the overview of our contributions in this paper, after formulating previous closely related results that we obtained in [10].

As is usual in computational complexity, we will use the notions of problem (a set of instances), instance, and reduction. We start this section by defining the following problems, by specifying their instances (inputs) and solutions (outputs). Rather than more common decision problems, which map strings over an alphabet to the answers {yes, no}, we formulate our problems as *function problems*, which map strings over an alphabet to strings over an alphabet.

**Problem:** MINSUM( $Y$ ) where  $Y \subseteq \overline{\mathbb{Q}}$

**Instance:**  $(V, E, K, \mathbf{g})$  where  $\mathbf{g} \in Y^I$ . (Thus,  $Y$  specifies the set of values the costs can take. E.g., in MINSUM( $\mathbb{Z}$ ) the costs can take values from  $\mathbb{Z}$  rather than from  $\overline{\mathbb{Q}}$ .)

**Solution:** If the optimal value of problem (1) is finite, it returns an optimal argument  $\mathbf{k} \in K^V$ . Otherwise, it answers 'infeasible'.

**Problem:** MINSUM( $Y$ )-LP

**Instance:**  $(V, E, K, \mathbf{g})$  where  $\mathbf{g} \in Y^I$  and  $Y \subseteq \overline{\mathbb{Q}}$ .

**Solution:** If the LP (3) is feasible, it returns an optimal argument  $\boldsymbol{\mu} \in [0, 1]^I$ . If (3) is infeasible, it answers 'infeasible'.

**Problem:** POTTS

**Instance:**  $(V, E, K, \mathbf{g})$  where  $\mathbf{g} \in \mathbb{Q}^I$  and pairwise costs in  $\mathbf{g}$  have the form (6).

**Solution:** An optimal argument  $\mathbf{k} \in K^V$  of problem (1).

**Problem:** POTTS-LP

**Instance:**  $(V, E, K, \mathbf{g})$  where  $\mathbf{g} \in \mathbb{Q}^I$  and pairwise costs in  $\mathbf{g}$  have the form (6).

**Solution:** An optimal argument  $\boldsymbol{\mu} \in [0, 1]^I$  of problem (3).

**Problem:** LININEQ

**Instance:**  $(\mathbf{A}, \mathbf{b})$  where  $\mathbf{A} \in \mathbb{Z}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{Z}^m$ .

**Solution:** If the system  $\{\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  has a solution, it returns its arbitrary solution. Otherwise, it answers 'infeasible'.

**Problem:** LINPROG

**Instance:**  $(\mathbf{A}, \mathbf{b}, \mathbf{c})$  where  $\mathbf{A} \in \mathbb{Z}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{Z}^m$ ,  $\mathbf{c} \in \mathbb{Z}^n$ .

**Solution:** If the linear program  $\min\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  is feasible and bounded, it returns a solution  $\mathbf{x} \in \mathbb{Q}^n$ . If the program is infeasible, it answers 'infeasible'. If the program is unbounded, it answers 'unbounded'.

**Instance Sizes.** In general, the size of a problem instance is the length of the (binary) string needed to encode it. We will use  $\langle x \rangle$  to denote the size of a number  $x \in \mathbb{Z}$ . Using one bit for the sign, storing  $x$  takes  $\langle x \rangle = \lceil \log_2(|x| + 1) \rceil + 1$  bits. For a vector  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{Z}^n$ , we define its size to be  $\langle \mathbf{x} \rangle = \sum_{i=1}^n \langle x_i \rangle$ . We will use this definition of size for vectors  $\mathbf{g}$  and  $\mathbf{c}$ .

For  $\mathbf{A}$  and  $\mathbf{b}$  we use a slightly different definition of size. The pair  $(\mathbf{A}, \mathbf{b})$  can be seen as the extended matrix  $\bar{\mathbf{A}} = [\mathbf{A} \mid \mathbf{b}] \in \mathbb{Z}^{m \times (n+1)}$ . Encoding  $\bar{\mathbf{A}}$  by storing all its entries (including zeros) would take  $L = \sum_{i=1}^m \sum_{j=1}^{n+1} \langle a_{ij} \rangle$  bits. This would describe the dense encoding of  $\bar{\mathbf{A}}$ . However, we define

$$\langle \bar{\mathbf{A}} \rangle = \sum_{i=1}^m \sum_{j=1}^{n+1} \lceil \log_2(|\bar{a}_{ij}| + 1) \rceil. \quad (7)$$

As zero entries  $\bar{a}_{ij} = 0$  do not contribute to  $\langle \bar{\mathbf{A}} \rangle$ , this describes a *sparse* encoding of  $\bar{\mathbf{A}}$ . Note that  $\langle \bar{\mathbf{A}} \rangle \leq L$ , therefore (7) covers both sparse and dense encoding because  $\bar{\mathbf{A}}$  will always describe input (never output) instances of our reductions. Indeed, for every  $f: \mathbb{N} \rightarrow \mathbb{N}$ , if the complexity of a reduction is  $\mathcal{O}(f(\langle \bar{\mathbf{A}} \rangle))$  then it is also  $\mathcal{O}(f(L))$ .

For convenience, we defined the instance of POTS and POTS-LP the same way as for  $\text{MINSUM}(Y)$ -LP and  $\text{MINSUM}(Y)$ , namely by the tuple  $(V, E, K, \mathbf{g})$  with  $\mathbf{g} \in \mathbb{Q}^I$ . However, the components of  $\mathbf{g}$  are not independent since they satisfy (6). This must be taken into account when computing  $\langle \mathbf{g} \rangle$  for POTS and POTS-LP.

**Existing Results.** The results obtained in [10] can be formulated as follows.

**Theorem 1.** *LINPROG reduces in linear time to  $\text{MINSUM}(\overline{\mathbb{Z}})$ -LP with 3 labels.*

**Theorem 2.** *LINPROG reduces in quadratic time to  $\text{MINSUM}(\mathbb{Z})$ -LP with 3 labels.*

**Theorem 3.** *LININEQ reduces in linear time to  $\text{MINSUM}(\{0, \infty\})$ -LP with 3 labels.*

Theorem 3 is not explicitly stated in [10]. It holds because LININEQ is LINPROG with  $\mathbf{c} = \mathbf{0}$ , in which case the output min-sum problem has costs in  $\{0, \infty\}$  [10, §5].

**Contributions.** Our contributions in this paper are two reductions of the general min-sum problem to the Potts min-sum problem that preserve both the optimum of (1) and the optimum of its LP relaxation (3). These reductions lead to the following results.

**Theorem 4.**  *$\text{MINSUM}(\{0, \infty\})$  reduces in linear time to POTS with 3 labels.*

**Theorem 5.**  *$\text{MINSUM}(\{0, \infty\})$ -LP reduces in linear time to POTS-LP with 3 labels.*

**Corollary 6.** *LININEQ reduces in linear time to POTS-LP with 3 labels.*

*Proof.* Combine Theorem 3 and Theorem 5. □

**Theorem 7.**  *$\text{MINSUM}(\overline{\mathbb{Z}})$  with  $p$  object pairs,  $k$  labels and size  $L$  reduces in time  $\mathcal{O}(pk^2L)$  to POTS with 3 labels.*

**Theorem 8.**  *$\text{MINSUM}(\overline{\mathbb{Z}})$ -LP with  $p$  object pairs,  $k$  labels and size  $L$  reduces in time  $\mathcal{O}(pk^2L)$  to POTS-LP with 3 labels.*

**Corollary 9.** *LINPROG reduces in quadratic time to POTS-LP with 3 labels.*

*Proof.* By Theorem 8,  $\text{MINSUM}(\overline{\mathbb{Z}})$ -LP reduces in quadratic time to POTS-LP with 3 labels, because  $pk^2 = \mathcal{O}(L)$  and so  $\mathcal{O}(pk^2L) \subseteq \mathcal{O}(L^2)$ . This is combined with Theorem 1. □

**Theorem 10.**  *$\text{MINSUM}(\mathbb{Z})$  with  $k$  labels and size  $L$  reduces in time  $\mathcal{O}(k^2L)$  to POTS with  $k$  labels.*

**Theorem 11.**  *$\text{MINSUM}(\mathbb{Z})$ -LP with  $k$  labels and size  $L$  reduces in time  $\mathcal{O}(k^2L)$  to POTS-LP with  $k$  labels.*

In §4 we will describe our first reduction for input costs in  $\{0, \infty\}$  and thereby prove Theorems 4 and 5. In §5 we generalize this to arbitrary costs, proving thus Theorems 7 and 8. In §6, we describe our second reduction and prove Theorems 10 and 11.

## 4 Encoding a Local Marginal Polytope

Consider the polyhedron

$$P = \{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \} \quad (8)$$

where  $\mathbf{A} \in \{-1, 0, 1\}^{m \times n}$  and  $\mathbf{b} \in \{0, 1\}^m$  satisfy the following conditions:

(P1)  $P \subseteq [0, 1]^n$ .

(P2) Each row of the matrix  $[-\mathbf{A} \mid \mathbf{b}]$  contains exactly one positive number.

(P3) Each row of  $\mathbf{A}$  contains at most  $d$  non-zeros.

(P4)  $\mathbf{A}$  has in total  $\mathcal{O}(n)$  non-zeros.

Every local marginal polytope with  $k$  labels and  $p$  object pairs has this form, where  $m = \mathcal{O}(kp)$ ,  $n = \mathcal{O}(k^2p)$ ,  $d = k + 1$ . Moreover,  $\text{MINSUM}(\{0, \infty\})$ -LP is equivalent to the problem that decides whether  $P$  is non-empty and if so, it finds an  $\mathbf{x} \in P$ .

In this section, we prove Theorems 4 and 5 by constructing, from the input polytope (8), the output Potts min-sum problem. More precisely, we construct a *reparameterized* Potts min-sum problem  $(V, E, K, \mathbf{g})$ , whose costs will satisfy

$$g_u(k) = 0, \quad \forall u \in V, \forall k \in K \quad (9a)$$

$$g_{uv}(k, \ell) = 2\llbracket k \neq \ell \rrbracket + \varphi_{uv}(k) + \varphi_{vu}(\ell), \quad \forall \{u, v\} \in E; \forall k, \ell \in K \quad (9b)$$

$$\min_{k, \ell \in K} g_{uv}(k, \ell) = 0, \quad \forall \{u, v\} \in E \quad (9c)$$

Note that (9) implies  $L(\mathbf{g}) = 0$ . By complementary slackness, any  $\boldsymbol{\mu} \in \Lambda$  and any  $\mathbf{g}$  of the form (9) are simultaneously optimal to (3) and its dual if and only if

$$g_{uv}(k, \ell) \mu_{uv}(k, \ell) = 0, \quad \forall \{u, v\} \in E; \forall k, \ell \in K. \quad (10)$$

Moreover, the output min-sum problem will be designed such that if  $P \neq \emptyset$  then  $\mathbf{g}$  is dual-optimal, i.e.,  $\min\{\mathbf{g}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \Lambda\} = L(\mathbf{g}) = 0$ .

We will depict min-sum problems by diagrams, as in Figure 1. In addition, we adopt the following conventions: non-zero values of  $\varphi_{uv}(k)$  are written next to node  $(u, k)$  on the side of object  $v \in N_u$ , and an edge  $\{(u, k), (v, \ell)\}$  is visible if  $g_{uv}(k, \ell) = 0$  and invisible if  $g_{uv}(k, \ell) > 0$ . Assuming  $P \neq \emptyset$ , (10) thus says that  $\boldsymbol{\mu} \in \Lambda$  is optimal to (3) if and only if pairwise pseudomarginals are zero on invisible edges.

### 4.1 Elementary Constructions

Similarly as in [10], we will construct the output min-sum problem by gluing certain smaller problems, each of them encoding a simple operation. We refer to these small problems as *elementary constructions*. Each elementary construction is a standalone min-sum problem whose costs  $\mathbf{g}$  satisfy (9) and are optimal to the dual LP.

We will use the following elementary constructions (see Figure 2):

- SWAP encodes a swap of two unary pseudomarginals, one of them zero. More precisely, the LP relaxation (3) of this min-sum problem achieves its optimal value (zero) if and only if the unary pseudomarginals linked by visible edges are equal and the unary pseudomarginals in the crossed-out labels are zero.



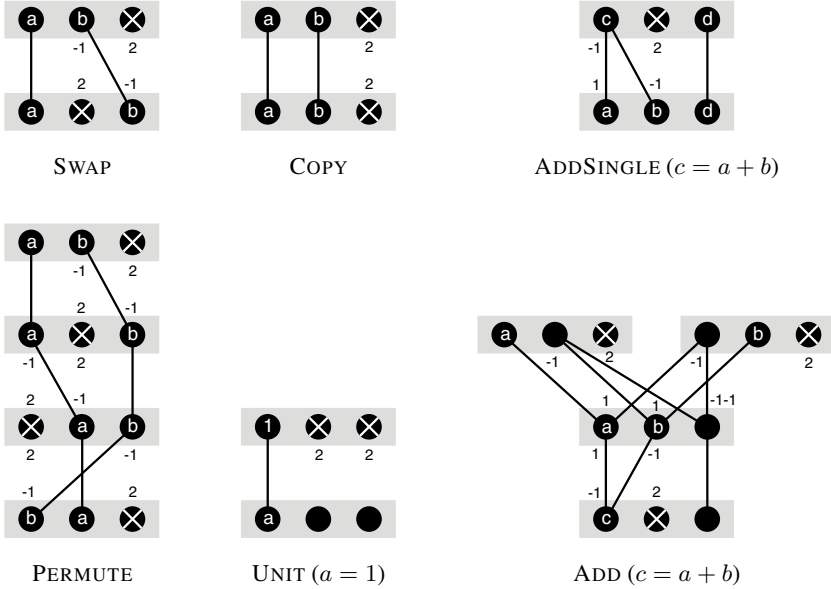


Fig. 2. Elementary constructions

- PERMUTE applies SWAP several times to arbitrarily permute all the three unary pseudomarginals, one of them zero. The figure shows one possible permutation.
- COPY copies all the three unary pseudomarginals, one of them zero, from one object to another object.
- UNIT enforces the value of a unary pseudomarginal to be 1. The other two unary pseudomarginals are necessarily zero.
- ADDSINGLE adds two unary pseudomarginals in a single object and copies the result in another object. The third (possibly nonzero) unary pseudomarginal is copied.
- ADD adds two unary pseudomarginals in two different objects. This is done by gluing three ADDSINGLE constructions.

For each elementary constructions (considered as a standalone min-sum problem), the LP relaxation is tight, i.e., the optimal values of (3) and (1) coincide.

## 4.2 The Encoding Algorithm

Using the elementary constructions, we now describe the algorithm to construct the output min-sum problem  $(V, E, K, \mathbf{g})$  from the polytope  $P$ . First, we rewrite the system  $\mathbf{Ax} = \mathbf{b}$  by moving negative terms to the right-hand side as

$$a_{i1}^+ x_1 + \dots + a_{in}^+ x_n = a_{i1}^- x_1 + \dots + a_{in}^- x_n + b_i, \quad i = 1, \dots, m \quad (11)$$

where  $a_{ij}^+, a_{ij}^- \in \{0, 1\}$  and  $a_{ij} = a_{ij}^+ - a_{ij}^-$ . Note that condition (P2) says that the RHS of (11) has exactly one non-zero term. This in turn ensures that both sides of (11) are

not greater than 1 for every  $\mathbf{x} \in P$ , thus all intermediate results are representable by pseudomarginals. We denote the labels as  $K = \{1, 2, 3\}$ .

The algorithm is initialized by setting  $V = \{1, \dots, n\}$  and  $E = \emptyset$ . Each variable  $x_j$  in (8) will be represented by unary pseudomarginal  $\mu_j(1)$ . Then each equation (11) is encoded one after another. The  $i$ -th equation is encoded as follows:

1. Construct a unary pseudomarginal with the value equal to the LHS of (11). This is done by repeatedly applying ADD, possibly permuting labels by PERMUTE.
2. Construct a unary pseudomarginal with value equal to the RHS of (11). Recall that the RHS of (11) has exactly one non-zero term. If  $a_{ij}^- = 1$  for some  $j$  and  $b_i = 0$ , we already have the desired pseudomarginal, namely  $\mu_j(1)$ . If  $a_{ij} = 0$  for all  $j$  and  $b_i = 1$ , we prepare a pseudomarginal with value  $b_i = 1$  using UNIT.
3. Enforce equality of both sides of (11) using COPY, permuting labels when necessary by PERMUTE.

Figure 3 shows the constructed min-sum problem for an example polytope  $P$ . By construction, the output min-sum problem has the following properties:

- If  $P \neq \emptyset$  then  $\min\{\mathbf{g}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \Lambda\} = 0$ . For every  $\boldsymbol{\mu}$  optimal to this problem, we have  $\mathbf{x} = (\mu_1(1), \dots, \mu_n(1)) \in P$ .
- If  $P \cap \{0, 1\}^n \neq \emptyset$  then  $\min\{\mathbf{g}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \Lambda \cap \{0, 1\}^n\} = 0$ . For every  $\boldsymbol{\mu}$  optimal to this problem, we have  $\mathbf{x} = (\mu_1(1), \dots, \mu_n(1)) \in P \cap \{0, 1\}^n$ .
- If  $P = \emptyset$  then  $\min\{\mathbf{g}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \Lambda\} > 0$ .

This proves Theorems 4 and 5, up to complexity.

### 4.3 Complexity of Encoding

Let us count the number of objects and object pairs in the output min-sum problem. Since for each elementary construction we have  $|E| = \mathcal{O}(|V|)$  and the output problem is constructed by gluing elementary constructions, we have  $|E| = \mathcal{O}(|V|)$ . The variables  $x_1, \dots, x_n$  are represented by  $n$  objects. Each equation (11) is represented by  $\mathcal{O}(d)$  objects. It follows from conditions (P3) and (P4) that  $n = \mathcal{O}(dm)$ . Thus, the total number of objects is  $\mathcal{O}(n + dm) = \mathcal{O}(n)$ . The time complexity of the algorithm is proportional to  $|V|$ , thus it is also  $\mathcal{O}(n)$ .

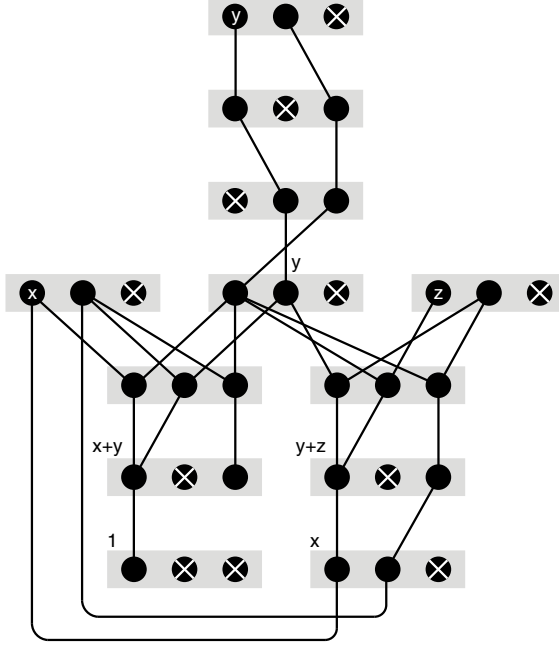
## 5 Encoding a Min-sum Problem

In this section, we show that any (integer) linear optimization over polyhedron (8),

$$\min\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{x} \in P \cap \{0, 1\}^n\}, \quad (12a)$$

$$\min\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{x} \in P\}, \quad (12b)$$

can be efficiently reduced to the Potts min-sum problem with 3 labels. Since every local marginal polytope has the form (8), this will prove Theorems 7 and 8.



**Fig. 3.** The constructed reparameterized Potts min-sum problem that encodes the polytope  $P = \{(x, y, z) \in [0, 1]^3 \mid x + y = 1, y + z = x\}$ . The labels representing variables  $x, y, z$  have the variables written in them in white. The messages  $\varphi_{uv}(k)$  are not shown, they are like in Figure 2.

The input of the reduction is a triplet  $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ , where  $(\mathbf{A}, \mathbf{b}) = \bar{\mathbf{A}}$  describes  $P$ . The output is a min-sum problem  $(V, E, K, \mathbf{g})$ , constructed as follows. First we construct min-sum problem  $(V, E, K, \mathbf{g}')$  according to §4.2. Then we set  $\mathbf{g} \in \mathbb{Z}^I$  as

$$g_j(k) = \begin{cases} c_j & \text{if } k = 1 \text{ and } j \leq n \\ 0 & \text{otherwise} \end{cases} \quad (13a)$$

$$g_{ij}(k, \ell) = M g'_{ij}(k, \ell) \quad (13b)$$

where  $M \in \mathbb{N}$  is a big enough number (derived below) to ensure that every optimal  $\boldsymbol{\mu} \in [0, 1]^I$  and every integer optimal  $\boldsymbol{\mu} \in \{0, 1\}^I$  to the output problem satisfies (10).

We first derive  $M$  for the simpler case, the ILP (12a). It suffices to set

$$M = C_u - C_\ell + 1 \quad (14)$$

where

$$C_\ell = \sum_{j=1}^n \min\{0, c_j\}, \quad C_u = \sum_{j=1}^n \max\{0, c_j\} \quad (15)$$

is a lower and upper bound, respectively, on the optimal value of (12a).

Let us prove that every optimal solution  $\boldsymbol{\mu}$  of (12a) satisfies (10). The smallest non-zero pairwise cost  $g'_{uv}(k, \ell)$  is 1, thus the smallest non-zero  $g_{uv}(k, \ell)$  is  $M$ . Assume

that for some  $\{u, v\} \in E$  and  $k, \ell \in K$  we have  $g_{uv}(k, \ell) > 0$  and  $\mu_{uv}(k, \ell) = 1$ . Then  $\mathbf{g}^\top \boldsymbol{\mu} \geq M + C_\ell > C_u$ , which is a contradiction.

Let us derive the complexity of the reduction. We have  $\langle M \rangle = \mathcal{O}(\langle \mathbf{c} \rangle)$ , because we must consider the worst case when the sizes of  $c_1, \dots, c_n$  are very unequally distributed, e.g.,  $\langle c_1 \rangle = \mathcal{O}(\langle \mathbf{c} \rangle)$ . Each unary cost  $g_u(k)$  is a sum of at most  $|V|$  values not greater than  $2M$ , hence  $\langle g_u(k) \rangle = \mathcal{O}(\langle \mathbf{c} \rangle + \log |V|) = \mathcal{O}(\langle \mathbf{c} \rangle)$ . Thus the description length of the output problem is<sup>1</sup>  $\mathcal{O}(n \langle \mathbf{c} \rangle)$ . This concludes the proof of Theorem 7.

We now derive  $M$  for the more difficult case, the LP (12b). We first need a lemma.

**Lemma 12.** *Let  $(x_1, \dots, x_n)$  be a vertex of the polytope  $P$  defined by (8). For every  $j = 1, \dots, n$  we have  $x_j = 0$  or  $x_j \geq (d + 1)^{-m/2}$ .*

*Proof.* The proof is analogical to that of [10, Lemma 7].

At least one optimal solution to (12b) is attained at a vertex of  $P$ . The coordinates of a vertex are fractions, however, by Lemma 12, each non-zero coordinate is not less than  $(d + 1)^{-m/2}$ . This means it suffices to choose

$$M = (C_u - C_\ell)(d + 1)^{m/2} + 1. \quad (16)$$

In the worst case,  $\langle M \rangle = \mathcal{O}(\langle \mathbf{c} \rangle + m \log d) = \mathcal{O}(\langle \mathbf{c} \rangle)$ . This proves the claimed complexity  $\mathcal{O}(n \langle \mathbf{c} \rangle)$  and thus concludes the proof of Theorem 8. Note that while the number (16) is much larger than (14), asymptotically they have the same bit size.

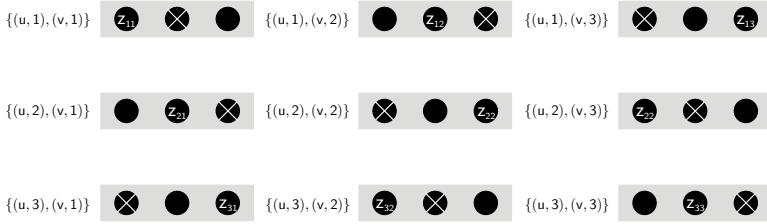
## 6 Direct Encoding of a Min-sum Problem

The reduction described in §5 involves large output costs (14) and (16), which makes it impractical and affects its theoretical complexity. Here we present a more direct reduction, which does not produce large output costs but applies only to finite input costs. By that, we prove Theorems 10 and 11.

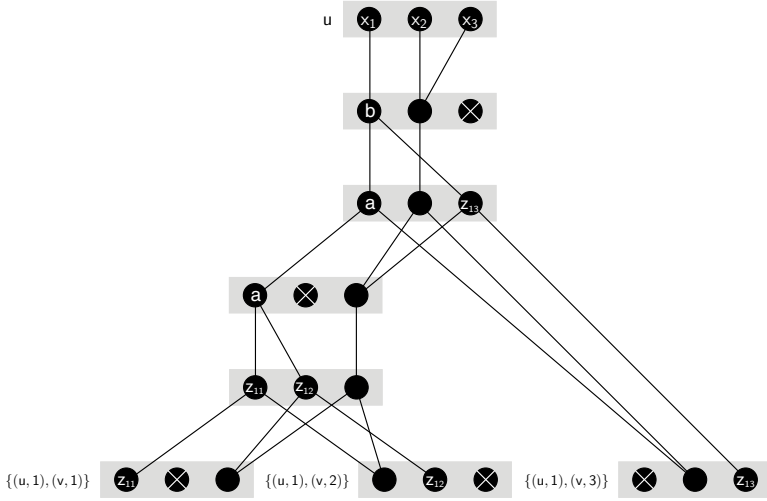
We construct a reparameterized Potts min-sum problem  $(V', E', K, \mathbf{g}')$  that encodes an input min-sum problem  $(V, E, K, \mathbf{g})$ . Note that both problems have the same label set. Each object  $u \in V$  of the input problem is represented by one object of the output problem, so that  $V \subseteq V'$ . Precisely, the unary pseudomarginals of the input problem are represented by unary pseudomarginals in objects  $V$  in the output problem, which automatically enforces normalization constraints (2a). Similarly, the unary costs of the input problem are copied to unary costs in objects  $V$  of the output problem.

Each object pair  $\{u, v\} \in E$  of the input problem is replaced by the following construction (see Figure 4). For each input label pair  $\{(u, k), (v, \ell)\}$  we introduce a new object  $\{(u, k), (v, \ell)\}$  into  $V'$ . One selected label in object  $\{(u, k), (v, \ell)\} \in V'$  of the output problem represents the label pair  $\{(u, k), (v, \ell)\}$  of the input problem, such that

<sup>1</sup> The derived complexity can be improved if some additional knowledge is available. First, we may obtain better bounds on the optimal value of (12a) than (15). E.g., if a feasible solution  $\mathbf{x}$  to (12a) can be obtained cheaply, it yields an upper bound  $\mathbf{c}^\top \mathbf{x} \leq C_u$ . Second,  $\langle M \rangle = \mathcal{O}(\langle \mathbf{c} \rangle)$  holds in the unfavorable case when the distribution of the sizes  $\langle c_i \rangle$  is very non-uniform. Under some additional assumptions on  $\mathbf{c}$ , this worst-case bound can be made much smaller. Assume, e.g., that  $\langle c_i \rangle \leq 2 \langle \mathbf{c} \rangle / n$  for every  $i$ . Then  $M \leq n 2^{2 \langle \mathbf{c} \rangle / n}$  and  $\langle M \rangle = \mathcal{O}(\langle \mathbf{c} \rangle / n + \log n)$ . Thus the description length of the output problem would be only  $\mathcal{O}(\langle \mathbf{c} \rangle + n \log n)$ .



**Fig. 4.** Objects added to  $V'$  for one input object pair  $\{u, v\} \in E$  and  $|K| = 3$ . For brevity,  $\mu_{uv}(k, \ell)$  is denoted by  $z_{k\ell}$ .



**Fig. 5.** The ADDK elementary construction, enforcing  $z_{11} + z_{12} + z_{13} = x_1$ . Note that  $a = z_{11} + z_{12}$  and  $b = a + z_{13}$ . For brevity,  $\mu_u(k)$  is denoted by  $x_k$ .

the unary pseudomarginal of this label represents the pseudomarginal  $\mu_{uv}(k, \ell)$  of the input problem and the unary cost of this label equals the input cost  $g_{uv}(k, \ell)$ .

Each marginalization constraint (2b) is encoded by the ADDK construction, shown in Figure 5 for  $|K| = 3$  labels. It is built from several constructions ADDSINGLE and ADD. For brevity, we denote  $\mu_u(k)$  and  $\mu_{uv}(k, \ell)$  by  $x_k$  and  $z_{k\ell}$ , respectively. The LP relaxation of ADDK attains zero optimal value if and only if  $z_{k1} + z_{k2} + z_{k3} = x_k$ , i.e., if and only if the marginalization constraint is satisfied.

Let  $f(z_{k1}, z_{k2}, z_{k3}, x_k)$  denote the optimal value of the LP relaxation of ADDK subject to the constraint that the unary pseudomarginals  $z_{k1}, z_{k2}, z_{k3}, x_k$  are fixed. As we said, if  $z_{k1} + z_{k2} + z_{k3} = x_k$  then  $f(z_{k1}, z_{k2}, z_{k3}, x_k) = 0$ . Otherwise, one can show<sup>2</sup> that there is a small constant  $C \in \mathbb{N}$  such that

$$Cf(z_{k1}, z_{k2}, z_{k3}, x_k) \geq |z_{k1} + z_{k2} + z_{k3} - x_k|. \tag{17}$$

<sup>2</sup> We omit the proof, which is long. For illustration, we state the similar claim for the ADDSINGLE construction (see Figure 2). Denoting by  $f(a, b, c)$  the optimal value of the LP relaxation of ADDSINGLE subject to fixed  $a, b, c$ , it is easy to show that  $f(a, b, c) \geq |a + b - c|$ .

It is straightforward to generalize the ADDK construction to  $|K| \geq 3$ , e.g., by using more objects and adding  $|K| - 3$  dummy labels to each object. Then we can write (17) as  $R_{uv}(k) \geq |r_{uv}(k)|$  where

$$r_{uv}(k) = \sum_{\ell \in K} \mu_{uv}(k, \ell) - \mu_u(k). \quad (18)$$

Let us multiply all pairwise costs in each ADDK construction by  $CM_{uv}$ , where  $M_{uv} \in \mathbb{N}$ . Then the LP relaxation of the output min-sum problem can be written as

$$\min \left\{ \mathbf{g}^\top \boldsymbol{\mu} + \sum_{u \in V} \sum_{v \in N_u} \sum_{k \in K} M_{uv} R_{uv}(k) \mid \boldsymbol{\mu} \in \mathbb{R}_+^I, \boldsymbol{\mu} \text{ satisfies (2a)} \right\}. \quad (19)$$

The numbers  $M_{uv}$  ( $u \in V, v \in N_u$ ) must be big enough to ensure that for every  $\boldsymbol{\mu}$  optimal to (19) all the residuals  $r_{uv}(k)$  vanish. It suffices to set

$$M_{uv} = M_{vu} = \left\lceil \frac{1}{2} \max_{k, \ell \in K} g_{uv}(k, \ell) \right\rceil + 1. \quad (20)$$

To prove this, observe that if unary pseudomarginals  $\mu_u$  are fixed, one can optimize over pairwise pseudomarginals  $\mu_{uv}$  separately for each  $\{u, v\} \in E$ . The rest follows from Proposition 13.

**Proposition 13.** *Consider a single pair  $\{u, v\} \in E$ . Let functions  $\mu_u, \mu_v: K \rightarrow \mathbb{R}_+$  satisfy (2a). Let  $g_{uv}: K \times K \rightarrow \mathbb{R}_+$ . Every optimal  $\mu_{uv}$  in the problem*

$$\min_{\mu_{uv}: K \times K \rightarrow [0,1]} \left( \sum_{k, \ell \in K} g_{uv}(k, \ell) \mu_{uv}(k, \ell) + \sum_{k \in K} M_{uv} (|r_{uv}(k)| + |r_{vu}(k)|) \right) \quad (21)$$

satisfies  $r_{uv}(k) = r_{vu}(k) = 0$  for all  $k \in K$ .

*Proof.* Suppose that some of the numbers  $r_{uv}(\cdot), r_{vu}(\cdot)$  are non-zero. We will show that then  $\boldsymbol{\mu}$  cannot be optimal to (21). Since  $\sum_k r_{uv}(k) = \sum_\ell r_{vu}(\ell)$ , at least one of the following cases must occur. For each case, we show that by changing  $\mu_{uv}$  (but keeping them feasible) the objective of (21) can be decreased:

1.  $r_{uv}(k) > 0$  for some  $k, r_{uv}(k') < 0$  for some  $k', r_{uv}(\ell) = 0$  for all  $\ell$ :  
Pick any  $\ell$  such that  $\mu_{uv}(k, \ell) > 0$ . Because  $r_{uv}(k) > 0$  and  $r_{uv}(\ell) = 0$ , we have  $\mu_{uv}(k', \ell) < 1$ . Decrease  $\mu_{uv}(k, \ell)$  by a small  $\delta > 0$  and increase  $\mu_{uv}(k', \ell)$  by the same  $\delta$ . This changes the objective by  $g_{uv}(k', \ell)\delta - g_{uv}(k, \ell)\delta - 2M_{uv}\delta < 0$ .
2.  $r_{uv}(k) < 0$  for some  $k, r_{vu}(\ell) < 0$  for some  $\ell$ :  
Because  $r_{uv}(k) < 0$ , we have  $\mu_{uv}(k, \ell) < 1$ . Increase  $\mu_{uv}(k, \ell)$  by a small  $\delta > 0$ . This changes the objective by  $g_{uv}(k, \ell)\delta - 2M_{uv}\delta < 0$ .
3.  $r_{uv}(k) > 0$  for some  $k, r_{vu}(\ell) > 0$  for some  $\ell, \mu_{uv}(k, \ell) > 0$ :  
Decrease  $\mu_{uv}(k, \ell)$  by a small  $\delta > 0$ . This decreases the objective by  $2M_{uv}\delta + g_{uv}(k, \ell)\delta$ .
4.  $r_{uv}(k) > 0$  for some  $k, r_{vu}(\ell) > 0$  for some  $\ell, \mu_{uv}(k, \ell) = 0$ :  
Pick any  $k'$  and  $\ell'$  such that  $\mu_{uv}(k, \ell') > 0$  and  $\mu_{uv}(k', \ell) > 0$ . Such  $k'$  and  $\ell'$  exist because  $r_{uv}(k) > 0$  and  $r_{vu}(\ell) > 0$ . Then proceed as follows:

- If  $\mu_{uv}(k', \ell') = 1$  then  $r_{uv}(k') > 0$  and  $r_{vu}(\ell') > 0$ . Proceed as in case 3.
- If  $\mu_{uv}(k', \ell') < 1$ , decrease  $\mu_{uv}(k, \ell')$  by a small  $\delta > 0$ , decrease  $\mu_{uv}(k', \ell)$  by  $\delta$ , and increase  $\mu_{uv}(k', \ell')$  by  $\delta$ . This changes the objective by  $-g_{uv}(k, \ell')\delta - g_{uv}(k', \ell)\delta + g_{uv}(k', \ell')\delta - 2M_{uv}\delta < 0$ .  $\square$

## 6.1 Complexity of the Reduction

Let us derive the complexity of the reduction. Clearly,  $|V'| = \mathcal{O}(|V| + |K|^2|E|)$  and  $|E'| = \mathcal{O}(|K|^2|E|)$ . The cumulative size of all numbers  $M_{uv}$  ( $\{u, v\} \in E$ ) is  $\mathcal{O}(\langle \mathbf{g} \rangle)$ . Each value  $M_{uv}$  appears as the Potts pairwise cost in  $\mathcal{O}(|K|^2)$  object pairs, thus all the Potts pairwise costs are described by a vector of size  $\mathcal{O}(|K|^2 \langle \mathbf{g} \rangle)$ . The cumulative size of the unary costs in  $\mathbf{g}'$  is bounded by the sum of sizes of all messages. Every  $M_{uv}$  induces  $\mathcal{O}(|K|^2)$  messages, each of them having the absolute value at most  $2M_{uv}$ . It means all the messages are described by a vector of size  $\mathcal{O}(|K|^2 \langle \mathbf{g} \rangle)$ , which proves the output has the size  $\mathcal{O}(|K|^2 \langle \mathbf{g} \rangle)$ . Note that the numbers (20) are (possibly much) smaller than (14) and (16). If  $|K|$  is fixed, the complexity of the reduction is linear.

## 7 Conclusion

Our results (Corollaries 6 and 9, Theorem 11) suggest that solving the LP relaxation of the pairwise min-sum problem with attractive Potts costs cannot be expected much easier than solving the LP relaxation of the general min-sum problem.

This statement may sound misleading in case of reduction with higher than linear complexity, because in that case efficiency of solving the LP relaxation of the Potts min-sum problem does not fully translate to efficiency of solving the general LP. However, our argument is more subtle: if a new principle were invented to solve the LP relaxation of Potts min-sum problems (e.g., similar to network flow algorithms), it would mean this principle is applicable to an arbitrary LP. Since there are only few principles to solve general LPs in polynomial time, this is unlikely.

In particular, message passing algorithms do not solve the LP relaxation of a general min-sum problem exactly, but find only a local (with respect to block-coordinate updates) dual optimum. It would be desirable to modify these algorithms to alleviate this drawback. One might hope this might be easier for Potts min-sum than for general min-sum. However, inventing a message passing algorithm that avoids local optima for Potts min-sum problems would mean it can solve general LPs.

Besides the results for the LP relaxation, we obtained similar reductions for the non-relaxed problems (Theorems 4, 7, 10). These may have practical impact in the case of exact (e.g., branch-and-bound) solvers, which can be tuned only for Potts problems. Unfortunately, they may not be useful for approximate solvers (such as primal move-making algorithms [2]) or solvers obtaining persistency [8,14], because the reductions may not preserve approximation ratio or persistency.

**Acknowledgment.** This work has been supported by the Czech Science Foundation project P202/12/2071 (both authors) and the EC project FP7-ICT-270138 DARWIN (the second author only).

## References

1. Boros, E., Hammer, P.L.: Pseudo-Boolean optimization. *Discrete Applied Mathematics* 123(1-3), 155–225 (2002)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence* 23(11), 1222–1239 (2001)
3. Chekuri, C., Khanna, S., Naor, J., Zosin, L.: A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics* 18(3), 608–625 (2005)
4. Chuzhoy, J., Naor, J.: The hardness of metric labeling. *SIAM J. Computation* 36(5), 1376–1386 (2007)
5. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Lellmann, J., Komodakis, N., Rother, C.: A comparative study of modern inference techniques for discrete energy minimization problem. In: *Conf. on Computer Vision and Pattern Recognition* (2013)
6. Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *J. ACM* 49(5), 616–639 (2002)
7. Koster, A., van Hoesel, S.P.M., Kolen, A.W.J.: The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters* 23(3-5), 89–97 (1998)
8. Kovtun, I.: Partial optimal labelling search for a NP-hard subclass of (max,+) problems. In: *Conf. German Assoc. for Pattern Recognition*, pp. 402–409 (2003)
9. Osokin, A., Vetrov, D., Kolmogorov, V.: Submodular decomposition framework for inference in associative Markov networks with global constraints. In: *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1889–1896 (2011)
10. Průša, D., Werner, T.: Universality of the local marginal polytope. In: *Conf. on Computer Vision and Pattern Recognition*, pp. 1738–1743. *IEEE Computer Society* (2013)
11. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: *Conf. on Computer Vision and Pattern Recognition* (2007)
12. Schlesinger, D., Flach, B.: Transforming an arbitrary MinSum problem into a binary one. *Tech. Rep. TUD-FI06-01*, Dresden University of Technology, Germany (April 2006)
13. Shlezinger, M.I.: Syntactic analysis of two-dimensional visual signals in noisy conditions. *Cybernetics and Systems Analysis* 12(4), 612–628 (1976)
14. Swoboda, P., Savchynskyy, B., Kappes, J.H., Schnörr, C.: Partial optimality via iterative pruning for the Potts model. In: *Conf. on Scale Space and Variational Methods in Computer Vision* (2013)
15. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 30(6), 1068–1080 (2008)
16. Živný, S.: *The Complexity of Valued Constraint Satisfaction Problems*. Cognitive Technologies. Springer (2012)
17. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1(1-2), 1–305 (2008)
18. Werner, T.: A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(7), 1165–1179 (2007)



# Coarse-to-Fine Minimization of Some Common Nonconvexities

Hossein Mobahi and John W. Fisher III

<sup>1</sup> Computer Science and Artificial Intelligence Lab. (CSAIL)

<sup>2</sup> Massachusetts Institute of Technology (MIT)

{hmobahi, fisher}@csail.mit.edu

**Abstract.** The continuation method is a popular heuristic in computer vision for nonconvex optimization. The idea is to start from a simplified problem and gradually deform it to the actual problem while tracking the solution. There are many choices for how to map the nonconvex objective to some convex task. One popular principle for such construction is Gaussian smoothing of the objective function. This involves an integration which may be expensive to compute numerically. We argue that often simple tricks at the problem formulation plus some mild approximations can make the resulted task amenable to closed form integral.

**Keywords:** Continuation Method, Diffusion Equation, Nonconvex Optimization, Graduated Nonconvexity.

## 1 Introduction

Nonconvex optimization tasks are ubiquitous in computer vision [25,32,21,13,23]. However, solving such problems (to global optimality) is generally intractable. Hence, often the nonconvex problem is either relaxed to a convex task [28,15], or heuristic optimization methods are utilized [7,5,29,8,24]. Each of these two methods has its own pros and cons. The global optimum of the convex relaxed task can be found efficiently. However, some aspects of the original problem may be lost because of the relaxation, which sometimes could be crucial. On the other hand, heuristic methods are not guaranteed to find the global optimum. In return, they directly target solving the nonconvex task. Hence, they sometimes can offer good local minima if not the global one.

A long standing deterministic heuristic for handling nonconvex tasks in computer vision is Blake and Zisserman's *Graduated Non-Convexity* (GNC) [6]. The idea, introduced about three decades ago, is to start from a convex problem. The latter is then progressively deformed to the actual objective while tracking the solution along the way. Around the same time, Terzopoulos used similar ideas for surface interpolation problems [30]. Outside of computer vision field, GNC technique is known under a broader class of optimization by *homotopy continuation* method [33].

The idea of optimization by continuation has been utilized in several interesting works. For example, Brox's thesis on image segmentation relies on this technique for optimization [7]. Black and Rangarajan used continuation for analyzing

spatial discontinuities and applied it to several problems in early vision [5]. In fact, coarse-to-fine image representation which is widely used in computer vision is related to the continuation method [24]. Note that state-of-the-art solutions for optical flow [29,8] and shape estimation [1] rely on multiscale representation to avoid poor local minima.

There is also a growing interest in using similar optimization methods within machine learning community. Some example applications include semi-supervised kernel machines [27], multiple instance learning [14,18], semi-supervised structured output [11], and language modeling [2]. It has also been suggested that recent training algorithms for deep architectures [17,12], which have made a breakthrough, in fact approximate continuation methods [3].

There is an infinite number of ways to progressively deform the nonconvex objective to some convex task. One possible principle is by Gaussian smoothing of the objective function [26,24]. In fact, we have recently shown that Gaussian smoothing has optimality for homotopy construction in a certain sense [22]. The Gaussian smoothing method convolves the nonconvex objective with an isotropic Gaussian kernel. This results in a collection of functions ranging from a highly smoothed to the actual nonconvex function, depending on the bandwidth parameter of the Gaussian. The continuation method processes this collection successive, starting the smoothed function and ending at the actual nonconvex function. Since going from high to low bandwidth reveals more details of the objective function, optimization by Gaussian homotopy continuation is also called *coarse-to-fine optimization*.

From practical viewpoint, the key challenge for using Gaussian homotopy is computing the convolution integral. In fact, using this approach makes sense only if this integral can be computed analytically<sup>1</sup>. This may seem disappointing at first for several reasons. First, the integrands that lead to a closed form integration are often rare and must have a very simple and nice form. In addition, some applications involve objective functions defined over discrete variables, for which Gaussian convolution is not well-defined.

Despite these challenges, we argue that sometimes simple tricks at the problem formulation and some mild approximations can make the resulted task amenable to closed form integral. To be concrete, we demonstrate this within two example tasks<sup>2</sup>. The first one focuses on handling discrete valued variables in a combinatorial setting. The example application is establishing correspondence between a pair of point clouds. The second example shows the use indicator functions as well as robust loss functions within an image denoising setup. For both applications, we show that the objective becomes convex after enough smoothing.

---

<sup>1</sup> The dimension of the integration domain is the number optimization variables. The numerical computation of this integral can be as expensive as exhaustive search of the domain for finding the global optimum.

<sup>2</sup> Both applications are formulated in their simplest form to allow focusing on the homotopy construction task. We do not aim at beating state of the art in such a simple setup, but rather produce comparable results against common alternatives.

In addition, we demonstrate that the minimizer of the convexified (i.e. highly smoothed) problem can be expressed in closed form.

Although the paper investigates two example applications, the underlying ideas may be generalized to some other tasks. Specifically, the energy function in both applications consist of *polynomials* and *Gaussian functions*. Both of these forms are amenable to closed form Gaussian convolution. Hence, Gaussian smoothing can be analytically computed for any energy function that can be constructed from these components. Note that these two components are very rich. For example, in the alignment example we will show that *discrete* variables can be replaced by continuous through simple polynomial penalties. Furthermore, in the denoising example, we will show that *indicator* functions and some *robust loss* functions can be expressed by a Gaussian form.

Throughout this paper, we use  $x$  for scalars,  $\mathbf{x}$  for vectors,  $\mathbf{X}$  for matrices, and  $\mathcal{X}$  for sets. Here  $\|\mathbf{x}\|$  means  $\|\mathbf{x}\|_2$  and  $\nabla$  means  $\nabla_{\mathbf{x}}$ , and  $\triangleq$  means equality by definition. The convolution operator is denoted by  $\star$ . The isotropic Gaussian kernel with standard deviation  $\sigma$  is shown by  $k_\sigma$ ,

$$k_\sigma(\mathbf{x}) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}.$$

## 2 Discrete Valued Variables

In this section we show how simple tricks at the problem formulation level can handle discrete valued variables. We use 3D point cloud alignment as the example task.

### 2.1 Formulation

Given two sets of points  $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^m$  and  $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^n$ , where each point is in  $\mathbb{R}^d$  and  $d = 3$  for 3D point clouds. Consider the affine transformation  $\mathbf{A}\mathbf{p} + \mathbf{b}$ , where  $\mathbf{A}$  is  $d \times d$  and  $\mathbf{b}$  is  $d \times 1$ . We define the optimal affine alignment as the following,

$$\begin{aligned} (\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*) &= \arg \min_{\mathbf{A}, \mathbf{b}, \mathbf{c}} \sum_{i=1}^m \sum_{j=1}^n (c_{i,j} \|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|)^2 & (1) \\ \text{s.t. } \forall j \sum_{i=1}^m c_{i,j} &= 1, \quad , \quad \forall i \forall j \quad c_{i,j} \in \{0, 1\}. \end{aligned}$$

The binary variables  $c_{i,j}$  determine the correspondence among the point pairs in  $\mathcal{P}$  and  $\mathcal{Q}$ . We arranged the elements  $c_{i,j}$  into vector of size  $m n$  denoted by  $\mathbf{c}$ . Without loss of generality, we assume that the point set  $\mathcal{P}$  is uncorrelated and has zero mean, with the largest variance being one as below,

$$\frac{1}{m} \sum_{i=1}^m \mathbf{p}_i = \mathbf{0} \quad , \quad \frac{1}{m} \sum_{i=1}^m \mathbf{p}_i \mathbf{p}_i^T = \text{diag}([1, \lambda_2, \lambda_3]), \quad (2)$$

where  $1 \geq \lambda_2 \geq \lambda_3 > 0$ . If  $\mathcal{P}$  does not have this property, we can easily process the points to get them this way<sup>3</sup>, as explained in the following. Suppose the original point sets are named  $\mathcal{P}^\circ$  and  $\mathcal{Q}^\circ$ . Let the spectral decomposition of  $\mathcal{P}^\circ$  be as below,

$$\frac{1}{m} \sum_{i=1}^m (\mathbf{p}_i^\circ - \bar{\mathbf{p}}^\circ)(\mathbf{p}_i^\circ - \bar{\mathbf{p}}^\circ)^T = \mathbf{V} \text{diag}(\mathbf{d}) \mathbf{V}^T, \quad (3)$$

where  $\bar{\mathbf{p}}^\circ = \frac{1}{m} \sum_{i=1}^m \mathbf{p}_i^\circ$ ,  $\mathbf{d}$  is the vector of eigenvalues and  $\mathbf{V}$  is a matrix with columns being eigenvectors of the covariance of  $\mathcal{P}^\circ$ . Then we define the sets  $\mathcal{P}$  and  $\mathcal{Q}$  by applying the shift  $\bar{\mathbf{p}}^\circ$ , rotation  $\mathbf{V}$ , and scaling  $1/\max(\mathbf{d})$  to the initial sets  $\mathcal{P}^\circ$  and  $\mathcal{Q}^\circ$  as shown below,

$$\forall \mathbf{p}_i^\circ \in \mathcal{P}^\circ, \mathbf{p}_i \triangleq \frac{1}{\max(\mathbf{d})} \mathbf{V}^T (\mathbf{p}_i^\circ - \bar{\mathbf{p}}^\circ) \mathbf{V} \quad (4)$$

$$\forall \mathbf{q}_j^\circ \in \mathcal{Q}^\circ, \mathbf{q}_j \triangleq \frac{1}{\max(\mathbf{d})} \mathbf{V}^T (\mathbf{q}_j^\circ - \bar{\mathbf{p}}^\circ) \mathbf{V}. \quad (5)$$

It is easy to check that the transformed set  $\mathcal{P}$  now has the assumed properties. Thus, we can apply the proposed affine alignment algorithm. Suppose the algorithm returns affine parameters  $\mathbf{A}$  and  $\mathbf{b}$  in the sense that it best transforms the set  $\mathcal{P}$  to the set  $\mathcal{Q}$  via  $\mathbf{A}\mathbf{p} + \mathbf{b}$ . We can easily use this solution to relate the original sets  $\mathcal{P}^\circ$  and  $\mathcal{Q}^\circ$  via  $\mathbf{A}^\circ \mathbf{p}^\circ + \mathbf{b}^\circ$ , where  $\mathbf{A}^\circ = \mathbf{V} \mathbf{A} \mathbf{V}^T$  and  $\mathbf{b}^\circ = (\mathbf{I} - \mathbf{A}^\circ) \bar{\mathbf{p}}^\circ + \max(\mathbf{d}) \mathbf{V} \mathbf{b} \mathbf{V}^T$ .

## 2.2 Smoothing

In order to apply Gaussian smoothing, the optimization must be in continuous variables and unconstrained. To achieve the first property, we express the discrete constraint  $c_{i,j} \in \{0, 1\}$  equivalently by the continuous equality constraint of form  $c_{i,j}(1 - c_{i,j}) = 0$ . Thus, the optimization task becomes as the following,

$$\begin{aligned} (\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*) &= \arg \min_{\mathbf{A}, \mathbf{b}, \mathbf{c}} \sum_{i,j} (c_{i,j} \|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|)^2 \\ \text{s.t. } &-1 + \sum_{i=1}^m c_{i,j} = 0 \quad , \quad c_{i,j}(1 - c_{i,j}) = 0. \end{aligned} \quad (6)$$

To satisfy the second property, we approximate the problem by replacing equality constraints  $h(\mathbf{c}) = 0$  by the objective penalty  $h^2(\mathbf{c})$ . Thus, the approximate objective function becomes as the following,

<sup>3</sup> If the point set is degenerate, i.e. its covariance matrix has some eigenvalues equal to zero, then we cannot have  $\mathcal{P}$  in the desired form. In that case, the null space of the data can be removed to obtain a lower dimensional representation for the points. Everything else in the paper remains the same for the new set, as there is nothing special in our analysis to force  $d = 3$ .

$$\begin{aligned}
f(\mathbf{A}, \mathbf{b}, \mathbf{c}) &\triangleq \frac{\epsilon}{m n} \left( \sum_{i=1}^m \sum_{j=1}^n c_{i,j}^2 \|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|^2 \right) \\
&\quad + \frac{1}{m n} \left( \sum_{j=1}^n \left(1 - \sum_{i=1}^m c_{i,j}\right)^2 + \sum_{i=1}^m \sum_{j=1}^n c_{i,j}^2 (1 - c_{i,j})^2 \right), \quad (7)
\end{aligned}$$

where  $\epsilon > 0$  is a small number. We can now convolve the objective function with the Gaussian kernel  $k_\sigma(\text{vec}(\mathbf{A}, \mathbf{b}, \mathbf{c}))$ , where  $\text{vec}$  concatenates all variables into a long vector as below,

Due to diagonal form of the covariance, we first compute convolution w.r.t. variables  $\{c_{i,j}\}$ , and then convolve the result with variables  $\text{vec}(\mathbf{A}, \mathbf{b})$ . Convolution in variables  $\{c_{i,j}\}$  is easily computed as follows,

$$\begin{aligned}
g_1(\mathbf{A}, \mathbf{b}, \mathbf{c}; \sigma) &\triangleq [f \star k(\cdot; \sigma^2)](\mathbf{c}) \\
&= \frac{\epsilon}{m n} \sum_{i=1}^m \sum_{j=1}^n (c_{i,j}^2 + \sigma^2) (\|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|^2) \\
&\quad + \frac{1}{m n} \sum_{j=1}^n \left(1 - \sum_{i=1}^m c_{i,j}\right)^2 + \frac{1}{m n} \sum_{i=1}^m \sum_{j=1}^n (c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2 \left(c_{i,j} - \frac{1}{2}\right)^2.
\end{aligned}$$

We now apply the convolution w.r.t.  $\boldsymbol{\theta} \triangleq \text{vec}(\mathbf{A}, \mathbf{b})$ , and denote the affine transform by as  $\boldsymbol{\tau}(\mathbf{p}; \boldsymbol{\theta}) \triangleq \mathbf{A}\mathbf{p} + \mathbf{b}$ .

$$\begin{aligned}
g(\boldsymbol{\theta}, \mathbf{c}; \sigma) &\triangleq \left[ g_1(\cdot, \mathbf{c}; \sigma) \star k(\cdot; \sigma^2) \right](\boldsymbol{\theta}) \\
&= \frac{\epsilon}{m n} \left( \sum_{i,j}^{m,n} (c_{i,j} + \sigma^2) \int_{\mathbb{R}^3} \|\mathbf{r} - \mathbf{q}_j\|^2 k_{\sigma\sqrt{1+\|\mathbf{p}_i\|^2}}(\boldsymbol{\tau}(\mathbf{p}_i, \boldsymbol{\theta}) - \mathbf{r}) d\mathbf{r} \right) \quad (8) \\
&\quad + \frac{1}{m n} \sum_{j=1}^n \left(1 - \sum_{i=1}^m c_{i,j}\right)^2 + \frac{1}{m n} \sum_{i,j}^{m,n} (c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2 \left(c_{i,j} - \frac{1}{2}\right)^2 \\
&= \frac{\epsilon}{m n} \sum_{i,j}^{m,n} (c_{i,j}^2 + \sigma^2) (\|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|^2 + 3\sigma^2(1 + \|\mathbf{p}_i\|^2)) \\
&\quad + \frac{1}{m n} \left( \sum_{j=1}^n \left(1 - \sum_{i=1}^m c_{i,j}\right)^2 + \sum_{i,j}^{m,n} (c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2 \left(c_{i,j} - \frac{1}{2}\right)^2 \right). \quad (9)
\end{aligned}$$

where (8) uses the *transformation kernel* for the affine map [24]. This kernel allows writing the high dimensional convolution w.r.t.  $\boldsymbol{\theta}$  equivalently by a  $d$ -dimensional integral transform, where here  $d = 3$ .

### 2.3 Asymptotic Minimizer

It is easy to check that, as  $\sigma \rightarrow \infty$ , the Hessian of the objective converges to a matrix with zero off-diagonals and positive diagonals (hence asymptotically

convex). In addition, zero crossing the gradient when  $\sigma \rightarrow \infty$  leads to the following asymptotic minimizer,

$$\begin{aligned} \mathbf{A}^* &= \left( \frac{1}{m n} \sum_{i=1}^m \sum_{j=1}^n \mathbf{q}_j \mathbf{p}_i^T \right) \text{diag} \left( \left[ 1, \frac{1}{\lambda_2}, \frac{1}{\lambda_3} \right] \right) \\ \mathbf{b}^* &= \frac{1}{n} \sum_{j=1}^n \mathbf{q}_j \quad , \quad c_{i,j}^* = \frac{1}{2 + \epsilon (1 + \|\mathbf{p}_i\|^2)} . \end{aligned} \quad (10)$$

## 2.4 Continuation Updates

The continuation process moves from the stationary point attained at the previous smoothing level  $\sigma$  to the (possibly local) minimum of the current objective function formed by a smaller  $\sigma$ , i.e. reduced smoothing. For each fixed  $\sigma$ , the stationary point of the problem is obtained by looping over gradient descent with the line search until convergence. The sequence for  $\sigma$  is generated by starting from  $\sigma_0 = 2$  and updating it by  $\sigma_{k+1} = 0.9\sigma_k$ , until the value of  $\sigma$  falls below 0.01. A great advantage of the low-order polynomial formulation of the alignment task is that, we can compute the *optimal line search* in closed form as explained below. The gradient of  $g(\mathbf{A}, \mathbf{b}, \mathbf{c}; \sigma)$  in (9) can be expressed as follows,

$$\frac{\partial g}{\partial \mathbf{A}} = 2\epsilon \sum_{i=1}^m \sum_{j=1}^n c_{i,j}^2 \left( (\mathbf{A} \mathbf{p}_i + \mathbf{b} - \mathbf{q}_j) \mathbf{p}_i^T \right), \quad (11)$$

$$\frac{\partial g}{\partial \mathbf{b}} = 2\epsilon \sum_{i=1}^m \sum_{j=1}^n c_{i,j}^2 \left( \mathbf{A} \mathbf{p}_i + \mathbf{b} - \mathbf{q}_j \right). \quad (12)$$

$$\begin{aligned} \frac{\partial g}{\partial c_{i,j}} &= \epsilon (\|\mathbf{A} \mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|^2 + 3\sigma^2 (1 + \|\mathbf{p}_i\|^2)) - 2 + 2 \sum_{k=1}^m c_{k,j} \\ &+ 2 \left( c_{i,j} (c_{i,j} - 1) (2c_{i,j} - 1) + 6\sigma^2 (c_{i,j} - \frac{1}{2}) \right). \end{aligned} \quad (13)$$

Define the updated solution as  $\mathbf{A}^+ \triangleq \mathbf{A} + \alpha \frac{\partial g}{\partial \mathbf{A}}$ ,  $\mathbf{b}^+ \triangleq \mathbf{b} + \alpha \frac{\partial g}{\partial \mathbf{b}}$ , and  $\mathbf{c}^+ \triangleq \mathbf{c} + \alpha \frac{\partial g}{\partial \mathbf{c}}$ . The optimal step size  $\alpha$  can be obtained by zero crossing  $\frac{d}{d\alpha} g(\mathbf{A}^+, \mathbf{b}^+, \mathbf{c}^+; \sigma)$ . By collecting different exponents of  $\alpha$ , the latter can be written as below,

$$\frac{d}{d\alpha} g(\mathbf{A}^+, \mathbf{b}^+, \mathbf{c}^+; \sigma) = t_3 \alpha^3 + t_2 \alpha^2 + t_1 \alpha + t_0, \quad (14)$$

where  $t_0$ ,  $t_1$ ,  $t_2$ , and  $t_3$  are constants introduced for brevity<sup>4</sup>.

It is obvious that (14) is a *cubic equation* with the following closed-form roots for choices of  $w \in \{-2, 1 + i\sqrt{3}, 1 - i\sqrt{3}\}$ .

$$\alpha = -\frac{1}{6t_3} \left( 2t_2 + w \left( \frac{T_1 + \sqrt{T_1^2 - T_2}}{2} \right)^{\frac{1}{3}} + w^* \left( \frac{T_1 - \sqrt{T_1^2 - T_2}}{2} \right)^{\frac{1}{3}} \right),$$

where  $w^*$  is the complex conjugate of  $w$ , and the auxiliary variables  $T_1$  and  $T_2$  are defined as follows,

$$T_1 \triangleq 2t_2^3 - 9t_1t_2t_3 + 27t_0t_3^2 \quad , \quad T_2 \triangleq 4(t_2^2 - 3t_1t_3)^3 .$$

Obviously, we only consider the *real roots* of the above equation. We can evaluate  $g(\mathbf{A}^+, \mathbf{b}^+, \mathbf{c}^+; \sigma)$  at all the real roots (at most three) and choose the one that attains the smallest value of  $g(\mathbf{A}^+, \mathbf{b}^+, \mathbf{c}^+; \sigma)$ .

Algorithm 1 shows the procedure for affine alignment by Gaussian smoothing and path following.

## 2.5 Results

In this section, we present that result obtained by Algorithm 1. We use Iterative Closest Point (ICP) algorithm [4] as a baseline result. The idea of ICP is to alternate between creating a correspondence between pair of points (of the two clouds) and refining the geometric transformation between the corresponding

<sup>4</sup> The constants  $t_0$ ,  $t_1$ ,  $t_2$ , and  $t_3$  have the following form,

$$\begin{aligned} t_0 &\triangleq \sum_{i,j} 2 \frac{\partial g}{\partial c_{i,j}} (2c_{i,j} - 1) ((c_{i,j} - 1)c_{i,j} + 3\sigma^2) \\ &\quad + 3\epsilon \sigma^2 \frac{\partial g}{\partial c_{i,j}} (1 + \|\mathbf{p}_i\|^2) + \epsilon \frac{\partial g}{\partial c_{i,j}} \|\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j\|^2 \\ &\quad + 2c_{i,j} \epsilon (\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j)^T \left( \frac{\partial g}{\partial \mathbf{A}} \mathbf{p}_i + \frac{\partial g}{\partial \mathbf{b}} \right) + 2 \sum_{j=1}^n \left( \left( \sum_{i=1}^m \frac{\partial g}{\partial c_{i,j}} \right) (-1 + \sum_{i=1}^m c_{i,j}) \right) \\ t_1 &\triangleq \sum_{i,j} 2 \left( \frac{\partial g}{\partial c_{i,j}} \right)^2 (1 + 6c_{i,j}(c_{i,j} - 1) + 6\sigma^2) \\ &\quad + 2\epsilon c_{i,j} \left\| \frac{\partial g}{\partial \mathbf{A}} \mathbf{p}_i + \frac{\partial g}{\partial \mathbf{b}} \right\|^2 + 2 \sum_{j=1}^n \left( \sum_{i=1}^m \frac{\partial g}{\partial c_{i,j}} \right)^2 \\ &\quad + 4\epsilon \frac{\partial g}{\partial c_{i,j}} (\mathbf{A}\mathbf{p}_i + \mathbf{b} - \mathbf{q}_j)^T \left( \frac{\partial g}{\partial \mathbf{A}} \mathbf{p}_i + \frac{\partial g}{\partial \mathbf{b}} \right) \\ t_2 &\triangleq \sum_{i,j} 3 \frac{\partial g}{\partial c_{i,j}} \left( 2 \left( \frac{\partial g}{\partial c_{i,j}} \right)^2 (2c_{i,j} - 1) + \epsilon \left\| \frac{\partial g}{\partial \mathbf{A}} \mathbf{p}_i + \frac{\partial g}{\partial \mathbf{b}} \right\|^2 \right) \\ t_3 &\triangleq \sum_{i,j} 4 \left( \frac{\partial g}{\partial c_{i,j}} \right)^4 . \end{aligned}$$

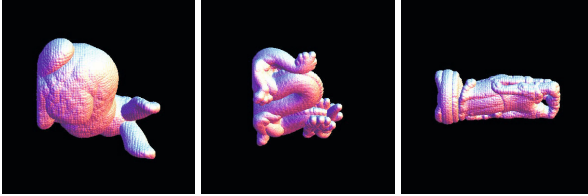
**Algorithm 1.** Point Cloud Alignment by Gaussian Homotopy Continuation

- 
1. Input: Point clouds  $\mathcal{P}$  (as eq. (2)) and  $\mathcal{Q}$ , a small  $\epsilon > 0$ , a sequence  $\sigma_1 > \sigma_2 > \dots > \sigma_N > 0$ .
  2.  $\mathbf{A} = (\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n \mathbf{q}_j \mathbf{p}_i^T) \text{diag}([1, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}])$
  3.  $\mathbf{b} = \frac{1}{n} \sum_{j=1}^n \mathbf{q}_j$
  4.  $c_{i,j} = \frac{1}{2+\epsilon(1+\|\mathbf{p}_i\|^2)}$
  5. **for**  $k = 1 \rightarrow N$  **do**
  6.   **repeat**
  7.      $\mathbf{A} = \mathbf{A} + \alpha \frac{\partial g}{\partial \mathbf{A}}$ .
  8.      $\mathbf{b} = \mathbf{b} + \alpha \frac{\partial g}{\partial \mathbf{b}}$ .
  9.      $\mathbf{c} = \mathbf{c} + \alpha \frac{\partial g}{\partial \mathbf{c}}$
  10.   **until** Convergence
  11. **end for**
  12. Output:  $(\mathbf{A}, \mathbf{b}, \mathbf{c})$
- 

points. Both algorithms share the same initialization, which is the asymptotic minimizer of the alignment objective presented in (10). For the continuation algorithm, we set  $\epsilon = 0.01$ .

We use some of the 3D objects provided by Stanford’s dataset [9,19,31], each of which comprises a set  $\mathcal{P}$ . We then create  $\mathcal{Q}$  by rotating points in  $\mathcal{P}$  by  $n$  degrees along all three  $x$ ,  $y$  and  $z$  axes, where  $n$  varies between 30 degrees to 90 degrees, in steps of 15 degrees. This way, for each  $\mathcal{P}$ , we derive a set of problems  $\{\mathcal{Q}_n\}$  that are increasingly more challenging as  $n$  grows.

Figures 1 and 2 indicate that ICP gets stuck in poor local minima more often than the continuation method, before reaching a reasonable alignment.



**Fig. 1.** The point set  $\mathcal{Q}$  for each 3D object

### 3 Indicator Function and Robust Loss

In this section we show how indicator function and a robust loss function (truncated quadratic) can be approximated in a way that become amenable to closed form integration. This is shown through an example task for image denoising.





**Fig. 2.** Each object occupies three successive rows, where each row has the following role. (Top) Input  $\mathcal{P}$ , which is a rotated version of  $\mathcal{Q}$ . (Middle) Transformed  $\mathcal{P}$  to match  $\mathcal{Q}$  using ICP. (Bottom) Transformed  $\mathcal{P}$  to match  $\mathcal{Q}$  using proposed method.

### 3.1 Formulation

Given an image matrix  $\mathbf{V}$  whose entries are affected additively by independent Gaussian noise. The resulted noisy image is denoted by  $\tilde{\mathbf{V}}$ . The noise has zero mean and unknown variance. Suppose the original image has a piecewise constant structure<sup>5</sup>. The denoising problem can be formulated as the following,

$$\mathbf{U}^* = \arg \min_{\mathbf{U}} \sum_{i,j} \lambda (u_{i,j} - \tilde{v}_{i,j})^2 + I_{\|\nabla u_{i,j}\| \neq 0}, \quad (15)$$

where  $I$  is an indicator function that is one if its argument is true and zero otherwise. This regularization resembles the  $\ell_0$  norm of gradient's magnitude map. Here  $\nabla u_{i,j}$  is the finite difference approximation of the gradient at the entry  $u_{i,j}$ , i.e.  $\|\nabla u_{i,j}\|^2 \triangleq (u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2$ . The parameter  $\lambda$  balances between fidelity and regularization.

Observe that an indicator function  $I_{x \neq 0}$  can be also expressed as  $1 - \lim_{\epsilon \rightarrow 0} e^{-\frac{x^2}{2\epsilon^2}}$ . For practical applications,  $1 - e^{-\frac{x^2}{2\epsilon^2}}$  with a small enough  $\epsilon$  provides a reasonable approximation to the limit case (Figure 3-Left). The advantage of this particular approximation for the indicator function is this it allows for a closed-form Gaussian convolution. Using that, the objective can be written as below,

$$\begin{aligned} f(\mathbf{U}) &= \sum_{i,j} \lambda (u_{i,j} - \tilde{v}_{i,j})^2 + 1 - e^{-\frac{\|\nabla u_{i,j}\|^2}{2\epsilon^2}} \\ &= \sum_{i,j} \lambda (u_{i,j} - \tilde{v}_{i,j})^2 + 1 - e^{-\frac{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2}{2\epsilon^2}}. \end{aligned}$$

Note that the Gaussian function, which a larger choice of  $\epsilon$ , can also provide a good approximation for the truncated quadratic form (Figure 3-Right). This approximation maintains the key property of robust loss functions, which is having flat tails. In the following, however, we continue with the simple (non-robust) quadratic loss.

### 3.2 Smoothing

Convolving this objective with Gaussian kernels in variables  $u_{i,j}$  and dropping constant terms leads to the following,

$$\begin{aligned} g(\mathbf{U}, \sigma) &= \sum_{i,j} \lambda (u_{i,j} - \tilde{v}_{i,j})^2 \\ &\quad - c e^{-\frac{\epsilon^2 \|\nabla u_{i,j}\|^2 + 2\sigma^2 (u_{i,j}^2 + u_{i,j+1}^2 + u_{i+1,j}^2 - u_{i,j+1} u_{i+1,j} - u_{i,j} (u_{i,j+1} + u_{i+1,j}))}{2(\sigma^2 + \epsilon^2)(3\sigma^2 + \epsilon^2)}}, \end{aligned}$$

where  $c$  is a constant factor.

<sup>5</sup> That is the case for most shape images.



**Fig. 3.** Using the function  $e^{-\frac{x^2}{2\epsilon^2}}$  to approximate indicator function (Left) by  $\epsilon = 0.005$  and (robust) truncated quadratic loss (Right) by  $\epsilon = 1/2$ . Both cases are plotted in the range  $x \in [-4, 4]$ .

### 3.3 Asymptotic Minimizer

As  $\sigma \rightarrow \infty$ , the regularization term vanishes and only the convex quadratic term remains. Hence, this problem is asymptotically convex, and its asymptotic minimizer is simply the solution of the convex quadratic part, which is  $u_{i,j} = \tilde{v}_{i,j}$ .

### 3.4 Continuation

The sequence for  $\sigma$  is generated by starting from  $\sigma_0 = 2$  and updating it by  $\sigma_{k+1} = 0.9\sigma_k$ , until the value of  $\sigma$  falls below 0.01. Sensitivity parameter  $\epsilon$  is set to  $\frac{2}{255}$ , which means 2 intensity levels out of 255 possible levels in an 8-bit representation. For each value of  $\sigma$ , gradient descent loop is performed until convergence. The loop starts by initializing the solution obtained from the previous value of  $\sigma$ . The exponential form appearing in this application prevents finding the optimal line search in closed form. Thus, here we use the plain gradient descent.

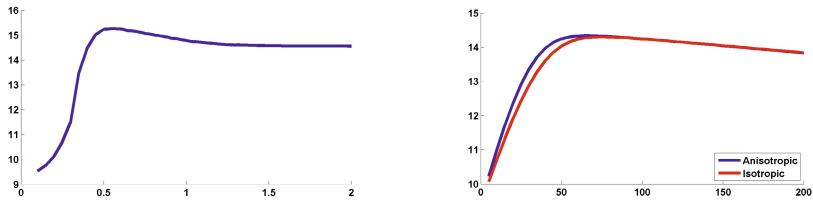
### 3.5 Results

The method is applied to an example shape image degraded by Gaussian noise. The intensity values of the image range between zero and one, and the standard deviation of the noise is 0.5, which is quite severe. We apply four other methods to these data for comparison. Specifically, we use isotropic and anisotropic total variation [16] using publicly available code<sup>6</sup>, BM3D denoising package<sup>7</sup> [10], and KMeans clustering [20] shipped with Matlab. Total variation essentially penalizes the  $\ell_1$  norm of the gradient's magnitude. Note that  $\ell_1$  norm is the convex envelope for the  $\ell_0$  norm, hence the best possible convex approximation of the actual problem.

For BM3D, we provide the algorithm with the true value of noise variance, which is to the advantage of this method. The total variation method, like ours,

<sup>6</sup> We used Matlab code published by Benjamin Tremoulheac.

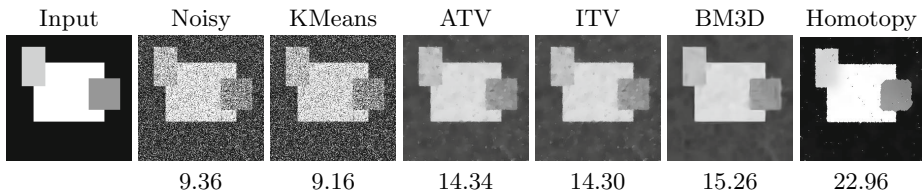
<sup>7</sup> Authors of this package have made their code publicly available. We used version 2.0 of this package.



**Fig. 4.** Plots of PSNR value versus algorithm’s parameter. Left: BM3D (horizontal axis shows multipliers of noise standard deviation, e.g. 2 means twice the value of the standard deviation). Right: Total Variation (the parameter balances between regularization and fidelity).

depends on a  $\lambda$  parameter that balances fidelity versus regularization. These parameters were carefully searched for each method to obtain maximally possible PSNR value (Figure 4). To ensure KMeans’s solution has not been unlucky with initialization, it is run 100 times, and only the one with the lowest cost function is reported here.

The output of each method and their associated PSNR values are shown in Figure 5.



**Fig. 5.** Denoising a shape image using different methods. The best PSNR attained by each method is show below its image.

## 4 Conclusion

In this work we argue that the convolution integral associated with the Gaussian homotopy continuation can be computed in closed form for some interesting scenarios. Such closed form expression is of great importance and it makes the Gaussian homotopy method useful in practice. We explored this idea within two simple scenarios that involve difficult combinatorial nonconvexities.

**Acknowledgments.** This work is partially funded by the Shell Research. First author is grateful to William T. Freeman for supporting this work.

## References

1. Barron, J.T., Malik, J.: Color constancy, intrinsic images, and shape estimation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 57–70. Springer, Heidelberg (2012)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: International Conference on Machine Learning, ICML (2009)
3. Bengio, Y.: Learning Deep Architectures for AI. Now Publishers Inc. (2009)
4. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14(2) (1992)
5. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision* 19(1), 57–91 (1996)
6. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press (1987)
7. Brox, T.: From pixels to regions: partial differential equations in image analysis. Ph.D. thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany (April 2005)
8. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(3), 500–513 (2011)
9. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, pp. 303–312. ACM, New York (1996)
10. Dabov, F., Katkovnik, E.: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing* 16(8), 2080–2095 (2007)
11. Dhillon, P.S., Keerthi, S.S., Bellare, K., Chapelle, O., Sundararajan, S.: Deterministic annealing for semi-supervised structured output learning. In: AISTATS 2012, vol. 15 (2012)
12. Erhan, D., Manzagol, P.A., Bengio, Y., Bengio, S., Vincent, P.: The difficulty of training deep architectures and the effect of unsupervised pre-training. In: AISTATS, pp. 153–160 (2009)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(9), 1627–1645 (2010)
14. Gehler, P., Chapelle, O.: Deterministic annealing for multiple-instance learning. In: AISTATS 2007, Microtome, Brookline, MA, USA, pp. 123–130 (March 2007)
15. Goldluecke, B., Strelakovsky, E., Cremers, D.: Tight convex relaxations for vector-valued labeling. *SIAM J. Imaging Sciences* 6(3), 1626–1664 (2013)
16. Goldstein, T., Osher, S.: The split bregman method for  $l_1$ -regularized problems. *SIAM J. Imaging Sciences* 2(2), 323–343 (2009)
17. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief networks. *Neural Computation* 18(7), 1527–1554 (2006)
18. Kim, M., Torre, F.D.: Gaussian processes multiple instance learning pp. 535–542 (2010)
19. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: SIGGRAPH, pp. 313–324 (1996)

20. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
21. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM International Conference Proceeding Series, vol. 382, p. 87. ACM (2009)
22. Mobahi, H., Fisher III, J.W.: On the link between gaussian homotopy continuation and convex envelopes. In: Tai, X.-C., Bae, E., Chan, T.F., Lysaker, M. (eds.) EMMCVPR 2015. LNCS, vol. 8932, pp. 43–56. Springer, Heidelberg (2015)
23. Mobahi, H., Rao, S., Ma, Y.: Data-driven image completion by image patch subspaces. In: Picture Coding Symposium (2009)
24. Mobahi, H., Ma, Y., Zitnick, L.: Seeing through the Blur. In: Proceedings of CVPR 2012 (2012)
25. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42(5), 577–685 (1989)
26. Nielsen, M.: Graduated non-convexity by smoothness focusing. In: Proceedings of the British Machine Vision Conference, p. 60. BMVA Press (1993)
27. Sindhvani, V., Keerthi, S.S., Chapelle, O.: Deterministic annealing for semi-supervised kernel machines. In: ICML 2006, pp. 841–848. ACM, New York (2006)
28. Strelakovsky, E., Chambolle, A., Cremers, D.: Convex relaxation of vectorial problems with coupled regularization. *SIAM J. Imaging Sciences* 7(1), 294–336 (2014)
29. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2439. IEEE (June 2010)
30. Terzopoulos, D.: The computation of visible-surface representations. *IEEE Trans. Pattern Anal. Mach. Intell.* 10(4), 417–438 (1988)
31. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, pp. 311–318. ACM, New York (1994)
32. Vural, E., Frossard, P.: Analysis of descent-based image registration. *SIAM J. Imaging Sciences* 6(4), 2310–2349 (2013)
33. Watson, L.T.: Theory of globally convergent probability-one homotopies for non-linear programming. *SIAM Journal on Optimization*, 761–780 (2001)

# Why Does Non-binary Mask Optimisation Work for Diffusion-Based Image Compression?

Laurent Hoeltgen and Joachim Weickert

Mathematical Image Analysis Group,  
Faculty of Mathematics and Computer Science, Campus E1.7  
Saarland University, 66041 Saarbrücken, Germany  
{hoeltgen,weickert}@mia.uni-saarland.de

**Abstract.** Finding optimal data for inpainting is a key problem for image-compression with partial differential equations. Not only the location of important pixels but also their values should be optimal to maximise the quality gain. The position of important data is usually encoded in a binary mask. Recent studies have shown that allowing non-binary masks may lead to tremendous speedups but comes at the expense of higher storage costs and yields prohibitive memory requirements for the design of competitive image compression codecs. We show that a recently suggested heuristic to eliminate the additional storage costs of the non-binary mask has a strong theoretical foundation in finite dimension. Binary and non-binary masks are equivalent in the sense that they can both give the same reconstruction error if the binary mask is supplemented with optimal data which does not increase the memory footprint. Further, we suggest two fast numerical schemes to obtain this optimised data. This provides a significant building block in the conception of efficient data compression schemes with partial differential equations.

**Keywords:** Laplace Interpolation, Inpainting, Convex Optimisation.

## 1 Introduction

A major challenge in data analysis is the reconstruction of a function, for example a 1D signal or an image, from a few data points. In image processing this interpolation problem is called inpainting [1, 2]. Often one has no influence on the given data and thus improvements can only be made by introducing more powerful reconstruction models. In some interesting applications however, one has the freedom to choose the data used for the reconstruction. For instance, in recent approaches related to image compression [3–12] the authors selected suitable interpolation data for reconstructions via partial differential equations (PDEs). Köstler et al. demonstrated in [13] that PDEs can also be used to compress video sequences. Let us emphasise that finding good data sets for interpolation is by no means a simple task. Choosing for example 5% of the pixels from a  $256 \times 256$  pixel large image offers more than  $10^{5000}$  possible combinations.

Besides a good selection for the position of the interpolation data, one can also consider an optimisation of corresponding data values in the co-domain.

Schmaltz et al. [6] used direct searching strategies to find good tonal values for the reconstruction of their nonlinear diffusion process. Mainberger et al. [8] presented a solid mathematical foundation of tonal optimisation and emphasised the benefits of a good spatial and tonal data selection. Since their inpainting was based on the Laplace equation, the optimal grey values could be found by solving a least squares approach. A related optimal control based model to find good inpainting masks was considered by Hoeltgen et al. [10]. This model, however, uses a regularised formulation that does not require the mask to be binary. It reduces an unfeasible combinatorial problem to a series of convex optimisation problems that can be solved in a highly efficient way. Similar models were also discussed in [11] by Chen et al., whereas Ochs et al. suggested fast numerics in [14, 15]. The approaches of Mainberger et al. [8], Hoeltgen et al. [10], and Chen et al. [11] achieve a similar high level of reconstruction quality. The benefits of the control based approach of Hoeltgen et al. [10] over the Mainberger method [8] is its significantly lower runtime. Unfortunately, storing non-binary masks is expensive in terms of memory requirements, especially in the context of image compression. As a remedy, Hoeltgen et al. [10] suggested a heuristic to reduce the storage requirements. They proposed to binarise the mask and to apply the tonal optimisation of Mainberger et al. [8] as a postprocessing step. Interestingly this heuristic yielded a intriguing phenomenon: *The error with optimal mask values and original data were almost identical to the errors with binary masks and optimised grey values.*

**Our Contribution.** The goal of our paper is to show that the similarity in the error measures discovered in [10] is no coincidence. We show that in a finite dimensional setting the reconstruction error with an optimal non-binary mask and original image data is always identical to the error with a binary mask combined with tonal optimisation. Thus, we provide a mathematically sound foundation for the development of a image compression codec based on the Laplace equation. Furthermore, we also propose two highly efficient algorithms to handle the latter tonal value optimisation on the CPU and the GPU.

**Structure of the Paper.** Our paper is organised as follows. In Section 2 we briefly introduce the underlying inpainting scheme as well as the related optimisation tasks that will be analysed in this paper. Section 3 shows the main result of this work, namely the equivalence between the optimisation problems from the first section. Next, Section 4 demonstrates two new numerical schemes that allow a fast and efficient optimisation on both the CPU and GPU. Finally, the paper is closed in Section 5 with a summary and an outlook on future challenges.

## 2 Inpainting with Homogeneous Diffusion

Inpainting with homogeneous diffusion (sometimes also called Laplace interpolation) is a rather simple reconstruction method that is well suited for highly scattered data in arbitrary dimensional settings. It can be modelled as follows. Let  $f : \Omega \rightarrow \mathbb{R}$  be a smooth function on some bounded domain  $\Omega \subset \mathbb{R}^n$



with a sufficiently regular boundary  $\partial\Omega$ . Throughout this work, we will restrict ourselves to the case  $n = 2$  (grey value images) even though many of the results hold for arbitrary  $n \geq 1$ . Moreover, let us assume that there exists a closed nonempty set of known data  $\Omega_K \subsetneq \Omega$  that will be interpolated by the underlying diffusion process. Homogeneous diffusion inpainting considers the following partial differential equation with mixed boundary conditions.

$$\begin{aligned} -\Delta u &= 0, & \text{on } \Omega \setminus \Omega_K \\ u &= f, & \text{on } \partial\Omega_K \\ \partial_n u &= 0, & \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{1}$$

where  $\partial_n u$  denotes the derivative of  $u$  in the outer normal direction. We assume that both boundary sets  $\partial\Omega_K$  and  $\partial\Omega \setminus \partial\Omega_K$  are nonempty. Equations of this type are commonly referred to as mixed boundary value problems and sometimes also as Zaremba's problem named after Stanislaw Zaremba who studied such equations already in 1910 [16]. The existence and uniqueness of solutions has been extensively studied during the last century. Showing that (1) is indeed solvable is by no means a trivial feat. We refer to [17] for an extensive study of linear elliptic partial differential equations. A particularly easy case is the 1-D setting, where the solution can obviously be expressed using piecewise linear splines interpolating data on  $\partial\Omega_K$ .

Following [8], we introduce the *confidence function*  $c: \Omega \rightarrow \mathbb{R}$  which states whether a point is known or not. It is defined by

$$c(\mathbf{x}) := \begin{cases} 1 & \text{for } \mathbf{x} \in \Omega_K, \\ 0 & \text{for } \mathbf{x} \in \Omega \setminus \Omega_K. \end{cases} \tag{2}$$

The confidence function lets us rewrite (1) as a more compact functional equation of the form

$$\begin{aligned} c(\mathbf{x})(u(\mathbf{x}) - f(\mathbf{x})) - (1 - c(\mathbf{x}))\Delta u(\mathbf{x}) &= 0, & \text{on } \Omega \\ \partial_n u(\mathbf{x}) &= 0, & \text{on } \partial\Omega \setminus \partial\Omega_K. \end{aligned} \tag{3}$$

As shown in [5, 8], the choice of  $c$  has a substantial influence on the solution. For most parts of this text we will prefer the formulation (3), as it is more comfortable to work with. Further, this formulation also makes sense when  $c$  is not binary-valued but takes arbitrary values. This observation was also exploited in [10] where the authors complemented (3) by a convex energy to obtain a sparse set of optimal values for  $c$ .

A discrete framework corresponding to (3) is easily obtained by a straightforward discretisation of the functions  $c$ ,  $u$  and  $f$  on a regular grid of size  $n_1 \times n_2$  and by placing the corresponding entries in vectors  $\mathbf{c}$ ,  $\mathbf{u}$  and  $\mathbf{f}$  respectively. If  $\mathbf{A}$  represents the symmetric  $N \times N$  matrix ( $N$  being the total numbers of pixels on

our grid, e.g.  $N = n_1 n_2$ ) of the discrete Laplace operator  $\Delta$  with homogeneous Neumann boundary conditions on  $\partial\Omega \setminus \partial\Omega_K$  we obtain

$$\text{diag}(\mathbf{c})(\mathbf{u} - \mathbf{f}) + (\mathbf{I} - \text{diag}(\mathbf{c}))(-\mathbf{A})\mathbf{u} = \mathbf{0} \quad (4)$$

where  $\mathbf{I}$  is the identity matrix,  $\text{diag}(\mathbf{c})$  is a diagonal matrix with the sampled values from  $\mathbf{c}$  as its entries on the main diagonal. By a simple reordering of the terms, (4) can be rewritten as the following linear system,

$$\left( \text{diag}(\mathbf{c}) + \left( \mathbf{I} - \text{diag}(\mathbf{c}) \right) (-\mathbf{A}) \right) \mathbf{u} = \text{diag}(\mathbf{c}) \mathbf{f} \quad (5)$$

If the vector  $\mathbf{c}$  contains as its entries only the values 0 or 1 and if it is not the zero vector, then it has been shown in [7] that this linear system of equations has a unique solution and that it can be solved efficiently by using bidirectional multigrid methods. Further, Mainberger et al. demonstrated in [8] that a careful tuning of the data  $\mathbf{f}$  can lead to large quality gains in the reconstruction, e.g. one seeks data  $\mathbf{g}$  such that solutions of

$$\left( \text{diag}(\mathbf{c}) + \left( \mathbf{I} - \text{diag}(\mathbf{c}) \right) (-\mathbf{A}) \right) \mathbf{u} = \text{diag}(\mathbf{c}) \mathbf{g} \quad (6)$$

are as close to our desired output  $\mathbf{f}$  as possible. Related investigations can also be found in [18], where the authors present subdivision strategies that exploit nonlinear PDEs. If the underlying diffusion process is based on a linear operator, then the optimisation can be formulated as a linear least squares problem by considering

$$\mathbf{g} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \frac{1}{2} \left\| \left( \text{diag}(\mathbf{c}) + \left( \mathbf{I} - \text{diag}(\mathbf{c}) \right) (-\mathbf{A}) \right)^{-1} \text{diag}(\mathbf{c}) \mathbf{x} - \mathbf{f} \right\|_2^2 \right\} \quad (7)$$

We refer to [8] for the original presentation of this model. In the context of nonlinear diffusion it is not possible to consider such a convex optimisation problem. Schmaltz et al. suggested in [6] to use clever searching strategies in this case.

To alleviate the upcoming discussion we introduce two definitions related to the just mentioned linear system needed for the reconstruction and the least squares problem required for the optimisation. We call *inpainting matrix* the following  $N \times N$  matrix

$$\mathbf{B}(\mathbf{c}) := \text{diag}(\mathbf{c}) + (\mathbf{I} - \text{diag}(\mathbf{c}))(-\mathbf{A}) \quad .$$

Further, if we have a mask  $\mathbf{c}$  to our avail for which the inpainting matrix is invertible, then we call the following  $N \times N$  matrix *reconstruction matrix*

$$\mathbf{M}(\mathbf{c}) := \mathbf{B}^{-1}(\mathbf{c}) \text{diag}(\mathbf{c}) \quad .$$

The exact requirements for the existence of  $\mathbf{B}^{-1}(\mathbf{c})$  will be covered in future work. For the moment we simply assume that this matrix exists. Using these definitions, we can rewrite the linear system (5) as

$$\mathbf{B}(\mathbf{c})\mathbf{u} = \text{diag}(\mathbf{c})\mathbf{f} \quad \Leftrightarrow \quad \mathbf{u} = \mathbf{M}(\mathbf{c})\mathbf{f} \quad (8)$$

and the grey value optimisation problem from (7) takes the form

$$\mathbf{g} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\mathbf{M}(\mathbf{c})\mathbf{x} - \mathbf{f}\|_2^2 \right\} \quad (9)$$

In order to quantify the quality of the results obtained from the inpainting we introduce the *reconstruction error* which simply measures the  $\ell_2$  distance between the reconstruction and the initially specified data. We denote it by

$$E(\mathbf{c}, \mathbf{g}) := \frac{1}{2} \|\mathbf{M}(\mathbf{c})\mathbf{g} - \mathbf{f}\|_2^2 \quad (10)$$

Note that the reconstruction error is simply a rescaled variant of the popular mean square error frequently used for error measures. We will use the reconstruction error as it is more directly related to the optimisation problem to be analysed in this paper.

### 3 Optimisation in the Co-domain

Let us introduce some further notations and definitions relevant for the forthcoming paragraphs. For the sake of simplicity we assume that all  $N$  pixels in our image have been labelled by a single index. Thus, the individual pixel locations are given by the set  $J := \{1, \dots, N\}$ . Further, we assume that the mask positions have been fixed beforehand and cannot be altered anymore. Also we require that the mask is not empty. We denote the set of mask positions by  $K \subseteq J$ . Clearly, it follows that  $c_i = 0$  for all  $i \in J \setminus K$ . For  $i \in K$  we are left with three possibilities. Either we fix the mask value  $c_i$  for all  $i \in K$  and manipulate the pixel value  $g_i$  to improve the reconstruction, or we fix  $g_i$  and optimise the value of  $c_i$ . Lastly we could also try to optimise both  $g_i$  and  $c_i$  for all  $i \in K$ . In this paper we are interested in the first two special cases. Setting  $c_i = 1$  for all  $i \in K$  and optimising  $\mathbf{g}$  yields the tonal optimisation problem described in [8]. Fixing  $\mathbf{g} = \mathbf{f}$  and optimising  $\mathbf{c}$  is related to the strategies from [10], even though the approach there did not require the support of  $\mathbf{c}$  to be specified beforehand. The question arises which of these two frameworks yields the smaller error. Both methods can only influence the reconstruction at locations indicated by the set  $K$ . Both optimisation strategies can be reduced to a system of  $|K|$  equations although these are only linear if we optimise  $\mathbf{g}$ . In order to analyse these problems let us denote by  $\bar{\mathbf{c}}$  the following mask:

$$\bar{c}_i := \begin{cases} 1, & i \in K \\ 0, & i \notin K \end{cases} \quad (11)$$

Then we can reformulate the two previously described settings as the following optimisation problems.

$$\mathbf{g} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \{E(\tilde{\mathbf{c}}, \mathbf{x})\} \quad \text{and} \quad \tilde{\mathbf{c}} = \arg \min_{c_i, i \in K} \{E(\mathbf{c}, \mathbf{f})\} \quad (12)$$

Let us emphasise, that the optimisation is always to be understood as unconstrained. We do not restrict the range of values that the mask or the data takes. The necessary conditions for a minimum of  $E$  with respect to  $\mathbf{g}$  (resp.  $\mathbf{c}$ ) are given by

$$\frac{\partial}{\partial g_i} E(\tilde{\mathbf{c}}, \mathbf{g}) = 0, \quad \forall i \in J \quad \text{resp.} \quad \frac{\partial}{\partial c_i} E(\mathbf{c}, \mathbf{f}) = 0, \quad \forall i \in K \quad (13)$$

In order to analyse the potential benefits of optimising the mask values or the grey values we need analytic representations of the gradient of  $E$  with respect to each of its variables. To this end we adapt a result from Ochs et al. [14] (Lemma 9). There, the authors stated it for the case  $\mathbf{x} = \mathbf{f}$ . We refer to the original work for the proof.

**Proposition 1 (Gradients of the Reconstruction Error).** *The gradients of the reconstruction error with respect to its two arguments are given by*

$$\nabla_{\mathbf{c}} E(\mathbf{c}, \mathbf{x}) = \text{diag}(\mathbf{x} - (\mathbf{I} + \mathbf{A})\mathbf{M}(\mathbf{c})\mathbf{x})\mathbf{B}^{-\top}(\mathbf{c}) (\mathbf{M}(\mathbf{c})\mathbf{x} - \mathbf{f}) , \quad (14)$$

$$\nabla_{\mathbf{x}} E(\mathbf{c}, \mathbf{x}) = \mathbf{M}^{\top}(\mathbf{c}) (\mathbf{M}(\mathbf{c})\mathbf{x} - \mathbf{f}) . \quad (15)$$

Note that both gradients of  $E$  have a certain similarity. If we denote

$$\mathbf{T} := \mathbf{B}^{-\top}(\mathbf{c}) (\mathbf{B}^{-1}(\mathbf{c}) \text{diag}(\mathbf{c})\mathbf{x} - \mathbf{f}) , \quad (16)$$

then we have

$$\nabla_{\mathbf{c}} E(\mathbf{c}, \mathbf{x}) = \text{diag}(\mathbf{x} - (\mathbf{I} + \mathbf{A})\mathbf{B}^{-1}(\mathbf{c}) \text{diag}(\mathbf{c})\mathbf{x})\mathbf{T} , \quad (17)$$

$$\nabla_{\mathbf{x}} E(\mathbf{c}, \mathbf{x}) = \text{diag}(\mathbf{c})\mathbf{T} . \quad (18)$$

Assume now that for fixed mask positions  $K$  we have found the optimal mask values  $\tilde{\mathbf{c}}$  for the reconstruction with respect to the original data  $\mathbf{f}$ . This means we have

$$(\nabla_{\mathbf{c}} E(\mathbf{c}, \mathbf{f})|_{\mathbf{c}=\tilde{\mathbf{c}}})_i = 0 \quad \forall i \in K . \quad (19)$$

Inserting the expression from (14) into (19) yields

$$\left( \text{diag}(\mathbf{f} - (\mathbf{I} + \mathbf{A})\mathbf{M}(\tilde{\mathbf{c}})\mathbf{f})\mathbf{B}^{-\top}(\tilde{\mathbf{c}}) (\mathbf{M}(\tilde{\mathbf{c}})\mathbf{f} - \mathbf{f}) \right)_i = 0 \quad \forall i \in K . \quad (20)$$

The previous equation is a product between a diagonal matrix and a vector. This comes down to a componentwise multiplication between the diagonal entries of the matrix and the vectors entries. Therefore, at least one of the two following equations must hold:

$$(\mathbf{f} - (\mathbf{I} + \mathbf{A})\mathbf{M}(\tilde{\mathbf{c}})\mathbf{f})_{i \in K} = 0 , \quad (21)$$

$$(\mathbf{B}^{-\top}(\tilde{\mathbf{c}}) (\mathbf{M}(\tilde{\mathbf{c}})\mathbf{f} - \mathbf{f}))_{i \in K} = 0 . \quad (22)$$

Our goal is to show that the second equation actually always holds for all  $i \in K$ . If for a certain entry  $i \in K$ , the first equation differs from 0, then the second one must be 0. Thus, we only need to show that the first equation can never hold. To this end, note that  $\mathbf{u} := \mathbf{M}(\tilde{\mathbf{c}}) \mathbf{f}$  solves by definition the equation

$$\text{diag}(\tilde{\mathbf{c}}) (\mathbf{u} - \mathbf{f}) - (\mathbf{I} - \text{diag}(\tilde{\mathbf{c}})) \mathbf{A} \mathbf{u} = \mathbf{0} \quad (23)$$

and that (21) is equivalent to

$$\text{diag}(\tilde{\mathbf{c}}) (\mathbf{f} - (\mathbf{I} + \mathbf{A}) \mathbf{M}(\tilde{\mathbf{c}}) \mathbf{f}) = \mathbf{0} . \quad (24)$$

From (23) it follows that

$$\text{diag}(\tilde{\mathbf{c}}) (\mathbf{u} - \mathbf{f} + \mathbf{A} \mathbf{u}) = \mathbf{A} \mathbf{u} . \quad (25)$$

Plugging (25) into (24) yields the requirement  $-\mathbf{A} \mathbf{u} = \mathbf{0}$ . Thus, if (21) would hold, then the reconstruction  $\mathbf{u} = \mathbf{M}(\tilde{\mathbf{c}}) \mathbf{f}$  would also solve  $\mathbf{A} \mathbf{u} = \mathbf{0}$ . This would contradict our assumption that the inpainting mask  $\tilde{\mathbf{c}}$  is nonempty. Therefore, (21) can never hold.

Similarly as for (24), we note that (22) can be extended to all indices  $i \in K$  by multiplying it from the left with  $\text{diag}(\tilde{\mathbf{c}})$ . This gives us

$$\text{diag}(\tilde{\mathbf{c}}) \mathbf{B}^{-\top} (\tilde{\mathbf{c}}) (\mathbf{M}(\tilde{\mathbf{c}}) \mathbf{f} - \mathbf{f}) = 0$$

which implies that

$$\nabla_{\mathbf{x}} E(\tilde{\mathbf{c}}, \mathbf{x}) \Big|_{\mathbf{x}=\mathbf{f}} = \mathbf{0} . \quad (26)$$

The previous equation implies that if we have found optimal mask values, then all necessary optimality conditions with respect to the mask values and with respect to the data values are fulfilled.

Conversely, we could also set  $c_i = 1$  for all  $i \in K$  to obtain a mask  $\bar{\mathbf{c}}$  and optimise the grey values for reconstruction. This yields the requirement

$$\begin{aligned} \nabla_{\mathbf{x}} E(\bar{\mathbf{c}}, \mathbf{x}) &= \mathbf{0} \\ \Leftrightarrow \text{diag}(\bar{\mathbf{c}}) \mathbf{B}^{-\top} (\bar{\mathbf{c}}) (\mathbf{B}^{-1} (\bar{\mathbf{c}}) \text{diag}(\bar{\mathbf{c}}) \mathbf{x} - \mathbf{f}) &= \mathbf{0}, \\ \Leftrightarrow (\mathbf{B}^{-\top} (\bar{\mathbf{c}}) (\mathbf{B}^{-1} (\bar{\mathbf{c}}) \text{diag}(\bar{\mathbf{c}}) \mathbf{x} - \mathbf{f}))_i &= 0, \quad \forall i \in K \end{aligned} \quad (27)$$

Assume that we are in possession of optimal data  $\mathbf{g}$  for given  $\bar{\mathbf{c}}$  such that (27) holds. In combination with (16), it follows then that we have

$$(\nabla_{\mathbf{c}} E(\mathbf{c}, \mathbf{g}) \Big|_{\mathbf{c}=\bar{\mathbf{c}}})_i = 0 \quad \forall i \in K$$

Thus, if we have a binary mask to our avail with optimised tonal values, then it follows again that all necessary optimality conditions are fulfilled. We summarise the previous results in the following theorem.

**Theorem 1 (Fulfilment of Optimality Conditions).** *Non-binary optimisation of the mask values while keeping the grey values fixed at the original data*

yields a pair of data that fulfils all necessary optimality conditions for minimising the error of the reconstruction. Similarly, fixing a binary sparsity pattern for the inpainting mask and optimising the grey values also returns a pair of data that fulfils all necessary optimality conditions for minimising the error of the reconstruction.

Ultimately we would like to show that the reconstruction error is the same regardless of whether we optimise the mask  $\mathbf{c}$  and keep the data fixed or whether we optimise the data and enforce a binary inpainting mask. In order to show this, we need to prove that

$$E(\tilde{\mathbf{c}}, \mathbf{f}) = E(\bar{\mathbf{c}}, \mathbf{g}) . \quad (28)$$

To this end let an optimal mask  $\tilde{\mathbf{c}}$ , such that  $E(\tilde{\mathbf{c}}, \mathbf{f})$  is minimal, be given and assume that there exists a vector  $\bar{\mathbf{g}}$  such that the reconstruction is the same with the binary mask  $\bar{\mathbf{c}}$  corresponding to  $\tilde{\mathbf{c}}$ . Thus, we have

$$\mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} = \mathbf{M}(\tilde{\mathbf{c}})\mathbf{f} . \quad (29)$$

By applying the definition of  $\mathbf{M}(\bar{\mathbf{c}})$  we obtain the following analytic expression for  $\bar{\mathbf{g}}$ :

$$\bar{\mathbf{g}} = \text{diag}(\bar{\mathbf{c}})\mathbf{B}(\bar{\mathbf{c}})\mathbf{M}(\tilde{\mathbf{c}})\mathbf{f} \quad (30)$$

For a given mask  $\tilde{\mathbf{c}}$  the right-hand side can always be computed provided that  $\mathbf{B}^{-1}(\tilde{\mathbf{c}})$  exists. In order to show that grey value optimisation comes with no loss compared to mask optimisation we have to show that the pair  $(\bar{\mathbf{c}}, \bar{\mathbf{g}})$  from (30) satisfies the normal equations (15). Thus, we have to show that

$$\mathbf{M}^\top(\bar{\mathbf{c}})(\mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} - \mathbf{f}) = \mathbf{0} \quad (31)$$

An essential observation in the verification of (31) is that  $\bar{\mathbf{c}}$  and  $\tilde{\mathbf{c}}$  have the same sparsity pattern, i.e.  $\bar{\mathbf{c}}_i = 1 \Leftrightarrow \tilde{\mathbf{c}}_i \neq 0$  and  $\bar{\mathbf{c}}_i = 0 \Leftrightarrow \tilde{\mathbf{c}}_i = 0$ . This implies that for the kernels we obtain  $\ker(\text{diag}(\bar{\mathbf{c}})) = \ker(\text{diag}(\tilde{\mathbf{c}}))$  and thus  $\ker(\mathbf{M}(\bar{\mathbf{c}})) = \ker(\mathbf{M}(\tilde{\mathbf{c}}))$ , too. Further, we note that for any linear operator  $\mathbf{K}$  from  $\mathbb{R}^n$  to  $\mathbb{R}^n$  we have  $\ker(\mathbf{K}^\top) = \text{ran}(\mathbf{K})^\perp$ , where  $\text{ran}$  denotes the range of the operator. Combining this identity with the first isomorphism theorem yields

$$\begin{aligned} \ker(\mathbf{M}^\top(\bar{\mathbf{c}})) &= (\text{ran}(\mathbf{M}(\bar{\mathbf{c}})))^\perp \simeq (\mathbb{R}^n / \ker(\mathbf{M}(\bar{\mathbf{c}})))^\perp \\ &= (\mathbb{R}^n / \ker(\mathbf{M}(\tilde{\mathbf{c}})))^\perp \simeq (\text{ran}(\mathbf{M}(\tilde{\mathbf{c}})))^\perp = \ker(\mathbf{M}^\top(\tilde{\mathbf{c}})) \end{aligned} \quad (32)$$

The importance of this identity will become clear in a moment. By assumption,  $\tilde{\mathbf{c}}$  was chosen optimal. This implies  $\nabla_{\mathbf{c}}E(\tilde{\mathbf{c}}, \mathbf{f}) = \mathbf{0}$ . Because of Theorem 1 it follows that  $\nabla_{\mathbf{x}}E(\tilde{\mathbf{c}}, \mathbf{f}) = \mathbf{0}$  is also true. Expanding this equation and using (29) gives us

$$\mathbf{M}^\top(\bar{\mathbf{c}})(\mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} - \mathbf{f}) = \mathbf{0} . \quad (33)$$

Two possibilities exist. Either  $\mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} - \mathbf{f} = \mathbf{0}$  in which case (31) holds trivially, or  $\mathbf{0} \neq \mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} - \mathbf{f} \in \ker(\mathbf{M}^\top(\bar{\mathbf{c}}))$ . From (32) it follows that  $\mathbf{M}(\bar{\mathbf{c}})\bar{\mathbf{g}} - \mathbf{f} \in \ker(\mathbf{M}^\top(\bar{\mathbf{c}}))$  and thus (31) is fulfilled, too. We conclude that our vector  $\bar{\mathbf{g}}$  contains the optimal grey values for a binary mask. We summarise our findings in the following theorem.

**Theorem 2 (Equivalence between Tonal and Spatial Optimisation).**  
 Let  $\tilde{\mathbf{c}}$  be a solution of

$$\min_{c_i, i \in K} \{E(\mathbf{c}, \mathbf{f})\} \quad (34)$$

and assume that  $\mathbf{B}^{-1}(\tilde{\mathbf{c}})$  exists. Then the vector  $\bar{\mathbf{g}}$  given by (30) solves

$$\min_{\mathbf{x}} \{E(\bar{\mathbf{c}}, \mathbf{x})\} \quad (35)$$

where  $\bar{\mathbf{c}}$  is the binary mask corresponding to  $\tilde{\mathbf{c}}$  and  $E(\tilde{\mathbf{c}}, \mathbf{f}) = E(\bar{\mathbf{c}}, \bar{\mathbf{g}})$ , i.e. the reconstruction error is the same in each case.

We note that the preceding theory also gives us an analytic expression for the optimal grey values in (30) in terms of optimal mask values.

## 4 Fast and Efficient Tonal Optimisation

In the previous section we have shown that a tonal optimisation comes with no loss compared to non-binary mask optimisation. Nevertheless, finding the best mask values is a tedious non-convex optimisation task whereas the grey value optimisation problem is a convex least squares problem. The latter family of problems is well studied and many highly efficient strategies exist. In this section we present two fast methods that allow an efficient computation of the perfect tonal values without having to resort to (30) and optimal mask values. Let us remark that our cost function  $E(\mathbf{c}, \cdot)$  is convex but not strictly convex. Indeed the reconstruction matrix  $\mathbf{M}(\mathbf{c})$  is only invertible if  $c_i = 1$  for all  $i$ . Further, it is easy to see that usually there exist infinitely many minimisers of  $E(\mathbf{c}, \cdot)$ . If  $\mathbf{g}$  is a minimiser, then we can arbitrarily change any entry  $i$  of  $\mathbf{g}$  where  $c_i = 0$ .

In the following we present two strategies. The first one is well suited for implementations on a CPU, whereas the second one exploits the massive parallelism provided by modern GPUs.

### 4.1 LSQR Approach

The venerable LSQR algorithm [19, 20] is a highly efficient method to solve general least squares problems of the form

$$\arg \min_{\mathbf{x} \in \mathbb{R}^n} \{\|\mathbf{K}\mathbf{x} - \mathbf{b}\|_2\} \quad (36)$$

with a large, sparse and unsymmetric matrix  $\mathbf{K}$ . The underlying iterative process applies the bidiagonalisation process of Golub and Kahan [21] and decreases the norm of the residual in each step. Although the algorithm generates a sequence of iterates that has the same properties as those from standard conjugate gradient methods it tends to behave much better in numerically ill-posed situations. Further, it is easy to implement, and it only requires the matrix  $\mathbf{K}$  for computing matrix vector products of the form  $\mathbf{K}\mathbf{u}$  and  $\mathbf{K}^\top \mathbf{v}$  for various vectors  $\mathbf{u}$  and  $\mathbf{v}$ . In presence of routines capable of computing these products efficiently, it is

not even necessary to know the matrix explicitly. This fact makes the algorithm attractive for solving (12). The adaptation is straightforward. It suffices to set  $\mathbf{K} = \mathbf{M}(\mathbf{c})$  in (36). For our setting we have

$$\begin{aligned} \mathbf{y} &= \mathbf{M}(\mathbf{c}) \mathbf{x} &\Leftrightarrow &\mathbf{B}(\mathbf{c}) \mathbf{y} = \text{diag}(\mathbf{c}) \mathbf{x} , \\ \mathbf{y} &= \mathbf{M}^\top(\mathbf{c}) \mathbf{x} &\Leftrightarrow &\mathbf{B}^\top(\mathbf{c}) \mathbf{z} = \mathbf{x}, \quad \mathbf{y} = \text{diag}(\mathbf{c}) \mathbf{z} . \end{aligned} \quad (37)$$

The linear systems  $\mathbf{B}(\mathbf{c}) \mathbf{y} = \text{diag}(\mathbf{c}) \mathbf{x}$  and  $\mathbf{B}^\top(\mathbf{c}) \mathbf{z} = \mathbf{x}$  can be solved in a highly efficient manner with either the multigrid methods from [7] or the multifrontal sparse LU decomposition from [22–24]. For the sparse LU solver the decomposition of the matrix  $\mathbf{B}(\mathbf{c})$  needs only be done once during the first iteration of the LSQR algorithm. Forthcoming iterations can then be computed at almost no additional cost. This yields an extremely fast strategy. The complete algorithm is depicted in Algorithm 1.

---

**Algorithm 1:** Tonal optimisation with the LSQR Algorithm.

---

**Input** : Reconstruction matrix  $\mathbf{M}(\mathbf{c})$ , data  $\mathbf{f}$ , number of iteration  $N$   
**Output** : Solution of the least squares problem (12)  $\mathbf{x}_N$   
**Initialise** :  $\bar{\mathbf{u}}_1 = \mathbf{b}$ ,  $\beta_1 = \|\bar{\mathbf{u}}_1\|$ ,  $\mathbf{u}_1 = \beta_1^{-1} \bar{\mathbf{u}}_1$ ,  $\bar{\mathbf{v}}_1 = \mathbf{M}^\top(\mathbf{c}) \mathbf{u}_1$ ,  $\alpha_1 = \|\bar{\mathbf{v}}_1\|$ ,  
 $\mathbf{v}_1 = \alpha_1^{-1} \bar{\mathbf{v}}_1$ ,  $\mathbf{w}_1 = \mathbf{v}_1$ ,  $\mathbf{x}_0 = \mathbf{0}$ ,  $\bar{\phi}_1 = \beta_1$ ,  $\bar{\rho}_1 = \alpha_1$

1 for  $k$  from 1 to  $N$  do

2      $\bar{\mathbf{u}}_{k+1} = \mathbf{M}(\mathbf{c}) \mathbf{v}_k - \alpha_k \mathbf{u}_k$ ,  $\beta_{k+1} = \|\bar{\mathbf{u}}_{k+1}\|$ ,  $\mathbf{u}_{k+1} = \beta_{k+1}^{-1} \bar{\mathbf{u}}_{k+1}$

3      $\bar{\mathbf{v}}_{k+1} = \mathbf{M}(\mathbf{c})^\top \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{v}_k$ ,  $\alpha_{k+1} = \|\bar{\mathbf{v}}_{k+1}\|$ ,  $\mathbf{v}_{k+1} = \alpha_{k+1}^{-1} \bar{\mathbf{v}}_{k+1}$

4      $\rho_k = \sqrt{\bar{\rho}_k^2 + \beta_{k+1}^2}$ ,  $c_k = \bar{\rho}_k / \rho_k$ ,  $s_k = \beta_{k+1} / \rho_k$

5      $\theta_{k+1} = s_k \alpha_{k+1}$ ,  $\bar{\rho}_{k+1} = -c_k \alpha_{k+1}$ ,  $\phi_k = c_k \bar{\phi}_k$ ,  $\bar{\phi}_{k+1} = s_k \bar{\phi}_k$

6      $\mathbf{x}_{k+1} = \mathbf{x}_{k-1} + (\phi_k / \rho_k) \mathbf{w}_k$

7      $\mathbf{w}_{k+1} = \mathbf{v}_{k+1} - (\theta_{k+1} / \rho_k) \mathbf{w}_k$

8 end

---

## 4.2 Primal Dual Method

Alternatively to the LSQR algorithm, we may also apply primal dual approaches that have enjoyed an increasing popularity in the previous years, especially in the domain of image processing. Starting from (12) we rewrite the optimisation problem by introducing a dummy variable  $\mathbf{d}$  and enforce that  $\mathbf{d}$  coincides with our reconstruction  $\mathbf{M}(\mathbf{c}) \mathbf{x}$ . Using the indicator function  $\iota_{\{\mathbf{0}\}}$  defined as

$$\iota_{\{\mathbf{0}\}}(\mathbf{x}) := \begin{cases} 0, & \mathbf{x} = \mathbf{0} \\ \infty, & \mathbf{x} \neq \mathbf{0} \end{cases} \quad (38)$$

we can reformulate our task in the following way:

$$\arg \min_{\mathbf{x}, \mathbf{d} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\mathbf{d} - \mathbf{f}\|_2^2 + \iota_{\{\mathbf{0}\}}(\mathbf{d} - \mathbf{M}(\mathbf{c}) \mathbf{x}) \right\} . \quad (39)$$



Note that  $\mathbf{d} = \mathbf{M}(\mathbf{c}) \mathbf{x}$  if and only if  $\mathbf{B}(\mathbf{c}) \mathbf{d} = \text{diag}(\mathbf{c}) \mathbf{x}$ . Thus, (39) is equivalent to

$$\arg \min_{\mathbf{x}, \mathbf{d} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\mathbf{d} - \mathbf{f}\|_2^2 + \iota_{\{0\}}(\mathbf{B}(\mathbf{c}) \mathbf{d} - \text{diag}(\mathbf{c}) \mathbf{x}) \right\}. \quad (40)$$

The benefit of the latter equation is that we have eliminated the inverse  $\mathbf{B}^{-1}(\mathbf{c})$  by introducing  $\mathbf{B}(\mathbf{c})$  at another position. Equation (40) can be efficiently handled with the algorithm presented in [25]. Applying the primal dual method from [25] only requires the evaluation of  $\mathbf{B}(\mathbf{c}) \mathbf{u}$  and  $\mathbf{B}^\top(\mathbf{c}) \mathbf{v}$  for vectors  $\mathbf{u}$  and  $\mathbf{v}$ . Since the matrix  $\mathbf{B}(\mathbf{c})$  is structured and extremely sparse, these computations can be handled in an efficient manner, leading to a high performing grey value optimisation strategy. A straightforward application of Algorithm 1 from [25] with  $G(\mathbf{x}) := \frac{1}{2} \|\mathbf{x} - \mathbf{f}\|_2^2$  and  $F(\mathbf{x}) = \iota_{\{0\}}(\mathbf{x})$  gives us the simple iterative strategy shown in Algorithm 2.

---

**Algorithm 2:** Tonal optimisation with primal dual methods.

---

**Input** :  $N$  the number of iterations.  
**Output** : Vectors  $\mathbf{x}^{N+1}$  and  $\mathbf{d}^{N+1}$  solving (40)  
**Initialise** :  $\tau, \sigma > 0$  such that  $\sigma\tau \|(\mathbf{B}(\mathbf{c}) - \text{diag}(\mathbf{c}))\|_2^2 < 1$ ,  $\theta \in [0, 1]$ ,  
 $\mathbf{u}^0, \mathbf{c}^0, \mathbf{y}^0$  arbitrary,  $\hat{\mathbf{u}}^0 = \mathbf{u}^0$  and  $\hat{\mathbf{c}}^0 = \mathbf{c}^0$

- 1 **for**  $k$  **from** 1 **to**  $N$  **do**
- 2      $\mathbf{y}^{k+1} = \mathbf{y}^k + \sigma (\mathbf{B}(\mathbf{c}) \hat{\mathbf{d}}^k - \text{diag}(\mathbf{c}) \hat{\mathbf{x}}^k)$
- 3      $\mathbf{d}^{k+1} = (1 + \tau)^{-1} (\mathbf{d}^k - \tau (\mathbf{B}(\mathbf{c})^\top \mathbf{y}^{k+1} - \mathbf{f}))$
- 4      $\mathbf{x}^{k+1} = \mathbf{x}^k + \tau \text{diag}(\mathbf{c}) \mathbf{y}^{k+1}$
- 5      $\hat{\mathbf{d}}^{k+1} = \mathbf{d}^{k+1} + \theta (\mathbf{d}^{k+1} - \mathbf{d}^k)$
- 6      $\hat{\mathbf{x}}^{k+1} = \mathbf{x}^{k+1} + \theta (\mathbf{x}^{k+1} - \mathbf{x}^k)$
- 7 **end**

---

This algorithm is better suited for parallel implementations than Algorithm 1 since almost all operations are pointwise and do not depend on each other. Further it does not have to solve any linear systems of equations. Let us also remark that additional optimisations like preconditioning strategies, as presented in [26], could further improve the performance of Algorithm 2.

### 4.3 Performance Comparison

We compare the performance with respect to speed of our LSQR solver, the primal dual solver and the stochastic tonal optimisation method from [8]. The algorithms were implemented in Fortran and C and all the tests were done on a standard desktop PC with an Intel Xeon processor (3.2GHz) and 24GB of memory. We also used a Nvidia GeForce GTX 460 for the GPU experiments. The runtimes are depicted in Table 1. The represented timings are the averages of three runs for each test case. We used different sizes of the *trui* test image (see Figure 1). Due to spatial constraints we only give results for a single image. The performance for other images are analogous. For each image size we

**Table 1.** Speed comparison between the different algorithms for tonal optimisation on the CPU and GPU. The approach of Mainberger et al. from [8] performs worst on every image size and its runtime increases much faster for larger images than for the other two algorithms. The LSQR approach has the best runtimes on the CPU whereas the primal dual method excels on the GPU. The runtime for computing the mask positions is not included as it is the same for every method.

Image Size	Runtime CPU (seconds)			Runtime GPU
	Method from [8]	Algorithm 1	Algorithm 2	Algorithm 2
$32 \times 32$	7.99	0.44	1.37	1.04
$48 \times 48$	32.57	1.23	2.90	1.35
$64 \times 64$	156.33	2.69	5.82	1.28
$80 \times 80$	360.42	4.63	8.50	1.47
$96 \times 96$	783.87	7.72	14.89	2.30
$112 \times 112$	1633.82	12.02	35.86	2.60
$128 \times 128$	3116.70	18.73	52.57	3.33
$256 \times 256$	95832.64	113.07	260.26	9.02



**Figure 1.** Data used for the experimental setup with a corresponding reconstruction. Left: original ( $256 \times 256$ ), Center: binary mask, Right: reconstruction after tonal optimisation.

computed a binary inpainting mask using the optimal control framework from [10]. All masks have a density within the range of  $5.0 \pm 0.1\%$ . We used the algorithm from [8] as a reference method and compared how our algorithms perform in terms of speed. All algorithms converged towards the same solution. The method from [8] uses a multigrid solver for the computation of the inpainting echos. It stopped when the error between two iterates dropped below  $10^{-3}$ . Algorithm 1 stopped when the increment in the solution dropped in norm below  $10^{-10}$  whereas Algorithm 2 halted its execution when the update in any variable was smaller than  $10^{-15}$  in norm. These tolerances were chosen such that the resulting images always had the same reconstruction error.

The exceptional performance of the LSQR algorithm stems from the fact that it reached a convergent state within 10 to 30 iterations which implies that it requires less than 100 inpaintings, whereas the method from [8] has to compute an inpainting for every mask pixel during each iteration. While Algorithm 1 is

well suited for CPU implementations, the fact that most of the computations in Algorithm 2 can be done in parallel and that no linear systems must be solved render this algorithm attractive for GPUs.

## 5 Summary and Conclusions

We have shown an equivalence result for inpainting with the Laplace equation when the data positions are fixed: Grey value optimisation with binary masks is equivalent to non-binary mask optimisation. This finding justifies the post-processing step proposed in [10] where the optimal mask values were exchanged with optimal data values. Our results show that this strategy comes with no loss in the reconstruction quality. Further, it significantly reduces the amount of data to be stored for compression purposes and marks a significant step towards a fast PDE based data compression codec. Finally, we have suggested two efficient algorithms to solve the tonal optimisation problem on the CPU and on the GPU.

It remains an open question whether a combined and simultaneous optimisation of the mask and the interpolation data can yield an even better reconstruction. The analysis of this problem as well as the development of a competitive image compression codec will be the subject of future work.

## References

1. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Proc. of the International Conference on Image Processing, vol. 3, pp. 259–263. IEEE (1998)
2. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. SIGGRAPH, pp. 417–424. ACM Press/Addison-Wesley Publishing Company, New Orleans, LI (2000)
3. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.-P.: Towards PDE-based image compression. In: Paragios, N., Faugeras, O., Chan, T., Schnörr, C. (eds.) VLSM 2005. LNCS, vol. 3752, pp. 37–48. Springer, Heidelberg (2005)
4. Liu, D., Sun, X., Wu, F., Li, S., Zhang, Y.Q.: Image compression with edge-based inpainting. *IEEE Transactions on Circuits, Systems and Video Technology* 7(10), 1273–1286 (2007)
5. Belhachmi, Z., Bucur, D., Burgeth, B., Weickert, J.: How to choose interpolation data in images. *SIAM Journal on Applied Mathematics* 70(1), 333–352 (2009)
6. Schmaltz, C., Weickert, J., Bruhn, A.: Beating the quality of JPEG 2000 with anisotropic diffusion. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 452–461. Springer, Heidelberg (2009)
7. Mainberger, M., Bruhn, A., Weickert, J., Forchhammer, S.: Edge-based compression of cartoon-like images with homogeneous diffusion. *Pattern Recognition* 44(9), 1859–1873 (2011)
8. Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for homogeneous diffusion inpainting. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 26–37. Springer, Heidelberg (2012)
9. Bourquard, A., Unser, M.: Anisotropic interpolation of sparse generalized image samples. *IEEE Transactions on Image Processing* 22(2), 459–472 (2013)

10. Hoeltgen, L., Setzer, S., Weickert, J.: An optimal control approach to find sparse data for Laplace interpolation. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) EMMCVPR 2013. LNCS, vol. 8081, pp. 151–164. Springer, Heidelberg (2013)
11. Chen, Y., Ranftl, R., Pock, T.: A bi-level view of inpainting - based image compression. Computing Research Repository (2014), <http://arxiv.org/abs/1401.4112v2>
12. Gomathi, R., Kumar, A.V.A.: A multiresolution image completion algorithm for compressing digital color images. *Journal of Applied Mathematics* 2014, Article ID 757318 (2014)
13. Köstler, H., Stürmer, M., Freundl, C., Rüdte, U.: PDE based video compression in real time. *Lehrstuhlbericht 07-11*, Friedrich-Alexander-Universität Erlangen-Nürnberg (2011)
14. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences* (to appear, 2014)
15. Ochs, P., Brox, T., Pock, T.: iPiasco: Inertial proximal algorithm for strongly convex optimization. Technical report, Universität Freiburg (2014)
16. Zaremba, S.: Sur un problème mixte relatif à l'équation de Laplace. *Bulletin de l'Académie des Sciences de Cracovie*, 313–344 (1910)
17. Gilbarg, D., Trudinger, N.: *Elliptic Partial Differential Equations of Second Order*. Springer (2001)
18. Galic, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision* 31(2-3), 255–269 (2008)
19. Paige, C.C., Saunders, M.A.: Algorithm 583; LSQR: Sparse linear equations and least-squares problems. *ACM Transactions on Mathematical Software* 8(2), 195–209 (1982)
20. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software* 8(1), 43–71 (1982)
21. Golub, G.H., Kahan, W.: Calculating the singular values and pseudoinverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics* 2(2), 205–224 (1965)
22. Davis, T.A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 30(2), 196–199 (2004)
23. Davis, T.A., Duff, I.S.: An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM Journal on Matrix Analysis and Applications* 18(1), 104–158 (1997)
24. Davis, T.A., Duff, I.S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Transactions on Mathematical Software* 25(1), 1–19 (1999)
25. Chambolle, A., Pock, T.: A first order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40(1), 120–145 (2011)
26. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: Metaxas, D., Quan, L., Sanfeliu, A., Van Gool, L. (eds.) 2011 International Conference on Computer Vision (ICCV 2011), pp. 1762–1769. IEEE (2011)

# Expected Patch Log Likelihood with a Sparse Prior

Jeremias Sulam and Michael Elad

Computer Science Department, Technion, Israel  
{jsulam,elad}@cs.technion.ac.il

**Abstract.** Image priors are of great importance in image restoration tasks. These problems can be addressed by decomposing the degraded image into overlapping patches, treating the patches individually and averaging them back together. Recently, the Expected Patch Log Likelihood (EPLL) method has been introduced, arguing that the chosen model should be enforced on the final reconstructed image patches. In the context of a Gaussian Mixture Model (GMM), this idea has been shown to lead to state-of-the-art results in image denoising and deblurring. In this paper we combine the EPLL with a sparse-representation prior. Our derivation leads to a close yet extended variant of the popular K-SVD image denoising algorithm, where in order to effectively maximize the EPLL the denoising process should be iterated. This concept lies at the core of the K-SVD formulation, but has not been addressed before due the need to set different denoising thresholds in the successive sparse coding stages. We present a method that intrinsically determines these thresholds in order to improve the image estimate. Our results show a notable improvement over K-SVD in image denoising and inpainting, achieving comparable performance to that of EPLL with GMM in denoising.

**Keywords:** K-SVD, EPLL, MAP, Sparse Representations, Image Restoration.

## 1 Introduction

Inverse problems in image processing consist of recovering an original image that has been degraded. Denoising, deblurring and inpainting are specific and common such examples. Put formally, these problems attempt to recover an underlying image  $\mathbf{x}$  given the measurement  $\mathbf{y}$  such that

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad (1)$$

where  $\mathbf{A}$  is a known linear operator and  $\mathbf{n}$  represents measurement noise, assumed to be independent and normally distributed. In dealing with this problem, it is common to work with image priors as regularizers and develop a Maximum a Posteriori (MAP) estimator for the unknown image  $\hat{\mathbf{x}}$ . This can be formulated as an optimization problem where we look for an estimate which is close enough

to the measured image while being likely under this prior. Most state of the art methods employ, either implicitly or explicitly, some prior knowledge of this form [6,10,8,4].

Learning specific priors from real data has shown to enable better performance under this approach [7,12]. However, this learning process is computationally hard and it is usually restricted to small dimensions, which leads naturally to the modeling of small image patches [1,15]. Such methods attempt to address the image restoration problem by breaking the image into small overlapping patches, solving their MAP estimate, and tiling the results back together by averaging them [6,4,3]. While this is a common and practical strategy, it is also known to cause visible texture-like artifacts in the final image. Recently, Zoran and Weiss [16] proposed a general framework based on the simple yet appealing idea that the *resulting final* patches should be likely under some specific prior, and not the intermediate ones. Their approach is based on maximizing the *Expected Patch Log Likelihood* (EPLL) which yields the average likelihood of a patch on the final image under some prior. This idea is general in the sense that it can be applied to any patch-based prior for which a MAP estimator can be formulated. In particular, the authors in [16] employed the classic Gaussian Mixture Model prior achieving state of the art results in image denoising and deblurring.

The concept of sparsity is a recurring idea in most state of the art restoration methods; namely, a natural signal or image patch can be well represented by a linear combination of a few atoms from a dictionary [2,8]. This leads to the natural question, could we use the EPLL framework with a sparsity-inspired prior? If so, how is this related to existing methods that explicitly target this problem and what is there to gain from this approach? In this paper we explore and formally address these questions, showing that indeed benefit can be found in employing EPLL with a patch sparsity-based prior.

## 2 Expected Patch Log Likelihood

We begin by briefly reviewing the EPLL framework as described in [16]. Given an image  $\mathbf{x}$ , the Expected Patch Log Likelihood under some prior  $p$  is defined as

$$EPLL_p(\mathbf{x}) = \sum_i \log p(\mathbf{P}_i \mathbf{x}), \quad (2)$$

where  $\mathbf{P}_i$  extracts the  $i^{\text{th}}$  patch from  $\mathbf{x}$ . Therefore, given the corruption model in Eq. (1) we can propose to minimize the following cost function:

$$f_p(\mathbf{x}|\mathbf{y}) = \frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 - EPLL_p(\mathbf{x}), \quad (3)$$

where the first term represents the log likelihood of the image. To get around the hard optimization of this function, the authors in [16] propose to use a *Half Quadratic Splitting* strategy by defining auxiliary patches  $\{\mathbf{z}^i\}$  for each patch  $\mathbf{P}_i \mathbf{x}$ , and then minimizing

$$c_{p,\beta}(\mathbf{x}, \{\mathbf{z}^i\}|\mathbf{y}) = \frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sum_i \frac{\beta}{2} \|\mathbf{P}_i \mathbf{x} - \mathbf{z}^i\|_2^2 - \log p(\mathbf{z}^i) \quad (4)$$

iteratively, while increasing the value of  $\beta$ . Note that for  $\beta \rightarrow \infty$ ,  $\mathbf{z}^i \rightarrow \mathbf{P}_i \mathbf{x}$ , so this parameter controls the distance between the auxiliary patches and the patches of the image  $\mathbf{x}$ . For a fixed value of  $\beta$ , the cost function is again broken into a two step inner minimization: first fix  $\{\mathbf{z}^i\}$  and solve for  $\mathbf{x}$  by

$$\mathbf{x} = \frac{\lambda \mathbf{A}^T \mathbf{y} + \beta \sum_i \mathbf{P}_i^T \mathbf{z}^i}{\lambda \mathbf{A}^T \mathbf{A} + \beta \sum_i \mathbf{P}_i^T \mathbf{P}_i}. \quad (5)$$

Then, fix  $\mathbf{x}$  and solve for  $\{\mathbf{z}^i\}$  by solving the MAP estimate for each patch under the prior in consideration. This process should be repeated 4-5 times, before increasing  $\beta$  and repeating the whole process again. Each time, the patches are taken from *the image estimate* at each iteration.

Within the EPLL scheme, the choice of  $\beta$  is crucial. In [16] the authors set this parameter manually to be  $\frac{1}{\sigma^2}[1, 4, 8, 16, 32, \dots]$ , where  $\sigma$  is the noise standard deviation. In the same work it is also suggested that  $\beta$  could be determined as  $\beta = \frac{1}{\sigma^2}$ , where  $\sigma$  is estimated in every iteration by an *off-the-shelf* white Gaussian noise estimator.

### 3 EPLL with a Sparse Prior

In the original formulation, Zoran and Weiss propose to use a Gaussian Mixture Model (GMM) prior which is learnt off-line from a large number of examples. In their case, the MAP estimator for each patch is simply given by the Wiener filter solution for the Gaussian component with the highest conditional weight [16]. However, the EPLL approach is a generic framework for potentially any patch-based prior. We now turn to explore the formulation of an equivalent problem with a sparsity inducing prior.

#### 3.1 Cost Function Formulation

Consider the signal  $\mathbf{z} = \mathbf{D}\alpha$ , where  $\mathbf{D}$  is a redundant dictionary of size  $n \times m$  ( $n < m$ ), and the vector  $\alpha$  is sparse; i.e.,  $\|\alpha\|_0 \ll n$ , where the  $l_0$  pseudo-norm  $\|\cdot\|_0$  basically counts the non zero elements in  $\alpha$ . Assuming that this is the model we impose on our patches  $\mathbf{z}^i$ , Eq. (4) becomes

$$c_{\mu, \beta}(\mathbf{x}, \{\alpha_i\} | \mathbf{y}) = \frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sum_i \frac{\beta}{2} \|\mathbf{D}\alpha_i - \mathbf{P}_i \mathbf{x}\|_2^2 + \mu_i \|\alpha_i\|_0. \quad (6)$$

In this case,  $\mu_i$  reflects the trade-off between the accuracy of the representation and the sparsity of  $\alpha_i$ . For the case  $\beta = 1$ , this last expression corresponds exactly to the formulation of the K-SVD denoising algorithm in [6], where  $\mathbf{A} = \mathbf{I}$ . In this work, Elad and Aharon proposed to use a block-coordinate minimization

---

<sup>1</sup> Note that this an abuse of notation as the denominator is a diagonal matrix to be inverted.

that starts by fixing  $\mathbf{x} = \mathbf{y}$ , and then seeking the optimal  $\alpha_i$  solving the MAP estimator for each patch:

$$\hat{\alpha}_i = \arg \min_{\alpha} \mu_i \|\alpha_i\|_0 + \|\mathbf{D}\alpha_i - \mathbf{P}_i\mathbf{x}\|_2^2. \quad (7)$$

Though this problem is NP-hard in general, its solution can be well approximated by greedy or pursuit algorithms [5]. In particular, the Orthogonal Matching Pursuit (OMP) [14] can be used with the noise energy as an error threshold to yield an approximation of the solution to Problem (7), and we employ this method in our work due to its simplicity and efficiency [13]. This way,  $\mu_i$  is handled implicitly by replacing the second term by a constraint of the form

$$\min_{\alpha} \|\alpha\|_0 \quad \text{subject to} \quad \|\mathbf{D}\alpha - \mathbf{P}_i\mathbf{x}\|_2^2 \leq nc\sigma^2, \quad (8)$$

where  $c$  is a constant factor set to 1.15 in [6]. Given the estimated sparse vectors  $\{\hat{\alpha}_i\}$ , the algorithm proceeds by updating for the unknown image  $\mathbf{x}$  which results in an equivalent expression to that in Eq. (5) - for a specific value of  $\beta$ . When denoising is done locally (training the dictionary on the corrupted patches) the dictionary gets updated together with the sparse vectors by using a K-SVD step. This adaptive method that trains the dictionary on the noisy image itself has proven to be better than using a dictionary trained offline.

The initial claim in [6] is that the above block-coordinate minimization should be iterated. In practice, however, repeating this process is problematic since after updating  $\mathbf{x}$ , the noise level has changed and it is spatially varying. Therefore, the sparse coding stage has no known thresholds to employ. Thus, the algorithm in [6] does not iterate after updating  $\mathbf{x}$ .

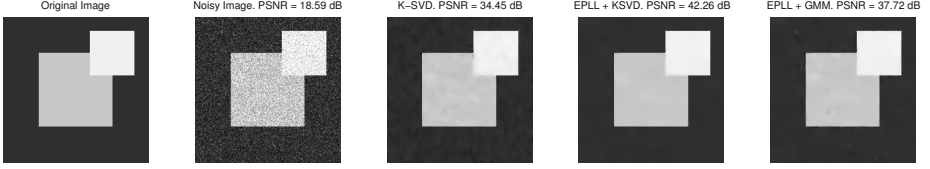
Increasing  $\beta$ , as practised in [16], forces the distance  $\|\mathbf{D}\alpha_i - \mathbf{P}_i\mathbf{x}\|_2$  to be smaller. Therefore, iterating the above algorithm for increasing values of  $\beta$  is equivalent to iterating the process described for the K-SVD with smaller thresholds. As we see, the algorithm proposed in [6] applies only the first iteration of the EPLL scheme with a sparse-enforcing prior, therefore losing important denoising potential. A synthetic example is shown in Fig. 1, where we compare the algorithms in [6] and [16] with the method proposed in this paper.

We now turn to address the matter of the threshold design for later stages of the K-SVD in order to practice the EPLL concept in an effective way.

### 3.2 Sparse Coding Thresholds

Consider the threshold in the sparse coding stage, at each iteration  $k$ , to be  $\nu_k^2$ . Naturally, in the first iteration of the process that aims to minimize Eq. (6) we set this threshold to be exactly the noise energy  $\sigma^2$  for all patches; i.e.  $\nu_1^2 = \sigma^2$ . In the following iterations, however, instead of trying to estimate the remaining noise with an *off-the-shelf* algorithm, we propose an intrinsic alternative by using the information we already have about each patch.





**Fig. 1.** Denoising of a synthetic image ( $\sigma = 30$ ). A similar demonstration was presented in [16], showing the benefits of the EPLL framework under a GMM approach. Note the texture-like resulting artifacts in the result by K-SVD. This problem is notably reduced by the EPLL with a Sparse Prior, the method we present in this work. We include for comparison the result by [16]. The evolution of the Peak Signal to Noise Ratios are depicted in Fig 4.

Consider the general problem of estimating the remaining noise after applying K-SVD on the noisy image; i.e., the first iteration of our method. From a global perspective, the estimated image can be expressed as

$$\hat{\mathbf{x}} = \frac{(\lambda \mathbf{A}^T + \sum_i \mathbf{P}_i^T \mathbf{D}_{S_i} \mathbf{D}_{S_i}^+ \mathbf{P}_i) \mathbf{y}}{\lambda \mathbf{A}^T \mathbf{A} + \sum_i \mathbf{P}_i^T \mathbf{P}_i}, \quad (9)$$

where  $S_i$  denotes the support of the sparse vector  $\hat{\alpha}_i$  chosen in the OMP, and  $\mathbf{D}_{S_i}$  is the set of the corresponding atoms in the dictionary. Leaving aside the selection of the support of each sparse vector, we can represent this operation by a linear operator as

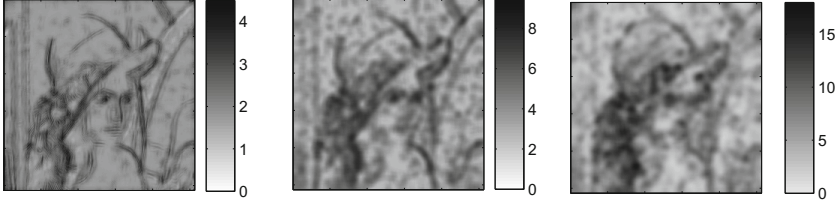
$$\hat{\mathbf{x}} = \mathbf{L}(\mathbf{x} + \mathbf{n}). \quad (10)$$

Assuming for a moment that  $\hat{\mathbf{x}} \approx \mathbf{L}\mathbf{x}$ , we could express the remaining noise as  $\mathbf{n}_r = \mathbf{L}\mathbf{n}$ , from which we could obtain the full covariance matrix as  $Cov(\mathbf{n}_r) = \sigma^2 \mathbf{L}\mathbf{L}^T$ . Then, we could either take into consideration the full covariance matrix, or make the simplifying assumption of white noise by considering just the diagonal of  $Cov(\mathbf{n}_r)$ . Though appealing, this approach does not work in practice because  $\|\hat{\mathbf{x}} - \mathbf{L}\mathbf{x}\|_2$  is considerably large, and thus the estimate of the remaining noise is considerably low. Also, note that  $\mathbf{L}$  is a band matrix of size  $N^2 \times N^2$ , where  $N$  is the number of pixels, and so the estimation of its covariance matrix is computationally intractable for practical purposes.

We thus turn to a similar but local alternative that will enable a practical solution. Each patch consists of the true underlying vector  $\mathbf{z}_{0i}$  and a noise component  $\mathbf{v}_i$ ,  $\mathbf{z}_i = \mathbf{z}_{0i} + \mathbf{v}_i$ . Given the chosen support  $S_i$ ,  $\hat{\mathbf{z}}_i$  is obtained as a projection onto the span of the selected atoms:

$$\hat{\mathbf{z}}_i = \mathbf{D}_{S_i} \mathbf{D}_{S_i}^+ \mathbf{z}_i = \mathbf{D}_{S_i} \mathbf{D}_{S_i}^+ (\mathbf{z}_{0i} + \mathbf{v}_i). \quad (11)$$

Assuming now that  $\mathbf{z}_{0i} \approx \mathbf{D}_{S_i} \mathbf{D}_{S_i}^+ \mathbf{z}_{0i}$  (if the correct support of the signal was chosen by the OMP), the contribution of the noise to the patch estimate would be given by  $\hat{\mathbf{v}}_i^r = \mathbf{D}_{S_i} \mathbf{D}_{S_i}^+ \mathbf{v}_i$ . This is an analogue assumption to that made for Eq. (10), but now for each patch instead of the global image. This way, considering



**Fig. 2.** Left: plot of the diagonal of the covariance matrix  $Cov(\mathbf{n}_r)$  after the first iteration of denoising the image Lena ( $\sigma = 20$ ). Center: the corresponding plot of the estimated  $\mathbf{R}^k$  in Eq. (13), and right: the corresponding average of the standard deviation per patch of the true error image.

the covariance matrix of the remaining noise  $Cov(\hat{\mathbf{v}}_i^r)$ , the mean squared error estimate at the  $i^{th}$  patch and iteration  $k$  will be given by  $\frac{1}{n}tr\{Cov(\hat{\mathbf{v}}_i^r)\}$ , leading to

$$(\hat{\sigma}_i^k)^2 = |S_i| \frac{\nu_k^2}{n}. \quad (12)$$

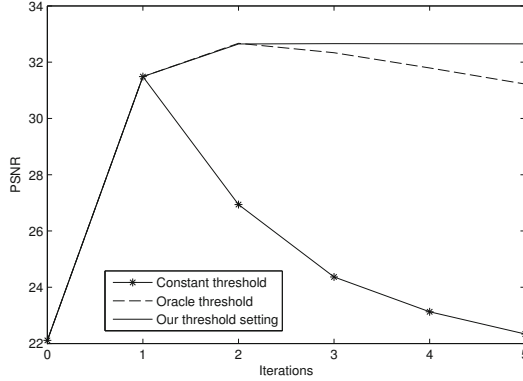
Therefore, the estimate of the remaining noise in each patch is simply proportional to the number of atoms used for that patch. Note that the remaining noise is no longer white after the back projection step, but we make this assumption in order to simplify further derivations.

Generalizing this patch analysis to the entire image, we can estimate the average remaining noise in the image  $\mathbf{x}$  by performing an estimate in the spirit of Eq. (5), tilling back and averaging the local estimates as

$$\mathbf{R}^k = \frac{\lambda \nu_k^2 \mathbf{I} + \sum_i \mathbf{P}_i^T \mathbf{1} (\hat{\sigma}_i^k)^2}{\lambda \nu_k^2 \mathbf{I} + \sum_i \mathbf{P}_i^T \mathbf{P}_i} = \Phi((\hat{\sigma}_i^k)^2), \quad (13)$$

where the operator  $\Phi(\cdot)$  relocates the local estimates  $\hat{\sigma}_i^k$  with the corresponding weighting. This way,  $\mathbf{R}^k$  stands for an estimation of the energy of the remaining noise pixel-wise, equivalent -but not equal- to the diagonal of  $Cov(\mathbf{n}_r)$ . An example is shown in Fig. 2 for the popular image Lena. We see that  $\mathbf{R}^k$  provides a fair estimate of the information in the diagonal of the full covariance matrix of the remaining noise  $Cov(\mathbf{n}_r)$ , and that it is closer to the average of the standard deviation per patch of the true error image. The reader should also note that computing  $\mathbf{R}^k$  is considerably cheaper than the computation of the operator in (10), since we only compute the local covariance matrices and their weighted average, and the matrix in the denominator of Eq. (13) is a diagonal one. Therefore we use  $\mathbf{R}^k$  to derive the threshold for the next iteration.

From this point two possibilities arise: use  $\mathbf{R}^k$  to evaluate a local patch-based noise energy, eventually denoising each patch with a different threshold, or finding a new global and common threshold for all the patches. Adopting the first alternative was found not to yield significant improvements in our results. Thus, in the following we adopt the second global alternative.



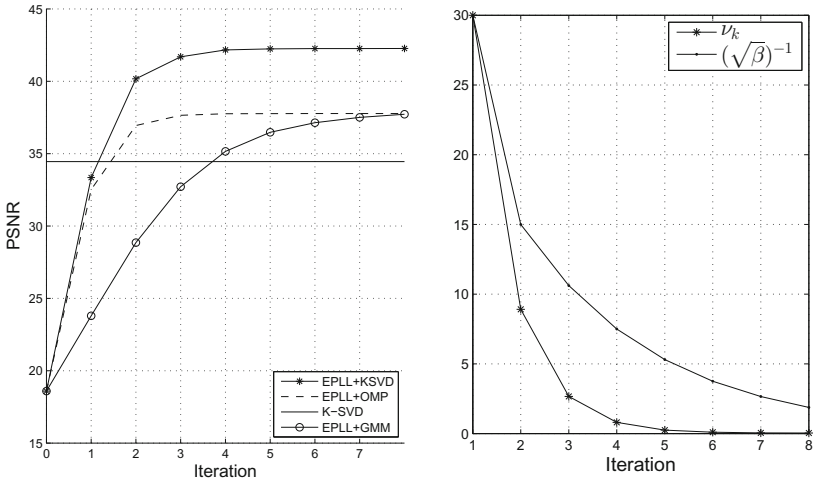
**Fig. 3.** PSNR evolution of the EPLL scheme with a sparse-representation prior for denoising the image Lena ( $\sigma = 20$ ) and three different threshold settings: a) using a constant threshold for all the iterations (equal to the initial noise energy  $\sigma^2$ ); b) using an oracle threshold by setting it to be the variance of the real error image (having access to the original image); and c) our threshold setting method.

The reader should bare in mind that the thresholds should tend to zero as we iterate, corresponding to  $\beta \rightarrow \infty$ . Certainly, this implies that our thresholds will not reflect the *real* remaining noise. As an example, in Fig. 3 we present the evolution of the PSNR by the proposed method for the image Lena for different thresholds. We see that if the threshold is not changed with the iterations, the PSNR of the resulting image  $\mathbf{x}$  decreases after the first iteration. On the other hand, if we set the threshold to be the variance of the real remaining noise (by having access to an oracle and the original image), the PSNR initially increases but eventually decreases since the threshold do not tend to zero. We include for comparison the results of our threshold-setting method.

This way, in what follows we propose a method that provides decreasing thresholds and which has been proven to be robust. In the subsequent iterations, we set the threshold  $\nu_k^2$  to be the mode of the values in  $\mathbf{R}^k$ . Furthermore, we have found that the multiplication by a constant factor  $\delta$  improves the performance in our method. To this end, assuming independence between the remaining noise and the patch estimate, and considering the residual per patch  $\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$ , we have that  $\tilde{\sigma}_i^2 = \sigma^2 - Var(\mathbf{r}_i)^2$ . With these estimates we can perform an analogue of Eq. (13) and obtain its mode,  $\tilde{\nu}^2$ . We then define the factor  $\delta = \tilde{\nu}^2 / \nu_k^2$ , and set the thresholds for the next iteration to be  $\nu_k^2 = \delta \cdot mode(\mathbf{R}^k)$ . A full description of our algorithm is depicted in Algorithm 1.

In the following iterations the assumption about the independence between the remaining noise and the patch estimate will be very weak, and so  $\tilde{\sigma}_i^2$  will not be accurate. Thus,  $\delta$  is determined after the first iteration only and kept fixed for the subsequent steps, while the estimate  $\nu_k^2$  provides decreasing estimates every time. An example of the obtained  $\nu_k^2$ 's can be seen in Fig. 4.

<sup>2</sup> The variance is calculated as  $Var(\mathbf{r}) = \frac{1}{n-1} \sum_j (\mathbf{r}_j - \bar{\mathbf{r}})^2$ , where  $\bar{\mathbf{r}}$  is the mean of  $\mathbf{r}$ .



**Fig. 4.** Left: PSNR evolution by EPLL with a sparsity inducing prior on the synthetic image in Fig. 1, compared to the original K-SVD algorithm [6] and the EPLL-GMM of [16]. Right: sequence of thresholds  $\nu_k$  determined by the proposed method and the equivalent  $1/\sqrt{\beta}$  by the method of [16].

## 4 Results

To gain some insight into the performance of our method and as a motivating example, in Fig. 1 we present the denoising results on a synthetic image obtained by the regular K-SVD algorithm, and the one achieved by applying the EPLL approach with the sparse-enforcing prior. A similar demonstration was presented in [16], and we include the results of this method as well. The K-SVD denoised image presents texture artifacts common to patch-based algorithms, while in the image denoised with our method the final patches are far more likely under the prior that we try to learn from the image itself.

Fig. 4 depicts the evolution of the PSNR of the denoised image in each iteration for this experiment. Note that given a fixed dictionary, solving the MAP estimate for each patch with a sparse prior implies applying OMP on each of them. This corresponds to the EPLL+OMP curve. On the other hand, we could minimize Eq. (8) w.r.t  $\mathbf{D}$  as well by applying a K-SVD step, updating the dictionary as well as the sparse vectors; this is the curve depicted as EPLL+K-SVD. The constant dotted line corresponds to the original K-SVD algorithm. Note that the result after the first iteration in our method is worse than the one obtained by K-SVD where  $c = 1.15$ . Choosing  $c = 1$  in our case, however, enables further improvement as we proceed maximizing the Expected Patch Log Likelihood. Notice also that our method converges in considerable fewer iterations than the method of [16]. The right side of Fig. 4 shows the evolution of the thresholds  $\nu_k$  used in the successive iterations, as well as the values  $1/\sqrt{\beta}$  used by EPLL-GMM.

The improvement obtained by training the dictionary in each iteration of our method is both important and intuitive. It is known that applying K-SVD on a

---

**Algorithm 1.** EPLL with a Sparse Prior, given the noisy image  $\mathbf{y}$  with a noise standard deviation of  $\sigma$  and an initial dictionary  $\mathbf{D}_0$ .

---

**Initialization:**  $\mathbf{x} = \mathbf{y}$ ,  $\mathbf{D} = \mathbf{D}_0$ ,  $\delta = 1$ ,  $k = 1$ ,  $\nu_k^2 = \sigma^2$ .

**for** *OuterIter* = 1 : 3 – 4 **do**

-  $\{\mathbf{D}^{k+1}, \mathbf{x}^{k+1}\} = \underset{\alpha_i, \mathbf{D}, \mathbf{x}}{\operatorname{argmin}} \lambda \|\mathbf{x}^k - \mathbf{y}\|_2^2 + \sum_i \|\mathbf{D}^k \alpha_i - \mathbf{P}_i \mathbf{x}^k\|_2^2 + \mu_i \|\alpha_i\|_0$ , by  
 K-SVD with error threshold  $\nu_k^2$ ;

- get local estimates  $(\hat{\sigma}_i^k)^2 = |S_i| \frac{\nu_k^2}{n}$ ,  $\forall i$ ;

- get global estimate  $\mathbf{R}^k = \Phi((\hat{\sigma}_i^k)^2)$  with Eq. (13);

**if**  $k = 1$  **then**

-  $\nu_{k+1}^2 = \operatorname{mode}(\mathbf{R}^k)$ ;

-  $\hat{\sigma}_i^2 = \sigma^2 - \operatorname{Var}(\mathbf{r}_i)$ ,  $\forall i$ ;

-  $\tilde{\nu}^2 = \operatorname{mode}(\Phi(\hat{\sigma}_i^2))$ ;

-  $\delta = \tilde{\nu}^2 / \nu_{k+1}^2$ ;

-  $\nu_{k+1}^2 = \delta \cdot \operatorname{mode}(\mathbf{R}^k)$ ;

-  $k = k + 1$ ;

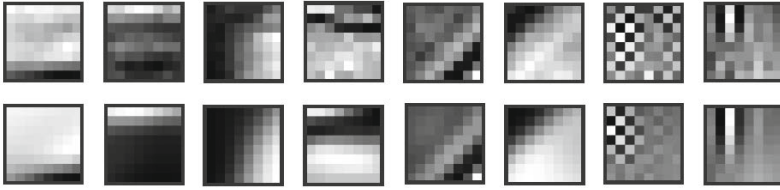
**Output:**  $\mathbf{x}, \mathbf{D}$ .

---

noisy image achieves good denoising results but yields somewhat noisy atoms [6]. By training the dictionary  $\mathbf{D}$  in the progressively cleaner estimates  $\mathbf{x}$  we obtain cleaner and more well defined atoms, which are later used to perform further denoising. In the top row of Fig. 5 we present 8 atoms trained on a noisy version of the image Lena after the first iteration, while the lower row shows the same atoms after 4 iterations.

## 4.1 Inpainting

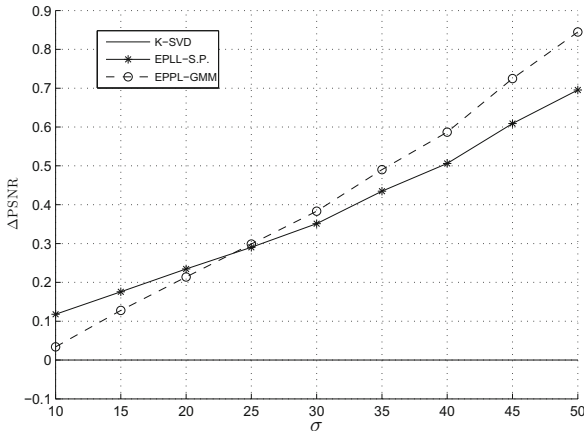
We next present results on image inpainting. In this particular application of image restoration, the signal is the outcome of a linear operator that deletes a number of pixels from the original image  $\mathbf{x}$ , plus the measurement noise. By considering a sparse prior on the original signal, we can formulate an equivalent problem to that of Eq. (6), where  $\mathbf{A}$  is the missing-pixels mask. The corresponding cost function can be minimized in a block coordinate manner, coding for the unknown sparse representation and updating the dictionary. In this case, however, the threshold in the OMP has to consider only the energy of existing pixel in each patch [9]. This again represents the first iteration of the Half Splitting strategy proposed in [16], and we may perform the next iterations by estimating the remaining noise as explained above. Furthermore, after the first iteration our estimate includes values of the missing pixel. We can then make use of the previous denoising strategy to tackle the next iteration, by having knowledge of the supports used to inpaint each patch, as it was previously explained.



**Fig. 5.** Atoms from a dictionary trained on a noisy version of the image Lena. The top row corresponds to the atoms after the first iteration of our method (essentially, after applying K-SVD), while the lower row corresponds to the same atoms after 4 iterations of the EPLL with a sparsity enforcing prior.

**Table 1.** Inpainting results in terms of Peak Signal to Noise Ratio (PSNR) for 25%, 50% and 75% missing pixels for the images *peppers* (left subcolumns) and *Lena* (right subcolumns), with additive white Gaussian noise ( $\sigma = 20$ ).

Missing Pixels	25%		50%		75%	
K-SVD	29.67	28.81	27.92	27.27	23.64	23.86
EPLL+K-SVD	<b>29.71</b>	<b>28.85</b>	<b>28.18</b>	<b>27.39</b>	<b>23.81</b>	<b>24.07</b>



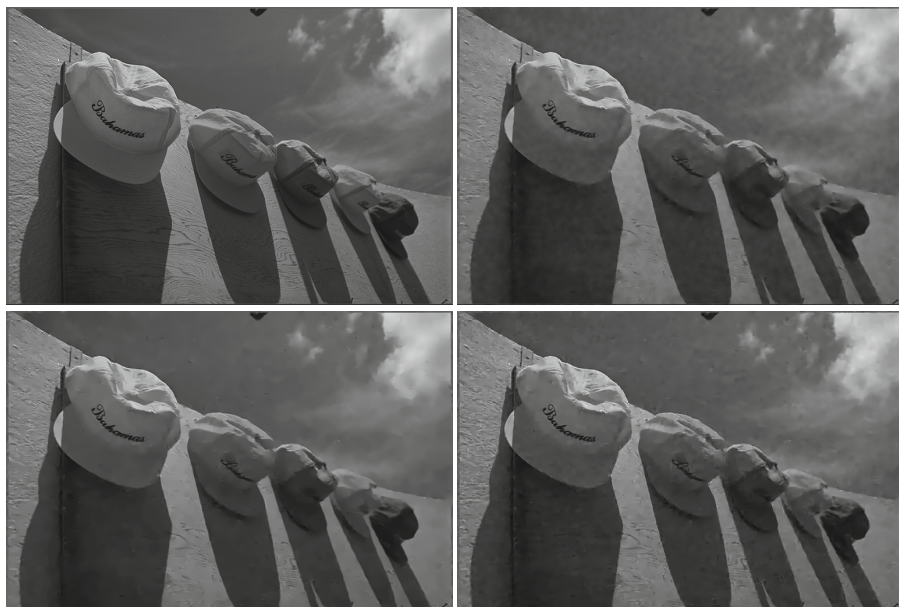
**Fig. 6.** Denoising results averaged over 12 images from the Kodak Dataset with respect to K-SVD [6] by EPLL with GMM [16] and the method presented here: EPLL with Sparse Prior, in terms of the Peak Signal to Noise Ratio (PSNR).

Table 1 shows the results on inpainting the popular images *peppers* and *Lena* with 25%, 50% and 75% missing pixels, with additive white Gaussian noise ( $\sigma = 20$ ). As it can be seen, the EPLL scheme leads to a slight improvement in the K-SVD inpainting results, with increased effect for higher missing pixels rates. The same concept could be applied to more sophisticated algorithms that use a sparsity-based prior, such as the state-of-the-art method of [11].

## 4.2 Denoising

We conclude this paper by presenting results on denoising of 12 images from the Kodak database, for different noise levels. We compare here the performance of the K-SVD denoising algorithm in [6] and our approach of the EPLL framework with a sparse prior (EPLL-K-SVD, where the dictionary is also updated in each iteration). In all cases we performed 4 iterations of this method, as this was found to be a convenient compromise between runtime and performance. For both K-SVD methods, an initial dictionary with 1024 atoms was trained on overlapping  $8 \times 8$  patches from 9 training images using K-SVD. We include for completion the results achieved by the EPLL with a Gaussian Mixture Model (GMM) as the image prior from [16].

In Fig. 6 we present the relative increase in PSNR, averaged over all 12 images. The EPLL with a Sparse enforcing Prior shows a clear improvement over the regular K-SVD. Furthermore, the complete implementation of the denoising algorithm closes the gap between the original K-SVD and EPLL-GMM, having comparable performance: our method achieves the best results for lower noise energy while EPLL with GMM is better for higher noise levels. In Fig.7 and Fig.8 we present two examples of denoised images by the three methods. Note how artifacts are notably reduced in the resulting images processed by our method.



**Fig. 7.** Denoising results of an image from the Kodak Database corrupted with a noise standard deviation of  $\sigma = 25$ . Top left: original image. Top right: K-SVD (PSNR = 32.14 dB). Bottom left: EPLL with Sparse Prior (PSNR = 32.42 dB). Bottom Right: EPLL with GMM (PSNR = 32.25 dB).



**Fig. 8.** Denoising results of an image from the Kodak Database, initially corrupted with additive white Gaussian noise ( $\sigma = 25$ ). From left to right: Original Image, K-SVD (PSNR = 31.42 dB), EPLL with Sparse Prior (PSNR = 31.83 dB), and EPLL with GMM (PSNR = 31.85 dB).

## 5 Conclusion

Maximizing the Expected Patch Log Likelihood with a sparse inducing prior leads naturally to a formulation of which the K-SVD algorithm represents the first iteration. In its original form, this method performed only one update of the image due to technical difficulties in assessing the remaining noise level. In this paper we have shown how to go beyond this first iteration, intrinsically determining the coding threshold in each step. This work completes the one in [6], providing the full path to the numerical minimization of the original cost function and exploiting all the potential of the sparse inducing prior.

Our algorithm shows a clear improvement over K-SVD in all the experiments. In denoising in particular, EPLL with a sparse prior achieved comparable performance to the state of the art method of EPLL with a GMM prior. Interestingly, both priors yield comparable results when applied within the EPLL framework. An approach like the one presented here could be employed in other applications where a MAP estimator for a sparse prior is used for image restoration.

**Acknowledgment.** This research was supported by the European Research Council under European Union's Seventh Framework Program, ERC Grant agreement no. 320649, by the Intel Collaborative Research Institute for Computational Intelligence and by the J.D. Erteschik Fund for Practical Research.

## References

1. Aharon, M., Elad, M., Bruckstein, A.M.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Process* 54(11), 4311–4322 (2006)
2. Bruckstein, A.M., Donoho, D.L., Elad, M.: From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review* 51(1), 34–81 (2009)



3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 60–65. IEEE (2005)
4. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3D filtering. In: Proc. SPIE-IS&T Electron. Imaging, vol. 6064, pp. 1–12 (2006)
5. Donoho, D., Elad, M., Temlyakov, V.: Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory* 52(1), 6–18 (2006)
6. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* 15(12), 3736–3745 (2006)
7. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online Dictionary Learning for Sparse Coding. In: 26th International Conference on Machine Learning, Montreal, Canada (2009)
8. Mairal, J., Bach, F., Sapiro, G.: Non-local Sparse Models for Image Restoration. In: 12th IEEE International Conference on Computer Vision, vol. 2, pp. 2272–2279 (2009)
9. Mairal, J., Elad, M., Sapiro, G., Member, S.: Sparse Representation for Color Image Restoration. *IEEE Transactions of Image Processing* 17(1), 53–69 (2008)
10. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing* 12(11), 1338–1351 (2003)
11. Romano, Y., Protter, M., Elad, M.: Single Image Interpolation via Adaptive Non-Local Sparsity-Based Modeling. *IEEE Transactions on Image Processing* 23(7), 3085–3098 (2014)
12. Roth, S., Black, M.J.: Fields of Experts. *International Journal of Computer Vision* 82(2), 205–229 (2009)
13. Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. Tech. - Comput. Sci. Dep. - Technical Report, pp. 1–15 (2008)
14. Tropp, J.: Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory* 50(10), 2231–2242 (2004)
15. Weiss, Y., Freeman, W.T.: What makes a good model of natural images? In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (June 2007)
16. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: 2011 International Conference on Computer Vision (ICCV), pp. 479–486 (November 2011)

# Blind Deconvolution via Lower-Bounded Logarithmic Image Priors

Daniele Perrone, Remo Diethelm, and Paolo Favaro

Department of Computer Science and Applied Mathematics  
University of Bern  
Neubrückestrasse 10  
3012 Bern, Switzerland

{perrone,favaro}@iam.unibe.ch, remo.diethelm@outlook.com

**Abstract.** In this work we devise two novel algorithms for blind deconvolution based on a family of logarithmic image priors. In contrast to recent approaches, we consider a minimalistic formulation of the blind deconvolution problem where there are only two energy terms: a least-squares term for the data fidelity and an image prior based on a lower-bounded logarithm of the norm of the image gradients. We show that this energy formulation is sufficient to achieve the state of the art in blind deconvolution with a good margin over previous methods. Much of the performance is due to the chosen prior. On the one hand, this prior is very effective in favoring sparsity of the image gradients. On the other hand, this prior is non convex. Therefore, solutions that can deal effectively with local minima of the energy become necessary. We devise two iterative minimization algorithms that at each iteration solve convex problems: one obtained via the primal-dual approach and one via majorization-minimization. While the former is computationally efficient, the latter achieves state-of-the-art performance on a public dataset.

**Keywords:** blind deconvolution, majorization-minimization, primal-dual, image prior.

## 1 Introduction

In the past decade, several high-performing blind deconvolution schemes using Bayesian principles have been proposed [1, 5, 6, 8, 9, 11, 17, 21–23]. The first step in the Bayesian framework is to devise a statistical distribution for both the gradients of the sharp image and the measurement noise or the model error. This joint distribution is used to pose a *maximum a posteriori* (MAP) problem, which yields point estimates for both the sharp image and the blur kernel. Also, one can marginalize the joint distribution with respect to one of the unknown random variables (typically, the sharp image) and then solve maximum a posteriori of the marginalized distribution. However, marginalization is typically computationally intractable. Thus, a variational Bayes upper bound is used together with several approximations such as independence of the random variables and explicit

simplified models of their distributions [21]. Whether one chooses one approach or another, the final algorithm is always an alternating minimization scheme. One iteratively improves the estimates of sharp image, blur kernel and some additional auxiliary variables [9]. The main differences among these schemes lie in how the coefficients weighing each term in the energy being minimized are updated at each iteration. Currently, the general wisdom is probably that the variational Bayes approach yields a better performance than the more classical MAP approach on the joint distribution. This belief is also reinforced by arguments showing how classical MAP approaches, such as total variation blind deconvolution [4], have fundamental shortcomings that would prevent them from achieving the desired solution [10, 21]. In contrast to those findings, in this paper we introduce two novel alternating minimization algorithms that can be cast as classical MAP approaches and that yield state of the art performance. Albeit only experimentally, one can then conclude that there is no inherent advantage in using a MAP or a variational Bayes approach. Moreover, other critical limitations of MAP [10] are overcome by using an alternating minimization with a delayed scaling [14].

As in the vast majority of blind deconvolution algorithms, our approach uses an image prior that strongly encourages sparsity in the image gradients of the reconstructed sharp image. We propose to use the logarithm of the norm of the gradients. While this prior was already introduced in [1] in a variational Bayes framework, here we use it in a MAP approach. Furthermore, to avoid the trivial solution (a constant), we introduce a lower-bound in the norm. This bound is essential to the correct functioning of the prior and needs to be carefully balanced with the data-fidelity term to yield a sparse gradient solution. The other challenges that we address are the non convexity of the image prior even when blur is given and the limited computational efficiency of the alternating minimization scheme. We do so by using two techniques: *majorization-minimization* [7] and the *primal-dual* method [3]. In the first case a tight upper bound of the image prior is obtained and iteratively updated. This algorithm achieves high accuracy. In the second case the Legendre-Fenchel transform and the proximal operator are used to produce iterations that mostly work on independent 1D updates, and that can therefore be executed in parallel. This algorithm is instead highly computationally-efficient.

## 2 Blind Deconvolution

Consider the following model for a blurry image  $f$

$$f = k * u + n \tag{1}$$

where  $k$  is the camera blur (or point spread function),  $u$  is the sharp image and  $n$  is the sensor noise. In this model, blur does not change across the image. This assumption does not hold in real scenes with depth variation and/or with general camera motions. Given both the blurry image  $f$  and the blur  $k$ , the estimation of

the sharp image  $u$  is a *deblurring* problem. When instead only the blurry image  $f$  is given, the problem of estimating both the sharp image  $u$  and the blur  $k$  is called *blind deconvolution*.

A widely used framework for solving deblurring or denoising (when  $k$  is the Dirac delta) is to look for a solution to the following minimization problem

$$u = \arg \min_u \|u\|_{TV} + \lambda \|k * u - f\|_2^2 \quad (2)$$

where the first term corresponds to the ubiquitous Total Variation (TV) of  $u$  [16],  $\lambda > 0$  is a regularization constant and the second term corresponds to the data fitting error. This problem is convex and therefore the global minimum can be achieved very efficiently. Often, however, one does not know the blur  $k$  and is therefore faced with the more challenging blind deconvolution problem, which is non convex. Currently, several approaches can successfully obtain high-quality results [1, 5, 6, 8, 9, 11, 17, 21–23]. A formulation of blind deconvolution inspired by eq. (2) is

$$\begin{aligned} u, k = \arg \min_{u, k} \|u\|_{TV} + \lambda \|k * u - f\|_2^2 \\ \text{s.t.} \quad k \succeq 0, \quad \|k\|_1 = 1, \end{aligned} \quad (3)$$

which has already been proposed in the past [4, 24]. This formulation, however, suffers from several limitations. Firstly, the global minima of this problem are the no-blur solutions, where  $u = f$  and  $k = \delta$ , up to translation [10, 14, 21]. Secondly, this is a non convex problem in both  $u$  and  $k$ . Thus, the solution obtained via an iterative method depends on the initialization of the unknowns. Despite the limitations outlined above, early alternating minimization implementations [4] for eq. (3) converged to desirable solutions. While many of current algorithms are derived via variational Bayes arguments or based on edge enhancements and noise suppression, they eventually result in alternating minimization schemes each resembling that used to solve eq. (3) [9]. The key differences are the introduction of additional auxiliary variables and the dynamic update of regularization parameters. As demonstrated experimentally, any such variation leads to a quite different performance. Moreover, most recent approaches choose to solve this problem by working on gradients of the images, or, more in general, filtered images rather than directly on the intensities of the images. Then, once the blur  $k$  has been estimated, a final deblurring step is performed to obtain the sharp image  $u$ .

In this paper we show that two minimalistic optimization schemes working directly on the image intensities are sufficient to achieve state of the art performance. This allows us to conclude that any additional modification to the alternating minimization scheme, including working on filtered images, is not essential to converge to the desired solution. Before doing so, however, we briefly review relevant work and clarify differences when compared to our solutions.

### 3 Prior Work

As mentioned in the Introduction, most approaches in blind deconvolution can be described as *maximum a posteriori* (MAP) approaches. The MAP approach relies on an explicit definition of the joint probability

$$p(u, k|f) \propto p(f|u, k)p(u)p(k), \quad (4)$$

where  $p(f|u, k)$  is a generative model of the noise,  $p(u)$  is a prior of the sharp image and  $p(k)$  is a prior of the blur. Commonly used sharp image priors approximate the heavy-tail distribution of natural image gradients [18] via sparsity-inducing norm of the gradients of  $u$ . The  $\ell_2$  norm of the gradients (isotropic total variation), or the  $\ell_1$  norm of the derivatives (anisotropic total variation) are classical choices [4]. In contrast to other sparsity-inducing norms, total variation (TV)[16] has the desirable property of being convex. However, it also introduces a loss of contrast in the recovered sharp image [14, 19]. Other methods use heuristics to encourage sharp gradients [5, 17, 22], or some reweighing strategy of the norm of the gradients [8, 9]. The latter methods aim at approximating the  $l_0$  “norm” of the gradients, as proposed also by Xu et al. [23]. In this paper we also encourage sparsity in the gradients. However, we use the logarithm of TV, which yields a simple energy term while providing a good approximation to the number of nonzero gradients. Indeed, this prior has already demonstrated promising results in blind deconvolution [1, 21] and denoising [13].

Despite the widespread use of the above MAP formulation, finding the mode of the posterior probability of  $u$  and  $k$  has received many criticisms. Levin et al. [10] and Perrone and Favaro [14] have shown that a large class of commonly used image priors favors the blurry image instead of the sharp one. Because of such limitation, Levin et al. [11] suggested to marginalize over all possible sharp images  $u$  and thus to solve the reduced problem

$$\max_k p(k|f) = \min_k -\log p(k|f) = \min_k -\log \int_u p(u, f|k)p(k)du. \quad (5)$$

In general, the integral in problem (5) is intractable. Therefore, typically one looks for an approximate solution. A common approach is to minimize an upper bound of  $-\log p(k|f)$  using a variational Bayes strategy [1, 6, 11, 21]. So far, this class of methods has shown better performance compared to methods that directly solve the MAP problem (4).

Despite their apparent superior performance, Wipf and Zhang [21] have shown that methods that solve (5) using a variational Bayes strategy are equivalent to a MAP strategy as in (4). They experimentally show that with an  $\ell_p$  norm with  $p \ll 1$ , MAP approaches are able to favor the right sharp solution. They also argue that a variational Bayes approach should be preferred because it is more robust when minimizing a highly non-convex function. The conclusions given by Wipf and Zhang [21] suggest that minimizing a cost functional as in (4) is not limited per se, as long as one finds a minimization strategy that carefully avoids its local minima. In this paper, we propose two non-variational Bayes strategies

to minimize a functional based on a logarithmic non-convex prior and show that they can achieve state-of-the-art results.

## 4 A Logarithmic Image Prior

In this section we introduce our image prior. From a Bayesian perspective, natural images can be described as having a sparse collection of gradients [18]. Hence, one could employ sparsity-inducing priors of the image gradients. However, another point of view is that blurring results in the average of shifted and scaled replicas of the same image gradients. The likelihood that such replicas combine to cancel each other is statistically insignificant. Vice versa, this averaging is more likely to multiply the number of gradients by the number of nonzero elements in the blur. Thus, a different perspective is that, in the context of deblurring, the role of an image prior is to favor solutions that have as few gradients as possible regardless of their magnitude. Both points of view lead to the same principle, *i.e.*, one should choose as prior

$$\text{Number of non zero elements of } (|\nabla u|) \doteq \|\nabla u\|_0 \quad (6)$$

where  $\|\cdot\|_0$  denotes the  $l_0$  “norm” (the Hamming distance to zero) and  $\nabla u$  is the 2-D gradient of  $u$ . Unfortunately, optimization with this prior is very challenging and, typically, smoother alternatives such as  $\ell_p$  norms  $\|\nabla u\|_p^p$ , with  $0 < p < 1$ , are used. In this work we also consider a prior with a similar behavior and simple form.

Let us consider the discrete setting. In the 2D discrete case, we have images with  $N \times M$  pixels. The  $(i, j)$ -th entry of the blurry image  $u$  will be denoted by  $u_{i,j}$ . The first order (discrete) derivatives of  $u$  will be denoted by  $\nabla u \doteq [u_{i+1,j} - u_{i,j} \quad u_{i,j+1} - u_{i,j}]^T$ . As image prior we propose using the following logarithmic prior<sup>1</sup>

$$\log \|\nabla u\|_{2,\epsilon}^p \doteq \sum_{i=1}^N \sum_{j=1}^M \log \|\nabla u_{i,j}\|_{2,\epsilon}^p = \frac{p}{2} \sum_{i=1,j=1}^{N,M} \log \|\nabla u_{i,j}\|_{2,\epsilon}^2 \quad (7)$$

with  $p > 0$  and where

$$\|\nabla u_{i,j}\|_{2,\epsilon}^2 \doteq (u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2 + \epsilon^2 \quad (8)$$

for  $\epsilon > 0$  so that the argument of the logarithm is never 0. The parameter  $\epsilon$  leads to a lower bound for this prior equal to  $MNp \log \epsilon$ . We can then formulate our blind deconvolution problem as

$$\begin{aligned} u, k &= \arg \min_{u,k} \lambda \|k * u - f\|_2^2 + \log \|\nabla u\|_{2,\epsilon}^p \\ \text{s.t.} \quad & k \succcurlyeq 0, \quad \|k\|_1 = 1. \end{aligned} \quad (9)$$

<sup>1</sup> Although we choose an  $\ell_2$  norm, any  $\ell_q$  norm could be used. However, we have found experimentally that for a wide set of values in  $q$  this makes little difference in the final performance.

Notice how the role of  $\epsilon$  is fundamental. If  $\epsilon = 0$  then the optimal solution will always be  $u = 0$  for any  $\lambda$ . To understand how  $\epsilon$ ,  $\lambda$  and  $p$  relate, consider the following limit

$$\lim_{\epsilon \rightarrow 0} \frac{p}{2} + \frac{1}{\log(1/\epsilon^2)} \log \|\nabla u\|_{2,\epsilon}^p = \frac{p}{2} \|\nabla u\|_0 \quad (10)$$

which shows how the log prior approximates the desired  $l_0$  “norm”. Now, assume that  $0 < \epsilon \leq 1$  and we substitute  $\lambda$  in problem (9) with  $-\lambda p \log \epsilon$ . Then, in the limit for  $\epsilon \rightarrow 0$  we are solving

$$\begin{aligned} u, k &= \arg \min_{u,k} \lambda \|k * u - f\|_2^2 + \|\nabla u\|_0 \\ \text{s.t.} \quad &k \succcurlyeq 0, \quad \|k\|_1 = 1. \end{aligned} \quad (11)$$

Finally, to avoid the degenerate constant solution we can compare two cases: one when  $u = \text{constant}$  and one when  $u = f$  and  $k = \delta$ . The idea is to make sure that the cost function favors the no-blur solution over the constant one. We can therefore plug the two cases in the cost of problem (9) and obtain the following inequality

$$\log \|\nabla f\|_{2,\epsilon}^p < \lambda \|\bar{f} - f\|_2^2 + \frac{p}{2} MN \log \epsilon^2 \quad (12)$$

or, alternatively,

$$\log \left\| \frac{1}{\epsilon} \nabla f \right\|_{2,1}^p < \lambda \|\bar{f} - f\|_2^2. \quad (13)$$

Then, we use Jensen’s inequality and the fact that the logarithm is a concave function to obtain an upper bound of the left hand side of eq. (13)

$$\frac{p}{2} \sum_{i=1,j=1}^{N,M} \log \left[ \left\| \frac{1}{\epsilon} \nabla f_{i,j} \right\|_{2,1}^2 \right] \leq \frac{pMN}{2} \log \left[ \frac{1}{MN} \sum_{i=1,j=1}^{N,M} \left\| \frac{1}{\epsilon} \nabla f_{i,j} \right\|_{2,1}^2 \right]. \quad (14)$$

Then, if we choose  $\epsilon$  such that

$$\epsilon > \sqrt{\frac{\frac{1}{MN} \sum_{i=1,j=1}^{N,M} \|\nabla f_{i,j}\|_2^2}{e^{\frac{2\lambda}{pMN} \|f - \bar{f}\|_2^2} - 1}}} \quad (15)$$

where  $\bar{f}$  is the average value of  $f$ , the degenerate constant solution will be avoided. Also, notice that  $\frac{2\lambda}{pMN} \|f - \bar{f}\|_2^2 > 0$  and  $\frac{1}{MN} \sum_{i=1,j=1}^{N,M} \|\nabla f_{i,j}\|_2^2 > 0$  unless  $f$  is constant (in this case  $u$  constant is a plausible solution and it should not be avoided). This means that an  $\epsilon$  that satisfies eq. (15) always exists.

Finally, to solve problem (9) we use the alternating minimization scheme

<pre> initialize     <math>k^1 = k_1</math> iterate <math>t \in [1, \dots, T]</math>     <math>u^{t+1} = \arg \min_u \lambda \ k^t * u - g\ _2^2 + \log \ \nabla u\ _{2,\epsilon}^p</math>     <math>k^{t+1} = \arg \min_k \ k * u^{t+1} - f\ _2^2</math>     s.t. <math>k \succeq 0, \quad \ k\ _1 = 1.</math>                 </pre>	(16)
--	------

While the iteration in the blur  $k$  entails solving a convex problem, and we solve it as in [4], the minimization in the update of the sharp image  $u$  is non convex and requires more attention. To this purpose we introduce two solvers: one based on a primal-dual approach and another on majorization-minimization.

### 4.1 A Primal-Dual Solver

Recall the deblurring problem (given the blur  $k^t$ ) in Algorithm (16); here we rewrite it as

$$u = \arg \min_u \sum_{i=1, j=1}^{N, M} ((k^t * u)_{i,j} - f_{i,j})^2 + \frac{1}{\mu} \log \|\nabla u_{i,j}\|_{2,\epsilon}^2, \tag{17}$$

where  $\mu = 2\lambda/p$ . By using the primal-dual approach of Chambolle and Pock [3] we obtain the following minimax problem

$$u = \arg \min_u \max_{z_1, z_2} \langle k^t * u, z_1 \rangle - F_1^*(z_1) + \langle \nabla u, z_2 \rangle - F_2^*(z_2) \tag{18}$$

where  $F_1^*$  and  $F_2^*$  are conjugate functions of  $F_1$  and  $F_2$  respectively, and we have defined

$$F_1(x) \doteq \sum_{i=1, j=1}^{N, M} (x_{i,j} - f_{i,j})^2, \quad F_2(\xi) \doteq \frac{1}{\mu} \sum_{i=1, j=1}^{N, M} \log \|\xi_{i,j}\|_{2,\epsilon}^2. \tag{19}$$

The conjugate functions can be computed via the Legendre-Fenchel (LF) transform [15] and are convex by construction. Thus problem (18) is a convex approximation in all variables  $z_1, z_2$  and  $u$  of the original problem (17). Notice that the convex approximation provided by the primal-dual formulation may not lead to the minima of the original non convex cost.

Our general primal-dual algorithm to solve problem (18) is

<pre> <math>z_1^{n+1} = \text{prox}_{\sigma F_1^*}(z_1^n + \sigma k^t * \bar{u}^n)</math> <math>z_2^{n+1} = \text{prox}_{\sigma F_2^*}(z_2^n + \sigma \nabla \bar{u}^n)</math> <math>u^{n+1} = u^n - \tau (k_-^t * z_1^{n+1} + \nabla \cdot z_2^{n+1})</math> <math>\bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n)</math>                 </pre>	(20)
---	------



where  $k_{-}^t$  denotes the mirrored blur kernel  $k^t$  (along both axes),  $n$  is the iteration index,  $\theta \in (0, 1]$  and  $\tau\sigma\|K\|^2 < 1$ , with  $\tau, \sigma > 0$ , where  $K$  is the matrix operator implementing both the blur  $k$  and the finite difference operator  $\nabla$ . Two of the 4 iterations in the above algorithm are defined based on the proximity operator. The proximity operator  $\text{prox}_{\sigma F_1^*}$  is computed via

$$\begin{aligned}\text{prox}_{\sigma F_1^*}(z) &= z - \sigma \text{prox}_{F_1/\sigma}(z/\sigma) \\ &= z - \sigma \arg \min_x \frac{1}{2} \left\| \frac{z}{\sigma} - x \right\|_2^2 + \sigma F_1(x) \\ &= \frac{1}{\sigma + 1} (z - \sigma f).\end{aligned}\quad (21)$$

The proximity operator  $\text{prox}_{\sigma F_2^*}$  is instead computed via

$$\text{prox}_{\sigma F_2^*}(z) = z - \sigma \arg \min_x \frac{1}{2} \left\| \frac{z}{\sigma} - x \right\|_2^2 + \sigma F_2(x). \quad (22)$$

Since the minimization problem is separable, let us consider the solution obtained for only one element  $x_{i,j}$  and  $z_{i,j}$  of the variables  $x$  and  $z$  respectively. With an abuse of notation, instead of the element-wise cumbersome notation  $x_{i,j}$  and  $z_{i,j}$  we simply refer to  $x$  and  $z$  in the next equations. We use the representation  $x \doteq \rho w$ , where  $\rho \geq 0$  and  $\|w\|_2 = \|z\|_2/\sigma$ . Then, let  $\xi = z/\sigma$  and we have

$$\arg \min_x \frac{1}{2} \|\xi - x\|_2^2 + \sigma F_2(x) = \arg \min_{\rho, w} \frac{\rho^2}{2} \left\| \frac{\xi}{\rho} - w \right\|_2^2 + \frac{\sigma}{\mu} \log \left( \rho^2 \frac{\|z\|_2^2}{\sigma^2} + \epsilon^2 \right). \quad (23)$$

Notice that the logarithmic term now depends only on  $\rho$ . Hence, we can first solve the minimization problem with respect to  $w$ . By simplifying the least squares term we obtain

$$\begin{aligned}\arg \min_{w, \|w\|=\|z\|/\sigma} \frac{\rho^2}{2} \left\| \frac{\xi}{\rho} - w \right\|_2^2 &= \arg \min_{w, \|w\|=\|z\|/\sigma} \|\xi\|_2^2/\rho^2 + \|w\|_2^2 - 2\langle \xi/\rho, w \rangle \\ &= \arg \min_{w, \|w\|=\|z\|/\sigma} \|\xi\|_2^2/\rho^2 + \|z\|_2^2/\sigma^2 - 2\langle \xi/\rho, w \rangle \\ &= \arg \max_{w, \|w\|=\|z\|/\sigma} \langle \xi, w \rangle\end{aligned}\quad (24)$$

which immediately yields  $w = \frac{\|z\|_2}{\sigma\|\xi\|_2}\xi = z/\sigma$ . By substituting the expression of  $w$  back into eq. (23) we finally have

$$\begin{aligned}\arg \min_x \frac{1}{2} \|\xi - x\|_2^2 + \sigma F_2(x) &= \xi \cdot \arg \min_{\rho} \frac{1}{2} \|\xi - \rho z/\sigma\|_2^2 + \frac{\sigma}{\mu} \log \left( \rho^2 \frac{\|z\|_2^2}{\sigma^2} + \epsilon^2 \right) \\ &= \xi \cdot \arg \min_{\rho} \frac{\mu}{2\sigma} (1 - \rho)^2 \|\xi\|_2^2 + \log \left( \rho^2 \|\xi\|_2^2 + \epsilon^2 \right).\end{aligned}\quad (25)$$

**Table 1.** The primal-dual algorithm

<b>initialize</b>
$h^1 = h_1$
<b>iterate</b> $t \in [1, \dots, T]$
<b>iterate</b> $n \in [1, \dots, N_0]$
$z_1^{n+1} = \frac{1}{\sigma + 1} (z_1^n + \sigma(k^t * \bar{u}^n - f))$
$z_2^{n+1} = \left(1 - \mathcal{H}\left(\frac{z_2^n + \sigma \nabla \bar{u}^n}{\sigma}, \mu, \epsilon, \sigma\right)\right) (z_2^n + \sigma \nabla \bar{u}^n)$
$\tilde{u}^{n+1} = \tilde{u}^n - \tau (k_-^t * z_1^{n+1} + \nabla \cdot z_2^{n+1})$
$\bar{u}^{n+1} = \tilde{u}^{n+1} + \theta(\tilde{u}^{n+1} - \tilde{u}^n)$
<b>end iterate</b> n
$u^{t+1} = \tilde{u}^{N_0+1}$
$h^{t+1} = \arg \min_k \ k * u^{t+1} - f\ _2^2$
s.t. $k \succcurlyeq 0, \quad \ k\ _1 = 1$
<b>end iterate</b> t

We can define  $\mathcal{H}$  as the solution of the 1D problem

$$\mathcal{H}(\xi, \mu, \epsilon, \sigma) = \arg \min_{\rho} \frac{\mu}{2\sigma} (\rho - 1)^2 \|\xi\|_2^2 + \log(\rho^2 \|\xi\|_2^2 + \epsilon^2) \quad (26)$$

and build it into a lookup table.<sup>2</sup> The proximity operator  $\text{prox}_{\sigma F_2^*}$  can then be written as

$$\text{prox}_{\sigma F_2^*}(z) = \left(1 - \mathcal{H}\left(\frac{z}{\sigma}, \mu, \epsilon, \sigma\right)\right) z. \quad (27)$$

The final algorithm is summarized in Table 1. Notice how several operations are parallelizable, thus leading to a very efficient implementation.

## 4.2 A Majorization-Minimization Approach

As a more accurate alternative to the primal-dual algorithm, one could use a majorization-minimization (MM) approach [7], in a similar manner as proposed by Candes *et al.* [2]. In the MM approach one defines an upper bound functional  $\psi(u|u^t)$  given the current estimate  $u^t$  at time  $t$ . This upper bound must satisfy the following properties

$$\psi(u|u^t) \geq \sum_{i=1}^N \sum_{j=1}^M \log \|\nabla u_{i,j}\|_{2,\epsilon}^p \quad \psi(u^t|u^t) = \sum_{i=1}^N \sum_{j=1}^M \log \|\nabla u_{i,j}^t\|_{2,\epsilon}^p. \quad (28)$$

<sup>2</sup> Notice that the 1D problem leads to a third order polynomial equation for which closed-form solutions are known.

**Table 2.** The majorization-minimization algorithm

<b>initialize</b> $h^1 = h_1$ <b>iterate</b> $t \in [1, \dots, T]$ $u^{t+1} = \arg \min_u \sum_{i=1, j=1}^{N, M} \lambda ((k^t * u)_{i,j} - f_{i,j})^2 + \frac{\ \nabla u_{i,j}\ _2^p}{\ \nabla u_{i,j}^t\ _2^p}$ $h^{t+1} = \arg \min_k \ k * u^{t+1} - f\ _2^2$ <p style="text-align: center;">s.t. <math>k \succcurlyeq 0, \quad \ k\ _1 = 1</math></p> <b>end iterate</b> $t$
--

Then, one can apply the following iterative scheme and provably reach a local minimum of the original function

$$u^{t+1} = \arg \min_u \sum_{i=1, j=1}^{N, M} \lambda ((k * u)_{i,j} - f_{i,j})^2 + \psi(u|u^t). \quad (29)$$

As upper bound we consider using the Taylor expansion of the logarithm around the  $t$ -th estimate of  $\|\nabla u\|_{2,\epsilon}^p$  up to the first term

$$\psi(u|u^t) = \sum_{i=1, j=1}^{N, M} \log \|\nabla u_{i,j}^t\|_{2,\epsilon}^p + \frac{\|\nabla u_{i,j}\|_{2,\epsilon}^p - \|\nabla u_{i,j}^t\|_{2,\epsilon}^p}{\|\nabla u_{i,j}^t\|_{2,\epsilon}^p}. \quad (30)$$

The properties (28) hold because of the concavity of the logarithm function. Finally, by plugging  $\psi$  in eq. (29) we obtain the following update

$$\begin{aligned} u^{t+1} &= \arg \min_u \sum_{i=1, j=1}^{N, M} \lambda ((k * u)_{i,j} - f_{i,j})^2 \\ &\quad + \log \|\nabla u_{i,j}^t\|_{2,\epsilon}^p + \frac{\|\nabla u_{i,j}\|_{2,\epsilon}^p - \|\nabla u_{i,j}^t\|_{2,\epsilon}^p}{\|\nabla u_{i,j}^t\|_{2,\epsilon}^p} \\ &= \arg \min_u \sum_{i=1, j=1}^{N, M} \lambda ((k * u)_{i,j} - f_{i,j})^2 + \frac{\|\nabla u_{i,j}\|_{2,\epsilon}^p}{\|\nabla u_{i,j}^t\|_{2,\epsilon}^p}. \end{aligned} \quad (31)$$

so that the majorization-minimization algorithm can be summarized in Table 2. Notice the similarity with reweighed least squares algorithms when  $p = 2$ .

## 5 Experiments

We evaluated the proposed algorithms on the dataset from Levin et al. [10]. The dataset is made of 4 images of size  $255 \times 255$  pixels blurred with 8 different

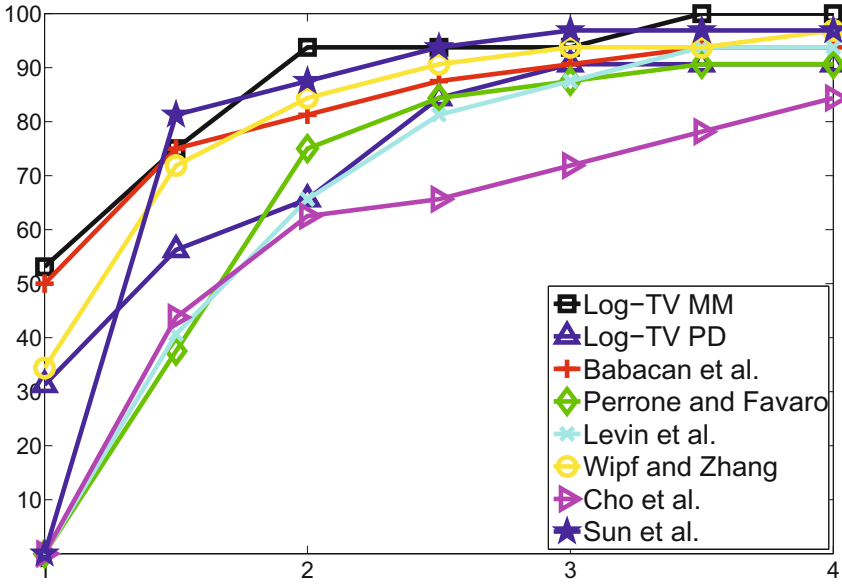


Fig. 1. Cumulative histogram of SSD ratio results on the dataset [10]

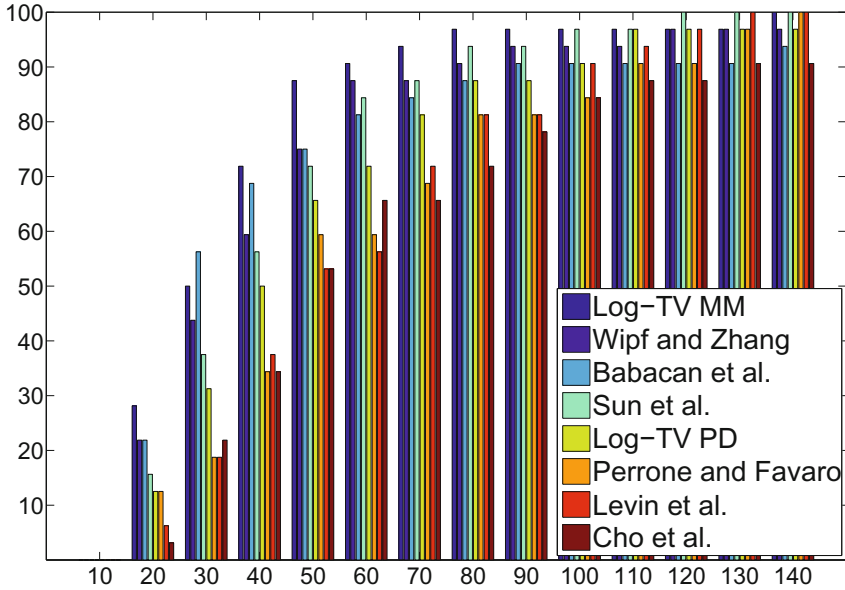


Fig. 2. Cumulative histogram of SSD results per image of the database [10]

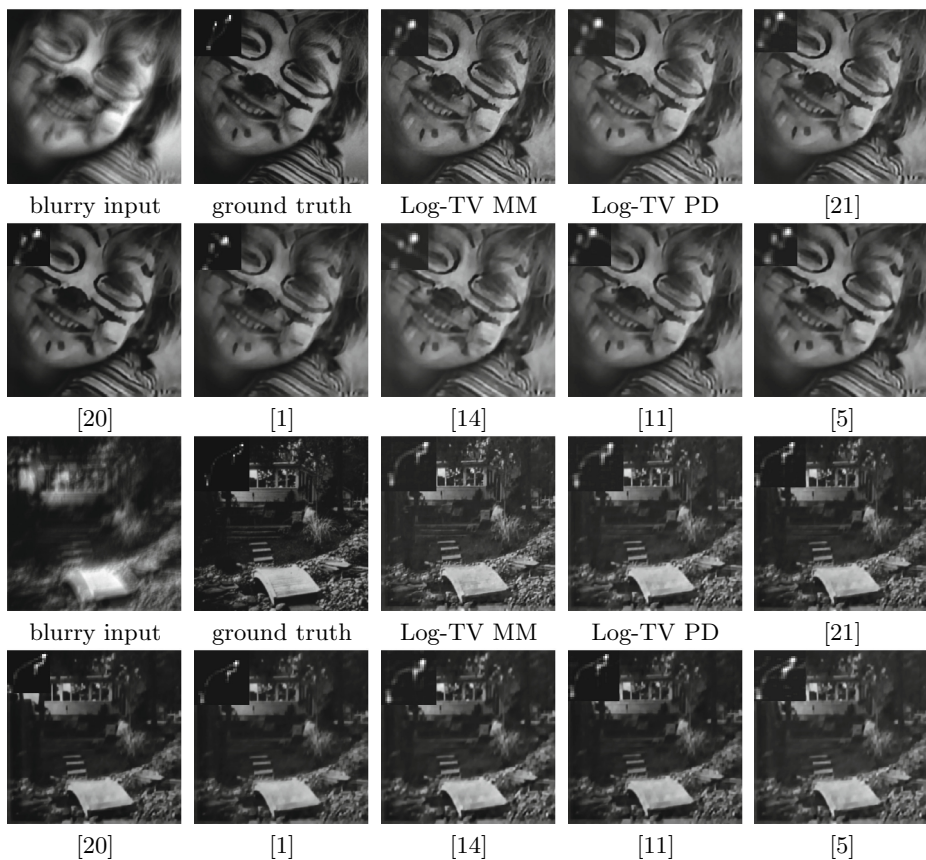


Fig. 3. Examples of deblurred images from Levin et al. [10] dataset

blurs, and it is provided with ground truth sharp images and blurs. Therefore it is possible to use metrics that take into account the intrinsic difficulty of each blur, such as the SSD ratio proposed in [10]. This ratio can be computed by

$$r = \frac{\sum_{i=1, j=1}^{N, M} (u_{i,j}^{k^e} - u_{i,j}^g)^2}{\sum_{i=1, j=1}^{N, M} (u_{i,j}^{k^g} - u_{i,j}^g)^2} \quad (32)$$

where  $u^g$  is the ground truth sharp image,  $u^{k^g}$  is the image obtained by solving a non-blind deconvolution problem with the ground truth blur, and  $u^{k^e}$  is the image obtained by solving a non-blind deconvolution problem with the estimated blur. For each method the same parameters are used for all the 32 blurry images of the dataset. For all the tests we used the non-blind deconvolution algorithm from Levin et. al. [12], where for each method we carefully selected the regularization parameter in order to have the best SSD ratio.

In Fig. 1 we show the cumulative histogram of the SSD ratios for several methods in the literatures and for our proposed algorithms (Log-TV MM and Log-TV PD). The MM algorithm achieves error ratio equal to 1 for more than 50% of the images, clearly outperforming the methods from Wipf and Zhang [21] and Babacan et. al. [1], and, for most error ratios, the method of Sun *et al.* [20]. Our primal-dual method is on par with high performing variational Bayesian algorithms such as the one from Levin et. al. [11]. In Fig. 2 we also show the cumulative histogram of the SSD errors, while in Fig. 3 we show some of the sharp images obtained on this dataset<sup>3</sup>.

For our methods we used the same regularization parameter  $\lambda = 30000$ ,  $\epsilon = 0.001$ ,  $p = 2$  and 3500 iterations for each pyramid level. For the primal-dual algorithm we set  $N_0 = 1$ ,  $\tau = 0.005$  and  $\sigma = \frac{1}{32\tau}$ . The parameter values have been found experimentally. We used a pyramid scheme where the input image and the blur are down sampled at each level by  $\sqrt{2}$ , and the parameter  $\lambda$  is divided by the number 2.1. The number of levels of the pyramid is computed such that at the top level the blur kernel has a support of 3 pixels. For the other methods we used the estimates provided by the authors, or we ran their algorithm using the tuning that gives the best results. The primal-dual method has the desirable feature of being parallelizable and therefore faster, but at the cost of being too coarse (due to the convex approximation of the logarithmic prior), thus unable to achieve the same accuracy of the MM algorithm.

## 6 Conclusions

In this paper we presented solutions to blind deconvolution based on a logarithmic image prior. The chosen prior is as effective as  $\ell_p$  norms with  $p < 1$  on the image gradients, while at the same time leading to simple optimization schemes despite its non convexity. To solve blind deconvolution with this image prior we propose a computationally efficient scheme via a primal-dual approach and a high-accuracy scheme via the majorization-minimization approach. Both approaches perform well and converge very robustly.

## References

1. Babacan, S.D., Molina, R., Do, M.N., Katsaggelos, A.K.: Bayesian blind deconvolution with general sparse image priors. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 341–355. Springer, Heidelberg (2012)
2. Candes, E.J., Wakin, M.B., Boyd, S.: Enhancing sparsity by reweighted  $\ell_1$  minimization. *Journal of Fourier Analysis and Applications* 14(5), 877–905 (2008)
3. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40(1), 120–145 (2011)

---

<sup>3</sup> A list of all the experiments is available at [www.cvg.unibe.ch/dperrone/logtv/](http://www.cvg.unibe.ch/dperrone/logtv/)

4. Chan, T., Wong, C.K.: Total variation blind deconvolution. *IEEE Transactions on Image Processing* 7(3), 370–375 (1998)
5. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graph.* 28(5), 145:1–145:8 (2009)
6. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graph.* 25(3), 787–794 (2006)
7. Hunter, D., Lange, K.: A tutorial on mm algorithms. *The American Statistician* 58, 30–37 (2004)
8. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 233–240 (June 2011)
9. Krishnan, D., Bruna, J., Fergus, R.: Blind deconvolution with re-weighted sparsity promotion. *CoRR abs/1311.4029* (2013)
10. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(12), 2354–2367 (2011)
11. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Efficient marginal likelihood optimization in blind deconvolution. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2657–2664 (June 2011)
12. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.* 26(3) (July 2007)
13. Ochs, P., Chen, Y., Brox, T., Pock, T.: ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM J. Imaging Sciences* 7(2), 1388–1419 (2014)
14. Perrone, D., Favaro, P.: Total variation blind deconvolution: The devil is in the details. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2909–2916 (June 2014)
15. Rockafellar, R.: *Convex Analysis*. Princeton University Press (1970)
16. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* 60(1-4), 259–268 (1992)
17. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph.* 27(3), 73:1–73:10 (2008)
18. Srivastava, A., Lee, A., Simoncelli, E.P., Zhu, S.C.: On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision* 18, 17–33 (2003)
19. Strong, D., Chan, T.: Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems* 19(6), S165 (2003)
20. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: 2013 IEEE International Conference on Computational Photography (ICCP), pp. 1–8 (April 2013)
21. Wipf, D., Zhang, H.: Analysis of bayesian blind deconvolution. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMMCVPR 2013*. LNCS, vol. 8081, pp. 40–53. Springer, Heidelberg (2013)
22. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: Dailidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part I*. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
23. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1107–1114 (June 2013)
24. You, Y.L., Kaveh, M.: A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing* 5(3), 416–428 (1996)

# Low Rank Priors for Color Image Regularization

Thomas Möllenhoff, Evgeny Strelakovsky, Michael Moeller, and Daniel Cremers

Department of Computer Science, Technical University of Munich, Germany

**Abstract.** In this work we consider the regularization of vectorial data such as color images. Based on the observation that edge alignment across image channels is a desirable prior for multichannel image restoration, we propose a novel scheme of minimizing the rank of the image Jacobian and extend this idea to second derivatives in the framework of total generalized variation. We compare the proposed convex and nonconvex relaxations of the rank function based on the Schatten- $q$  norm to previous color image regularizers and show in our numerical experiments that they have several desirable properties. In particular, the nonconvex relaxations lead to better preservation of discontinuities. The efficient minimization of energies involving nonconvex and nonsmooth regularizers is still an important open question. We extend a recently proposed primal-dual splitting approach for nonconvex optimization and show that it can be effectively used to minimize such energies. Furthermore, we propose a novel algorithm for efficiently evaluating the proximal mapping of the  $\ell^q$  norm appearing during optimization. We experimentally verify convergence of the proposed optimization method and show that it performs comparably to sequential convex programming.

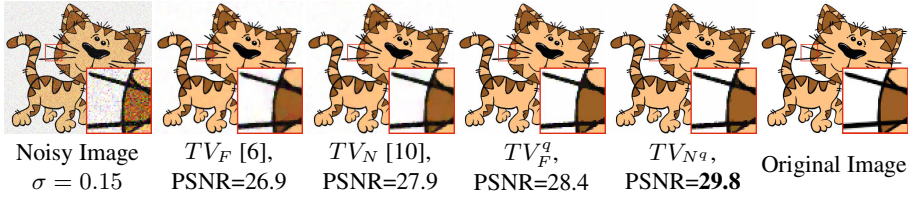
## 1 Introduction

Developing effective image regularization priors is of central importance for variational image reconstruction methods and inverse problems. The *total variation* ( $TV$ ) pioneered as a discontinuity-preserving regularizer [1], and still ranges among the most popular and versatile regularizers [2]. Since the classical total variation was proposed for grayscale images, a lot of recent research has focused on extending the TV to color images. Among these works are straightforward extensions of using TV regularization on each color channel separately [3], using a global coupling of the color channels by penalizing the  $\ell^2$  norm of the total variations of the channels [4], as well as using the Frobenius norm of the derivative matrix at each pixel [5,6]. Additionally, it has been proposed to incorporate a change of color space [7], as well as to couple the color channels with an  $\ell^\infty$  norm [8].

Based on the class of methods presented by Sapiro and Ringach [5], the authors of [9] proposed the penalization of the Schatten- $\infty$  norm of the derivative matrix at each pixel, i.e. the penalization of the largest singular value of the Jacobian. One approach we are particularly interested in was also motivated by [5]: The authors of [10] suggested to penalize the Schatten-1 norm, also known as the nuclear norm, of the Jacobian at each pixel, i.e. they suggested to penalize

$$TV_N(u) := \|\nabla u\|_{N,1} := \int_{\Omega} \left\| \begin{pmatrix} \partial_x u_1(x, y) & \partial_x u_2(x, y) & \partial_x u_3(x, y) \\ \partial_y u_1(x, y) & \partial_y u_2(x, y) & \partial_y u_3(x, y) \end{pmatrix} \right\|_N dx dy, \quad (1)$$





**Fig. 1.** We propose a novel regularizer based on the nonconvex relaxation of the rank norm ( $TV_{N^q}$ ). The above comparison shows that the nonconvex regularizers (for values of  $q < 1$ , here  $q = 0.5$ ) outperform the convex ones, as they are able to better preserve discontinuities. The proposed regularizer has significantly less color artefacts at discontinuities as it favors coherent jumps of the color channels.

for an image  $u : \Omega \rightarrow \mathbb{R}^3$ , where  $\|\cdot\|_N$  denotes the nuclear norm. Since the nuclear norm of the derivative is a convex relaxation for minimizing the rank of a matrix, we can interpret this approach as the rank minimization of the Jacobian. Note that the Jacobian being of rank one means that all gradient vectors are linearly dependent and thus point in the same (or opposite) direction. The latter is an interesting regularization property which has been exploited in other contexts such as nonlinear diffusion [11] or color Bregman iteration [12]. In this paper we propose a novel rank minimization of the derivative matrix through nonconvex relaxation by considering the penalization with the Schatten- $q$  norm for  $0 < q < 1$ .

Another motivation for such nonconvex relaxations comes from studies on the statistics of natural images. Filter responses are more faithfully represented by heavy-tailed distributions giving rise to nonconvex regularizers [13,14]. This led to the work of Krishnan et al. [14], who demonstrated that standard  $TV$  denoising and deblurring results can indeed be improved by replacing the usual  $\ell^1$  norm of the gradient with the nonconvex  $\ell^q$  norm for  $q < 1$ .

While penalizing the nuclear norm instead of the Frobenius norm of the Jacobian yields an improvement as shown by Lefkimmiatis et al. [10], and replacing the usual  $TV$ - $\ell^1$  norm by a  $TV$ - $\ell^q$  norm with  $q < 1$  yields another improvement [14], we will demonstrate that combining both ideas by replacing the nuclear norm with a Schatten- $q$  norm for  $q < 1$  leads to a regularization method superior to both previously mentioned approaches, as illustrated in Fig. 1.

One well known property of total variation regularization is the preference of piecewise constant images which can lead to so called staircasing effects. To avoid these artifacts, higher order methods such as the total generalized variation (TGV) have been proposed [15]. The  $TGV_2^\alpha$  model on a grayscale image  $u : \Omega \rightarrow \mathbb{R}$  can be interpreted as a particular type of infimal convolution written as

$$TGV(u) = \inf_{\nabla u = v + z} \alpha \|v\|_{2,1} + (1 - \alpha) \|\nabla z\|_{2,1} \tag{2}$$

where  $\|v\|_{2,1} = \int_{\Omega} \sqrt{v_1(x)^2 + v_2(x)^2} dx$  and  $\alpha \in [0, 1]$  is a weighting parameter between first and second order penalization. Extensions for the TGV model include replacing the  $\|\cdot\|_{2,1}$  norms by nonconvex  $\|\cdot\|_{2,q}^q$  penalty functions on grayscale images

[16], as well as extending the TGV model to color images by considering the Frobenius norms of the derivative matrices arising from having different color channels [17].

In this paper we propose a novel extension of the TGV approach to color images by considering the nuclear norm of the derivative matrices. We will demonstrate in the numerical results that our convex nuclear norm TGV approach outperforms the Frobenius norm TGV for color image denoising. Moreover, we show that, again, replacing the nuclear norm by a Schatten  $q$ -norm with  $q < 1$  can improve the denoising performance even further.

The minimization of the resulting nonsmooth and nonconvex energy is a challenging task. We will use a recent reformulation of the primal-dual hybrid gradient method [18,19,20], which makes it applicable to nonconvex energies [21]. Although a full convergence theory has not yet been established, we demonstrate that one obtains a very efficient numerical scheme for finding low energies, comparably to methods which rely on sequential convex programming such as [16].

The rest of this paper is organized as follows. In the next section we will further motivate the idea of penalizing the Schatten- $q$  norm of the derivative matrices in the TV as well as in the TGV case in greater detail. Section 3 discusses the numerical method for minimizing the proposed energies in detail. Particular emphasis is put on the efficient evaluation of the proximity operators of the  $\ell^q$  seminorms for  $q < 1$ . The numerical results in Section 4 demonstrate the superior behavior of derivative matrix rank minimization in the TV as well as in the TGV case and demonstrate the advantages of the nonconvex regularizations. Finally, we draw conclusions and point out directions of future research in Section 5.

## 2 TV and TGV Rank Minimization Approaches

In this section we will give more details on the idea and motivation for considering certain Schatten- $q$  norms for  $q < 1$ . We define the Schatten- $q$  “norm” as

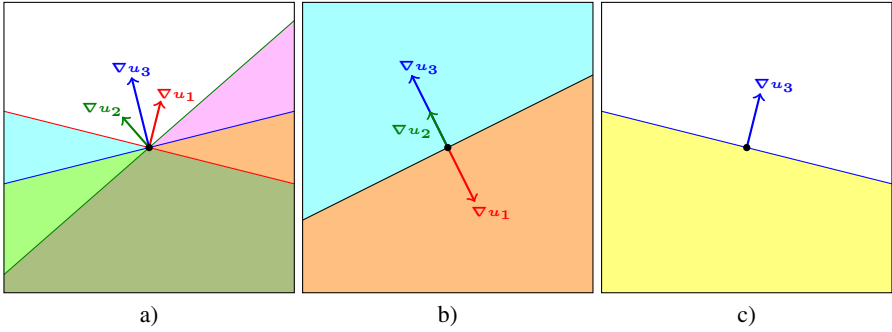
$$\|A\|_{N_q} := (\sigma_1^q + \dots + \sigma_n^q)^{1/q}, \quad (3)$$

where  $\sigma_i$  denotes the  $i$ -th singular value of  $A$ . Note that as a special case we obtain the rank function for  $q = 0$  (using the convention  $0^0 = 1$ ) and the nuclear norm for  $q = 1$ .

As pointed out in the introduction, the nuclear norm TV penalty (1) can be interpreted as a convex relaxation for encouraging a low rank of the Jacobian at each pixel. Our proposed Schatten- $q$  norm approximates the rank minimization, i.e. the penalty of the number of nonzero singular values, more closely.

$$TV_{N_q}(u) := \int_{\Omega} \left\| \begin{pmatrix} \partial_x u_1(x, y) & \partial_x u_2(x, y) & \partial_x u_3(x, y) \\ \partial_y u_1(x, y) & \partial_y u_2(x, y) & \partial_y u_3(x, y) \end{pmatrix} \right\|_{N_q}^q dx dy, \quad (4)$$

But why does it make sense to minimize the rank of this matrix? Note that the derivative matrix has at most rank two. A reduction of the rank could lead to a rank zero, which has the simple interpretation of all derivatives being zero, i.e. none of the channels changing. We therefore still expect the regularization to prefer piecewise constant



**Fig. 2.** Illustration of a point with gradient matrix of rank two **a)** and two different matrices with rank one in **b)** and **c)**. A Jacobian of rank zero would correspond to a locally constant region.

images. A derivative matrix with rank one on the other hand has the interpretation that all gradient vectors are linearly dependent and hence parallel (or antiparallel).

This is illustrated in Fig. 2, where on the left we show a rank two Jacobian and on the right two different rank one Jacobians. Note that the gradients always point in the normal direction to the level lines of each channel, such that the lines in Fig. 2 can be interpreted as particular level lines of the channels. The alignment of the normal lines in all channels seems to be a reasonable regularity assumption for natural images and leads to a reduction of color artifacts as we will see in the numerical results on color image denoising. As illustrated in the right image in Fig. 2, a derivative matrix with two derivative vectors being zero and one derivative vector being arbitrary also has rank one such that color edges are not necessarily forced to be aligned as in the middle image. We expect that the data term decides whether a full alignment as in the middle or a pointwise alignment as in the right image of Fig. 2 are to be preferred, such that we avoid overregularization or the introduction of artificial edges.

Furthermore, we propose to extend the idea of rank penalization of the derivatives to the TGV framework by minimizing

$$\begin{aligned}
 TGV_{N_q}(u) &:= \inf_{\nabla u=v+z} \alpha \int_{\Omega} \left\| \begin{pmatrix} v_{1,1}(x,y) & v_{1,2}(x,y) & v_{1,3}(x,y) \\ v_{2,1}(x,y) & v_{2,2}(x,y) & v_{2,3}(x,y) \end{pmatrix} \right\|_{N_q}^q dx dy \\
 &+ (1-\alpha) \int_{\Omega} \left\| \begin{pmatrix} \partial_x z_{1,1}(x,y) & \partial_x z_{1,2}(x,y) & \partial_x z_{1,3}(x,y) \\ \partial_x z_{2,1}(x,y) & \partial_x z_{2,2}(x,y) & \partial_x z_{2,3}(x,y) \\ \partial_y z_{1,1}(x,y) & \partial_y z_{1,2}(x,y) & \partial_y z_{1,3}(x,y) \\ \partial_y z_{2,1}(x,y) & \partial_y z_{2,2}(x,y) & \partial_y z_{2,3}(x,y) \end{pmatrix} \right\|_{N_q}^q dx dy. \quad (5)
 \end{aligned}$$

The above penalization can be motivated as follows. The Jacobian  $\nabla u$  of an image is optimally divided into two parts. The first part corresponds to  $v$  where the Schatten- $q$  norm of  $v$  is penalized. Thus, the interpretation of  $v$  is similar to the plain TV case discussed above: This part of the gradient of  $u$  should point in the same direction for all color channels. The second part of the functional penalizes the derivatives of  $z$  in the Schatten- $q$  norm and might be more difficult to interpret at first sight. The variable  $z$  contains parts of the Jacobian of  $u$ . For interpretation purposes let us assume that

$z = \nabla u$ . Then each column of the matrix in the second term of (5) is exactly the Hessian of one of the color channels. In this sense, the second term tries to align parts of the Hessian matrices of the color channels and therefore is the natural extension of aligning the first derivatives. Having the interpretation of a Hessian in mind, one could also motivate our approach by considering an image  $u$  whose color channels are twice continuously differentiable. In this case a second order Taylor expansion could describe the local behavior of each channel. Particularly, for color channels with parallel Hessians the second order behavior or the curvature of all color channels is the same up to a scaling and thus extends the coupling of different color channels from the first derivatives in the TV case to the second derivatives in the TGV case.

### 3 Application to Inverse Problems in Image Processing

We now consider inverse problems involving the proposed regularizers, given an input image  $f : \Omega \rightarrow \mathbb{R}^k$  with  $k$  channels on a  $d$ -dimensional discretized domain  $\Omega$ . For regularization of piecewise constant images we have the following variational problem

$$\min_u \frac{\lambda}{2} \|u - f\|^2 + \mathcal{R}(u), \quad (6)$$

where  $\mathcal{R}(u)$  is either  $TV_F^q(u) = \|\nabla u\|_{2,q}^q = \int_{\Omega} \|\nabla u(x)\|_2^q dx$  or  $TV_{N^q}(u)$  as defined in (4). For inverse problems involving *piecewise affine* and natural images we propose

$$\min_{u,v} \frac{\lambda}{2} \|u - f\|^2 + \mathcal{R}(u, v), \quad (7)$$

where  $\mathcal{R}(u, v)$  is  $TGV_F^q(u, v) = \alpha \|\nabla u - v\|_{2,q}^q + (1 - \alpha) \|\nabla_2 v\|_{2,q}^q$  or  $TGV_{N^q}(u, v)$  as defined in (5). Since for  $q < 1$  the regularizers are nonconvex and nonsmooth, their efficient numerical optimization is a challenging problem. In the next section we propose a minimization algorithm for energies involving the proposed regularizers.

#### 3.1 Splitting Methods in the Nonconvex Setting

Let us first introduce the *proximal mapping* associated with a proper, lower-semicontinuous function  $f : X \rightarrow \mathbb{R} \cup \{\infty\}$ :

$$\text{prox}_{\tau, f}(y) := \arg \min_x f(x) + \frac{1}{2\tau} \|x - y\|^2. \quad (8)$$

Note that if  $f$  is *nonconvex*, this mapping is not necessarily single-valued.

It has recently been shown experimentally that primal-dual splitting methods for convex optimization are also often applicable in the nonconvex setting [21,22]. Here we show how to generalize the recent approach [21] to our setting. In general, we aim to minimize cost functions of the form

$$\min_u G(u) + F(g) \quad \text{subject to} \quad Ku = g, \quad (9)$$

where  $G$  is convex,  $F$  possibly nonconvex and  $K$  a linear operator. The algorithm studied in [21,23] is given as

$$\begin{aligned}
 g^{n+1} &\in \text{prox}_{\sigma^{-1}, F} (K\bar{u}^n + \sigma^{-1}q^n), \\
 q^{n+1} &= q^n + \sigma(K\bar{u}^n - g^{n+1}), \\
 u^{n+1} &= \text{prox}_{\tau, G} (u^n - \tau K^T q^n), \\
 \bar{u}^{n+1} &= u^{n+1} + \theta(u^{n+1} - u^n),
 \end{aligned} \tag{10}$$

and reduces to the primal-dual hybrid gradient method (cf. [20]) for convex  $F$ . Interestingly, this update scheme can also be interpreted as gradient descent in the primal variables  $u$  and  $g$  and gradient ascent in the dual variable  $q$  on the following Lagrangian saddle-point formulation of (9):

$$\max_q \min_{u, g} G(u) + F(g) + \langle q, Ku - g \rangle. \tag{PD}$$

Since for nonconvex  $F$  it is generally not possible to interchange min and max, this is not the same as the primal-dual saddle point problem involving the Fenchel dual  $F^*$  from [2]. As observed in [23], a necessary condition on the dual step size for the algorithm to converge for *semiconvexity*  $F$ , i.e. for  $F$  with the property that  $F(u) + \frac{\omega}{2}\|u\|^2$  is convex, seems to be  $\sigma \geq 2\omega$ .

**Adaptive Step-Size Scheme.** As the  $\ell^q$  seminorms are neither semiconvex nor differentiable for  $q < 1$  one possibility would be to approximate it by a regularized or smoothed variant. However, this turns out to be difficult for the nonconvex relaxations of the rank function.

Instead we opt to employ a variable step size scheme where the dual step size approaches infinity ( $\sigma \rightarrow \infty$ ) as suggested in [21]:

$$\theta_n = 1/\sqrt{1 + 2\gamma\tau_n}, \quad \sigma_{n+1} = \sigma_n/\theta_n, \quad \tau_{n+1} = \tau_n\theta_n, \tag{11}$$

with  $\tau_0\sigma_0\|K\|^2 < 1$ . Here  $\gamma$  is an additional parameter which is usually chosen according to the strong convexity constant of  $G$ , e.g.  $\gamma = \lambda$  for  $G(u) = \frac{\lambda}{2}\|u - f\|^2$ . In the case of  $TGV$  regularization, the function  $G$  is not strongly convex due to the additional primal variable. We still pick  $\gamma = \lambda$  as a heuristical choice, as it works well in practice.

A similar approach is suggested by Storath et al. [22] for minimizing the Potts model, based on the direct application of the Alternating Direction Method of Multipliers (ADMM) to the nonconvex  $\ell^0$  regularizer while having a similar step size scheme where the penalty parameter in the ADMM method approaches infinity. Here, the adaptive step size scheme for the above primal-dual algorithm comes with an immediate interpretation in the *convex* setting for strongly convex  $G$ .

As the adaptive step size scheme yields  $\sigma_n \rightarrow \infty$ , the following interpretation is interesting. By considering the optimality conditions of the iterates produced by Algorithm (10) we see that the inclusion  $q^n \in \partial F(g^n)$  holds in every iteration (cf. [23]).

For *differentiable*  $F$  the subdifferential is a singleton and we can thus eliminate the variable  $q$ , and retrieve a formulation in terms of primal variables:

$$\begin{aligned}
 g^{n+1} &\in \text{prox}_{\sigma_n^{-1}, F} (K\bar{u}^n + \sigma_n^{-1}\nabla F(g^n)), \\
 u^{n+1} &= \text{prox}_{\tau_n, G} (u^n - \tau_n K^T \nabla F(g^n)), \\
 \bar{u}^{n+1} &= u^{n+1} + \theta_n(u^{n+1} - u^n).
 \end{aligned} \tag{12}$$

As  $\sigma_n \rightarrow \infty$  the proximity operator in the update in  $g$  becomes the identity and it can be seen that we approach the forward-backward splitting algorithm in the limit, applied to the primal problem (9) with an additional inertial term on  $u$ . This relation is interesting as convergence of forward-backward methods in the nonconvex setting is theoretically proven [24].

In order to apply Algorithm (10) to the problems at hand we require the efficient evaluation of the proximal mappings coming from the nonconvex regularizers, which is a nonconvex optimization problem itself.

### 3.2 Evaluation of the $\|\cdot\|_{2,q}^q$ Proximal Mapping

First we note that due to the separability, the proximal mappings reduce to pointwise evaluations of  $\|\cdot\|_2^q$ . We will focus on the former case first. Given  $g_0 \in \mathbb{R}^{d \times k}$ , we will consider the efficient minimization of the following proximal mapping for  $0 \leq q < 1$ :

$$\text{prox}_{\tau, \|\cdot\|_2^q}(g_0) = \arg \min_{g \in \mathbb{R}^{d \times k}} \frac{\|g - g_0\|_2^2}{2\tau} + \|g\|_2^q. \quad (13)$$

**The case  $q = 0$ .** An important special case is  $q = 0$ , which corresponds to Potts regularization. In this case the minimization (13) can actually be solved explicitly via *hard thresholding*:

$$\text{prox}_{\tau, \|\cdot\|_2^0}(g_0) = \begin{cases} 0 & \text{if } \|g_0\|_2 \leq \sqrt{2\tau}, \\ g_0 & \text{otherwise.} \end{cases} \quad (14)$$

For the general case  $0 < q < 1$ , we first note that the evaluation of the proximal operator (13) can be reduced to a scalar problem.

**Proposition 1.** *Given  $g_0 \in \mathbb{R}^{d \times k}$ ,  $\tau > 0$ ,  $q \in (0, 1)$  and  $\lambda > 0$ , the solution of the proximal operator*

$$\text{prox}_{\tau, \|\cdot\|_2^q}(g_0) = \arg \min_{g \in \mathbb{R}^{d \times k}} \frac{\|g - g_0\|_2^2}{2\tau} + \|g\|_2^q$$

has the form  $\hat{g} = tg_0$  for some real  $t \geq 0$ .

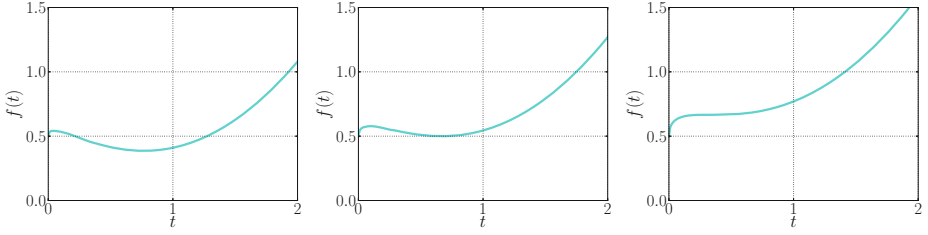
*Proof.* A proof is given in the appendix of [25].

**Solving the Scalar Problem.** Since we now know that the optimal solution is a scalar multiple of  $g_0$  we substitute  $g = tg_0$  in (13) and arrive at the following problem

$$\arg \min_{t \geq 0} \frac{(t-1)^2}{2} + \alpha t^q =: \arg \min_{t \geq 0} f(t) \quad (15)$$

for  $\alpha = \tau \|g_0\|_2^{q-2} \geq 0$ . Thus, evaluating the proximal operator (13) reduces to solving the above problem (15) for  $t \geq 0$ .

The minimization problem (15) can be solved in closed form for certain values of  $q$  such as  $1/2$  or  $3/4$  as described in [14]. In the following, we provide a more concise analytic solution for the special case  $1/2$  and an efficient algorithm based on Newton's method for the general case.



**Fig. 3.** The function (15) for three different values of  $\alpha$  and  $q = 0.5$ . **From left to right:**  $\alpha \approx 0.41 < \frac{2\sqrt{6}}{9}$ ,  $\alpha = \frac{2\sqrt{6}}{9} \approx 0.54$ ,  $\alpha = \frac{4}{3\sqrt{3}} \approx 0.77$ . It can be seen that the desired stationary point is also the global minimum for  $\alpha < \frac{2\sqrt{6}}{9}$  (left). In the equality case, the value at the stationary point is the same as the boundary value (center). For  $\alpha > \frac{4}{3\sqrt{3}}$  the function is increasing (right).

*Concise Closed Form Solution for  $q = 1/2$ .* Setting the derivative of the cost function (15) to 0 and substituting  $t = s^2$  we arrive at the cubic equation

$$s^3 - s + \frac{\alpha}{2} = 0. \quad (16)$$

Following the work of [26] we arrive at the following closed form expression for the root which corresponds to the minimum of (15):

$$\hat{s} = \frac{2}{\sqrt{3}} \sin\left(\frac{1}{3}\left(\arccos\left(\frac{3\sqrt{3}}{4}\alpha\right) + \frac{\pi}{2}\right)\right). \quad (17)$$

Interestingly this solution based on trigonometric expressions is quite a bit shorter than the one proposed in [14]. As shown in Fig. 3, we see that for some values of  $\alpha$  the value at the boundary is the optimal value. This is precisely for all  $\alpha$  satisfying the condition

$$\alpha > \frac{2\sqrt{6}}{9}. \quad (18)$$

If (18) is satisfied we simply set  $\hat{t} = 0$ , otherwise we find the root  $\hat{s}$  of (16) that corresponds to the local minimum using formula (17) and set  $\hat{t} = \hat{s}^2$ . As seen in Fig. 3, for  $\alpha > \frac{4}{3\sqrt{3}}$ , the function is increasing and does not have a stationary point. This corresponds to the case where the root in (17) is not real anymore.

*Newton's method for general  $0 < q < 1$ .* For general values of  $q$ , we solve the scalar  $\ell_q$  problem (15) using Newton's method. For all  $\alpha$  satisfying the condition

$$\alpha > \frac{1}{2-q} \left(2 \frac{1-q}{2-q}\right)^{1-q} \quad (19)$$

the boundary value is lower than the value at the local minimum as shown for  $q = 0.5$  in Fig. 3, so we set  $\hat{t} = 0$  if (19) is satisfied and otherwise we use Newton's method. For that we note that for  $\alpha = 0$  the optimal point is at  $\hat{t} = 1$ , and for  $\alpha > 0$  we have  $\hat{t} < 1$ . So we pick the starting point for Newton's method  $t_0 = 1$ . We perform the iteration

$$t_{k+1} = t_k - f'(t_k)/f''(t_k) \quad (20)$$

**Algorithm 1.** Newton's method for solving the nonconvex  $\ell^q$  proximal operator.**Input:** Parameters  $g_0 \in \mathbb{R}^{d \times k}$ ,  $\tau > 0$  and  $q \in (0, 1)$ , machine precision  $\varepsilon > 0$ **Output:** Minimizer  $\hat{g} \in \mathbb{R}^{d \times k}$  of (13)

```

if  $\|g_0\|_2 > 0$  then
   $\alpha = \tau \|g_0\|_2^{q-2}$ 
  if  $\alpha$  satisfies (19) then
     $\hat{t} = 0$ 
  else
    // Solve for optimum using Newton's method.
     $t_0 = 1$ 
    for  $k \geq 1$  until  $f'(t_k)/f''(t_k) < \varepsilon$  do
       $t_k = t_{k-1} - f'(t_k)/f''(t_k)$ 
     $\hat{t} = t_k$ 
   $\hat{g} = \hat{t} g_0$ 
else
   $\hat{g} = 0$  // In the case  $g_0 = 0$  we can just set  $\hat{g} = 0$ .

```

where  $f'$  and  $f''$  denote the first and second derivatives of  $f$ . It can be shown that the derivative  $f'$  is convex and increasing on the closed interval  $[\hat{t}, 1]$ , so Newton's method always converges to the minimum. The final algorithm to evaluate the proximal operator (13) for  $0 < q < 1$  is given as Algorithm 1.

### 3.3 Evaluation of the $\|\cdot\|_{N_q}^q$ Proximal Mapping

Similar to the previous section, due to the separability we are only interested in the pointwise evaluation of  $\|\cdot\|_{N_q}^q$ . Given  $g_0 \in \mathbb{R}^{d \times k}$  we wish to evaluate the proximal mapping

$$\text{prox}_{\tau, \|\cdot\|_{N_q}^q}(g_0) = \arg \min_{g \in \mathbb{R}^{d \times k}} \|g\|_{N_q}^q + \frac{1}{2\tau} \|g - g_0\|_2^2. \quad (21)$$

In order to do so, we start with the singular value decomposition of the input argument  $g_0 = U \Sigma_{g_0} V^T$  and substitute that into (21):

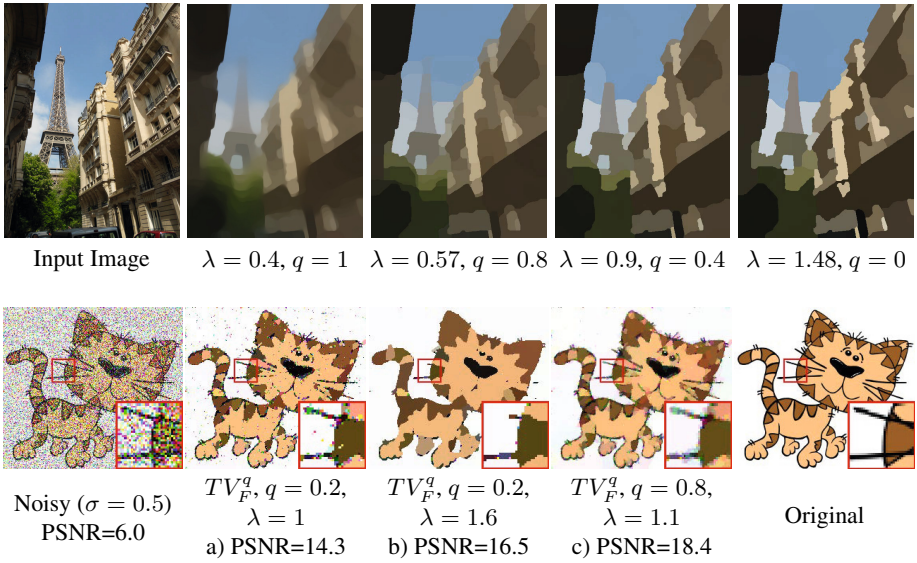
$$\arg \min_{g \in \mathbb{R}^{d \times k}} \|g\|_{N_q}^q + \frac{1}{2\tau} \|g - U \Sigma_{g_0} V^T\|_2^2. \quad (22)$$

Since the functions  $\|\cdot\|_{N_q}^q$  and  $\|\cdot\|_2$  are unitarily invariant, the optimization problem can be reduced to the following:

$$\arg \min_{\Sigma \in \mathbb{R}^{d \times k}} \|\Sigma\|_{N_q}^q + \frac{1}{2\tau} \|\Sigma - \Sigma_{g_0}\|_2^2, \quad (23)$$

where  $\Sigma \in \mathbb{R}^{d \times k}$  is a diagonal matrix. We can restrict the optimization problem (23) to diagonal matrices due to a result by Mirsky [27, Theorem 5].





**Fig. 4.** Effect of the parameter  $q$  illustrated on a color image and a denoising example. Values of  $q < 1$  lead to piecewise constant results and smaller values of  $q$  lead to higher contrast between the regions. In the second row we show the effect of the parameter  $q$  for image denoising. **a), b)** While smaller values of  $q$  lead to sharp boundaries and clearer regions, large noise outliers are not being removed since big jumps get penalized less. **c)** For such high noise levels we found values around  $q \approx 0.8$  to give the highest PSNR values as it describes a good trade-off.

As this minimization problem is separable, we can compute the  $\ell^q$  proximal mapping for each singular value in  $\Sigma_{g_0}$ . Given the solutions  $\hat{\Sigma}$  to the previous problem (23), the final solution  $\hat{g} \in \mathbb{R}^{d \times k}$  is recovered as

$$\hat{g} = U \hat{\Sigma} V^T = g_0 V \Sigma_{g_0}^+ \hat{\Sigma} V^T, \tag{24}$$

where  $\Sigma_{g_0}^+$  denotes the pseudoinverse of  $\Sigma_{g_0}$ . Note that it is not required to calculate a full singular value decomposition of  $g_0$ , but just the eigenvalue decomposition of  $g_0^T g_0$  to obtain  $V$ . In case of  $TGV_{N^q}$ , this is an eigenvalue decomposition of a  $3 \times 3$  matrix<sup>1</sup>.

## 4 Numerical Experiments

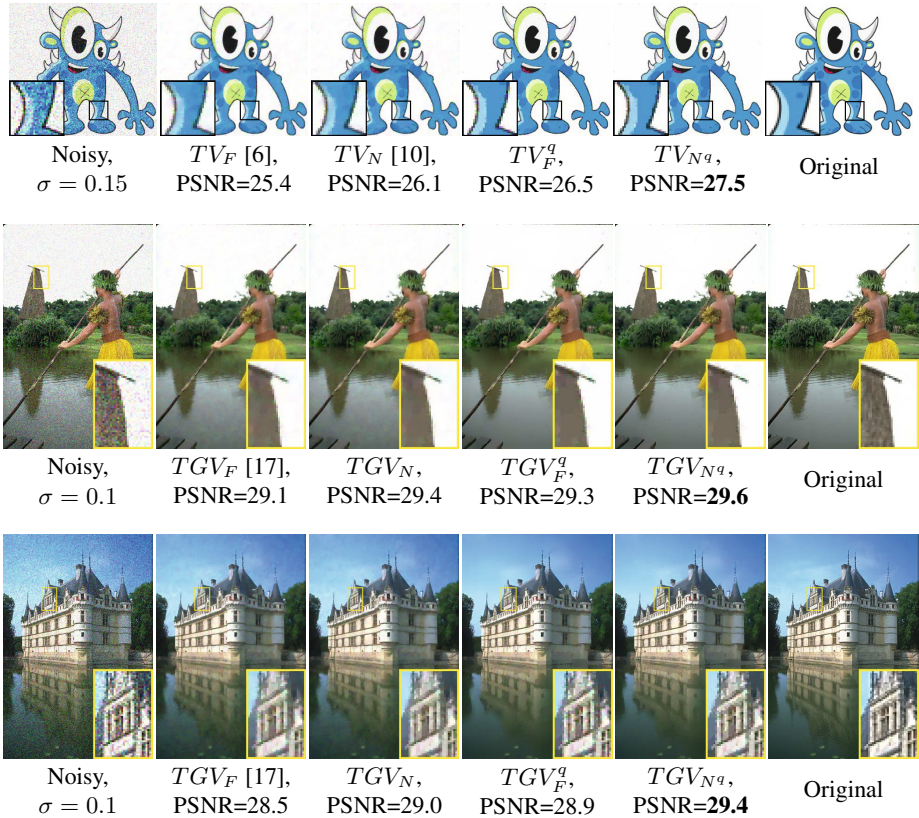
For all experiments we initialized the primal and dual variables ( $u, g$  and  $q$ ) with zero.

### 4.1 Effect of the Parameter $q$ in the $TV_F^q$ Model

In Fig. 4 we show the effect of the parameter  $q$  on a natural image for the  $TV_F^q$  model. Values of  $q < 1$  lead to piecewise constant approximations and for smaller values of

<sup>1</sup> Efficient evaluation:

<http://www.mpi-hd.mpg.de/personalhomes/globes/3x3/>



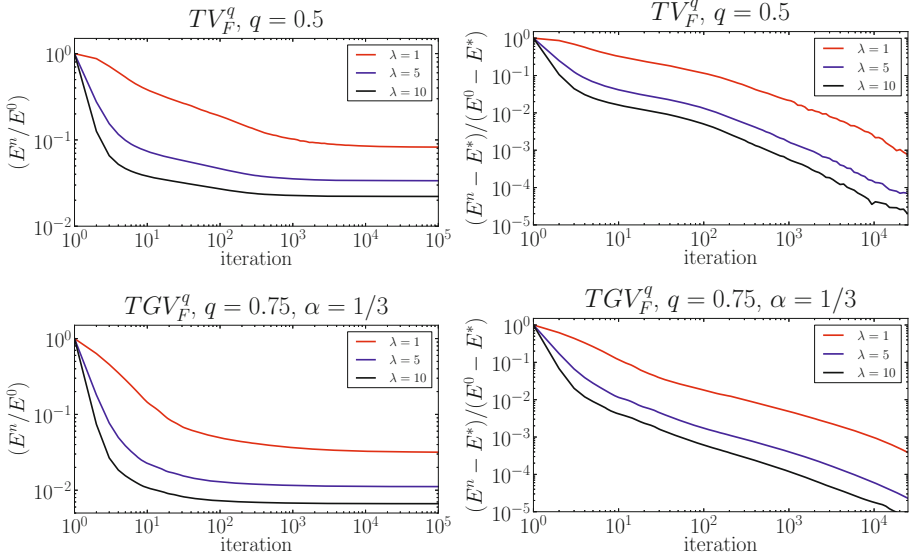
**Fig. 5.** Denoising of a piecewise constant image and natural images with  $TV^q$  ( $q = 0.5$ ) and  $TGV^q$  regularization ( $q = 0.75$ ) using the proposed primal-dual algorithm. We chose  $\alpha = 1/3$  for all the  $TGV$  experiments and the data fidelity parameters were optimized for maximal PSNR.

$q$  we observe higher contrast between the regions. That is because for smaller values of  $q$ , bigger jumps are penalized less and less until for  $q = 0$  all jumps are penalized equally. Note that the proposed algorithm produces consistent results in a sense that smaller values of  $q$  systematically lead to a higher contrast between the regions.

In the second row in Fig. 4 we illustrate how the denoising performance of the algorithm depends on the parameter  $q$ . While smaller values of  $q$  lead to desirable sharper boundaries and higher contrast, strong noise outliers do not get removed anymore due to the lower penalization of large jumps. Finding the correct value of  $q$  means finding a good trade off and values of  $q \approx 0.8$  lead to the highest PSNR for this particular noise level.

#### 4.2 Denoising of Piecewise Constant Images with $TV^q$ Regularization

In the first row of Fig. 5 and in Fig. 1 the denoising performance of the different regularizers on a piecewise constant image is shown. We chose  $q = 0.5$  and the hyperparameter



**Fig. 6.** Experimental convergence of the proposed algorithm on the  $256 \times 256$  RGB lena image for  $TV_F^q$  and  $TGV_F^q$  regularization and different data fidelities  $\lambda$ . We observe convergence for both the normalized energy  $(E^n - E^*)/(E^0 - E^*)$  and the normalized energy  $E^n/E^0$ . Similar convergence results are to be expected for the  $TV_{N^q}$  and  $TGV_{N^q}$  cases.

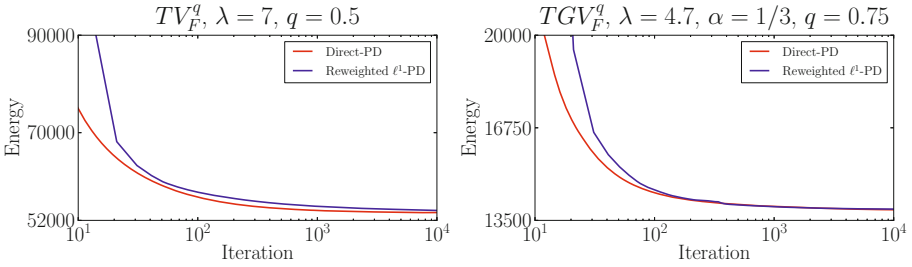
$\lambda$  was chosen in order to obtain the highest PSNR values. It can be seen that the nuclear norm reduces color artifacts at the jumps significantly and the use of nonconvex norms leads to less contrast loss and yields sharper discontinuities. Combining both aspects yields the overall highest PSNR and an improvement of 2 – 3 PSNR values over the baseline approach [6] in Fig. 1 and Fig. 5.

### 4.3 Denoising of Natural Images with $TGV^q$ Regularization

In the second and third row of Fig.5 we show the result of the proposed algorithm applied to the  $TGV$ -denoising functional for the different variants of  $TGV$  regularization. The data fidelity parameter  $\lambda$  was tuned for maximal PSNR. Again we see that nonconvex  $TGV^q$  yields sharper discontinuities and higher PSNR values while the use of the nuclear norm reduces color artefacts. The combination yields an improvement of at least  $1/2$  PSNR over [17] in the experiments in Fig. 5.

### 4.4 Convergence of the Energy

As the theoretical convergence of the algorithm is still an important open question we validated the convergence of the algorithm experimentally by precomputing a  $u^* = u^{10^5}$  as an approximation to the converged solution. It can be seen in Fig. 6 that the normalized energies  $(E(u^n) - E(u^*)) / (E(u^0) - E(u^*))$  and  $E(u^n) / E(u^0)$  converge.



**Fig. 7.** We show the energy decrease over iterations (total inner iterations for the iterative reweighted  $\ell_1$  algorithm) for the  $TV_F^q$  (left) and  $TGV_F^q$  (right) denoising examples in Fig. 1 and Fig. 5. Our proposed direct algorithm minimizes the energy functional comparably to the state-of-the-art iterative reweighted  $\ell^1$  algorithm [16].

We compare the energy decrease of the proposed method over iterations to iterative reweighted  $\ell_1$  (IRL1) optimization [16], and show the results in Fig. 7. For the iterative reweighting method we chose the smoothing parameter  $\varepsilon = 10^{-6}$  as a regularization parameter to make the  $\ell^q$  function Lipschitz continuous. The inner convex optimization problem is solved using the same primal-dual algorithm (but of course in the convex setting) and uses the same termination criterion for the inner iterations as detailed in [16]. We see that the direct application of the primal-dual method in the nonconvex setting performs overall comparably to the state-of-the-art iterative reweighted  $\ell^1$  method.

## 5 Conclusion

We proposed novel regularizers for vector valued images based on convex and non-convex relaxations of a rank minimization prior. Numerous experiments on piecewise constant and natural images show that the proposed regularizers yield overall state-of-the-art performance.

Furthermore, to deal with the nonconvex and nonsmooth optimization problem an efficient optimization method for solving related inverse problems was presented. We have shown how to efficiently find globally optimal solutions to the arising nonconvex proximal mapping. Our numerical experiments indicated that the direct application of a primal-dual splitting method in the nonconvex setting performs comparably to sequential convex programming methods. For future work we mainly wish to study the convergence properties of convex splitting methods in the nonconvex setting.

## References

1. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60, 259–268 (1992)
2. Chambolle, A., Caselles, V., Cremers, D., Novaga, M., Pock, T.: An introduction to total variation for image analysis. In: *Theoretical Foundations and Numerical Methods for Sparse Recovery*. De Gruyter (2010)

3. Attouch, H., Buttazzo, G., Michaille, G.: Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization. Mps-Siam Series on Optimization, vol. 6. SIAM (2005)
4. Blomgren, P., Chan, T.F.: Color TV: Total variation methods for restoration of vector valued images. *IEEE Trans. Image Processing* 7, 304–309 (1998)
5. Sapiro, G., Ringach, D.: Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Img. Proc.* 5(11), 1582–1586 (1996)
6. Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging* 2(4), 255–284 (2008)
7. Condat, C.: Joint demosaicking and denoising by total variation minimization. In: *IEEE Conference on Image Processing*, pp. 2781–2784 (2012)
8. Miyata, T., Sakai, Y.: Vectorized total variation defined by weighted l infinity norm for utilizing inter channel dependency. In: *2012 19th IEEE International Conference on Image Processing (ICIP)*, pp. 3057–3060 (September 2012)
9. Goldluecke, B., Cremers, D.: An approach to vectorial total variation based on geometric measure theory. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2010)
10. Lefkimmiatis, S., Roussos, A., Unser, M., Maragos, P.: Convex generalizations of total variation based on the structure tensor with applications to inverse problems. In: Pack, T. (ed.) *SSVM 2013*. LNCS, vol. 7893, pp. 48–60. Springer, Heidelberg (2013)
11. Ehrhardt, M.J., Arridge, S.: Vector-valued image processing by parallel level sets. *IEEE Trans. on Image Processing* 23, 9–18 (2014)
12. Moeller, M., Brinkmann, E., Burger, M., Seybold, T.: Color bregman tv Preprint. On ArXiv, <http://arxiv.org/abs/1310.3146>
13. Huang, J., Mumford, D.: Statistics of natural images and models. In: *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)* (1999)
14. Krishnan, D., Fergus, R.: Fast Image Deconvolution using Hyper-Laplacian Priors. In: *Proc. Neural Information Processing Systems*, pp. 1033–1041 (2009)
15. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Img. Sci.* 3(3), 492–526 (2010)
16. Ochs, P., Dosovitskiy, A., Pock, T., Brox, T.: An iterated L1 Algorithm for Non-smooth Non-convex Optimization in Computer Vision. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013)
17. Bredies, K.: Recovering piecewise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. In: Bruhn, A., Pock, T., Tai, X.-C. (eds.) *Global Optimization Methods*. LNCS, vol. 8293, pp. 44–77. Springer, Heidelberg (2014)
18. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In: *IEEE Int. Conf. on Comp. Vis. (ICCV)* (2009)
19. Esser, E., Zhang, X., Chan, T.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Img. Sci.* 3(4), 1015–1046 (2010)
20. 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)
21. Strelakovsky, E., Cremers, D.: Real-Time Minimization of the Piecewise Smooth Mumford-Shah Functional. In: *Proceedings of the European Conference on Computer Vision* (2014)
22. Storath, M., Weinmann, A., Demaret, L.: Jump-sparse and sparse recovery using potts functionals. *CoRR*, pp. 1–1 (2013)
23. Möllenhoff, T., Strelakovsky, E., Möller, M., Cremers, D.: The Primal-Dual Hybrid Gradient Method for Semiconvex Splittings (preprint, 2014), <http://arxiv.org/abs/1407.1723>

24. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial Proximal Algorithm for Non-convex Optimization. *SIAM Journal on Imaging Sciences (SIIMS)* (Preprint, 2014)
25. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse Iterative Closest Point. *Computer Graphics Forum (Symposium on Geometry Processing)* 32(5), 1–11 (2013)
26. McKelvey, J.P.: Simple transcendental expressions for the roots of cubic equations. *Amer. J. Phys.* 52(3), 269–270 (1984)
27. Mirsky, L.: Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser. (2)*, 50–59 (1960)

# A Novel Framework for Nonlocal Vectorial Total Variation Based on $\ell^{p,q,r}$ – norms

Joan Duran<sup>1,2</sup>, Michael Moeller<sup>2</sup>, Catalina Sbert<sup>1</sup>, and Daniel Cremers<sup>2</sup>

<sup>1</sup> Universitat de les Illes Balears, Department of Mathematics and Computer Science,  
Cra. de Valldemossa km. 7.5, 07122 Palma, Spain

{joan.duran,catalina.sbert}@uib.es

<sup>2</sup> Technische Universität München, Department of Mathematics and  
Computer Science, Boltzmannstrasse 3, 85748 Garching, Germany

{cremers,michael.moeller}@in.tum.de

**Abstract.** In this paper, we propose a novel framework for restoring color images using nonlocal total variation (NLTV) regularization. We observe that the discrete local and nonlocal gradient of a color image can be viewed as a 3D matrix/or tensor with dimensions corresponding to the spatial extend, the differences to other pixels, and the color channels. Based on this observation we obtain a new class of NLTV methods by penalizing the  $\ell^{p,q,r}$  norm of this 3D tensor. Interestingly, this unifies several local color total variation (TV) methods in a single framework. We show in several numerical experiments on image denoising and deblurring that a stronger coupling of different color channels – particularly, a coupling with the  $\ell^\infty$  norm – yields superior reconstruction results.

## 1 Introduction

Even after over 20 years of research, the total variation (TV) of Rudin, Osher and Fatemi [24] remains one of the most popular regularizations for image processing problems and has sparked a tremendous amount of research, e.g. on higher order TV [2, 8], cartoon texture decompositions [7, 19], the total generalized variation [4], and several extensions to vector valued data [1, 3, 5].

Motivated by the work on nonlocal algorithms [6, 26–28], Kindermann *et al.* [16] and Gilboa *et al.* [13] interpreted neighborhood filters as regularizations based on nonlocal functionals. In [14] Gilboa and Osher defined and extended the TV to the nonlocal total variation (NLTV) for grayscale image denoising. Thanks to the mathematical similarities between the local TV and the NLTV, the nonlocal framework was subsequently used for inpainting and super-resolution [21], image deblurring [18] or compressive sensing [29]. In a recent paper, Duan *et al.* [10] introduced the vectorial NLTV for image inpainting by coupling the channels with the help of the  $\ell^2$  norm.

### 1.1 The Proposed Framework

The typical approach for the NLTV regularization is to pre-compute weights  $w_{i,j}$  based on the similarity of patches around pixels  $f_i$  and  $f_j$  in the data  $f \in \mathbb{R}^N$

(using linear indexing) and then penalize the weighted difference of all pixels in the image, e.g. for NLTV denoising

$$\hat{u} = \arg \min_u \frac{\lambda}{2} \|u - f\|^2 + \sum_i \sum_j w_{i,j} |u_i - u_j|, \quad (1)$$

where  $\lambda > 0$  is a trade-off parameter.

In the literature that uses nonlocal filtering for color images,  $f \in \mathbb{R}^{N \times 3}$ , it is common to first compute the similarity weights  $w_{i,j}$  using all three color channels at the same time, but then solve the NLTV regularized problem (1) on each channel separately. In this paper, we propose to couple the different channels of a color image by determining the denoised image  $\hat{u} \in \mathbb{R}^{N \times 3}$  via

$$\hat{u} = \arg \min_u \frac{1}{2} \|u - f\|_F^2 + \|Ku\|_{p,q,r}, \quad (2)$$

where  $K$  is a linear operator such that  $Ku$  is a three dimensional tensor whose first dimension corresponds to the pixels, the second one contains the weighted differences to other pixels, and the third dimension corresponds to the color channels. Throughout the paper, we use the colon to denote all elements along the associated dimension of the data structure. For illustration purposes, let us give  $Ku$  for a color image  $u$  with only four pixels. The matrix  $(Ku)_{::,m}$  obtained by fixing the third dimension to  $m$  is

$$\begin{pmatrix} 0 & w_{1,2}(u_{1,m} - u_{2,m}) & w_{1,3}(u_{1,m} - u_{3,m}) & w_{1,4}(u_{1,m} - u_{4,m}) \\ w_{2,1}(u_{2,m} - u_{1,m}) & 0 & w_{2,3}(u_{2,m} - u_{3,m}) & w_{2,4}(u_{2,m} - u_{4,m}) \\ w_{3,1}(u_{3,m} - u_{1,m}) & w_{3,2}(u_{3,m} - u_{2,m}) & 0 & w_{3,4}(u_{3,m} - u_{4,m}) \\ w_{4,1}(u_{4,m} - u_{1,m}) & w_{4,3}(u_{4,m} - u_{3,m}) & w_{4,3}(u_{4,m} - u_{3,m}) & 0 \end{pmatrix},$$

where we used two indices for  $u$ , the first one corresponding to a linear indexing of the pixels and the second one corresponding to the color channels.

The penalty we propose to use in this paper is the  $\ell^{p,q,r}$  norm defined as

$$\|A\|_{p,q,r} = \left( \sum_i \left( \sum_j \left( \sum_k |A_{i,j,k}|^p \right)^{q/p} \right)^{r/q} \right)^{1/r}, \quad (3)$$

where we use the typical notation of  $\ell^p$  norms that any of the indices  $p$ ,  $q$ , or  $r$  being equal to infinity denotes taking the maximum of the absolute values along the corresponding dimension. The  $\ell^{p,q,r}$  norm first takes the  $\ell^p$  norm in the third matrix dimension, then the  $\ell^q$  norm in the second matrix dimension and finally the  $\ell^r$  norm of the remaining vector. Note that although the matrix  $(Ku)_{::,m}$  is an  $N \times N$  matrix (with  $N$  denoting the number of pixels in the image) for theoretical purposes, one typically only uses a few nonzero weights in practical applications. Interestingly, our new framework unifies several definitions for local TV regularization functionals, i.e. in the case where all weights  $w_{i,j}$  are zero except those corresponding to the right and the lower neighbor of each pixel (which are equal to one). As we will see in more details in the next section, up to a rearrangement of matrix dimensions, we can recover the local TV approaches in [1] by  $\|\cdot\|_{2,1,1}$ , in [3] by  $\|\cdot\|_{2,1,2}$ , and in [5] by  $\|\cdot\|_{2,2,1}$ .



## 1.2 Contributions

Our contribution in this paper is to present a generalized framework for nonlocal (and local) TV regularization of color images based on mixed  $\ell^{p,q,r}$  matrix norms. Almost all NLTV approaches proposed in the literature filter each channel separately. In fact, [10] is the only work on coupling color channels we are aware of. We propose to use more sophisticated coupling schemes that leads to novel regularization in the nonlocal case. Particularly, the new proposed NLTV- $\ell^{\infty,1,1}$  model yields superior image reconstruction results.

## 2 Vectorial Total Variation

### 2.1 Local Total Variation

In this section we will summarize the approaches that have proposed different extensions of the local TV to vector valued images, since it motivates our general framework of penalizing the  $\ell^{p,q,r}$  norms (3) of the gradient not only in the local but also in the nonlocal case.

The (local) TV regularization [24] for grayscale images is commonly used in two different forms. Let  $\Omega$  be an open bounded subset of  $\mathbb{R}^2$  and let us denote a pixel by  $x = (x_1, x_2) \in \Omega$ . The anisotropic TV for an image  $u : \Omega \rightarrow \mathbb{R}$  is defined as  $\int_{\Omega} (|\partial_{x_1} u(x)| + |\partial_{x_2} u(x)|) dx$ , whereas the isotropic TV is given by  $\int_{\Omega} \sqrt{(\partial_{x_1} u(x))^2 + (\partial_{x_2} u(x))^2} dx$ , where  $\partial_{x_1}$  and  $\partial_{x_2}$  denote the derivatives in the  $x_1$ - and  $x_2$ -direction, respectively. More general definitions for functions of bounded variation can be given by using a dual formulation, however, are omitted here and throughout the rest of this paper for the sake of simplicity.

For color images  $u : \Omega \rightarrow \mathbb{R}^M$ , where  $u(x) = (u_1(x), \dots, u_M(x))$  and  $M$  denotes the number of color channels, different TV penalizations have been proposed depending on the coupling of the spatial derivatives as well as on the coupling between color channels. All of the following examples could be formulated in an isotropic and an anisotropic version, but only the isotropic regularizations are given as examples. Although the following definitions were given different names, they can all be summarized in the  $\ell^{p,q,r}$  framework. For color images, there is the channel independent TV [1]

$$\sum_{m=1}^M \int_{\Omega} \sqrt{(\partial_{x_1} u_m(x))^2 + (\partial_{x_2} u_m(x))^2} dx, \tag{4}$$

which (in the discrete case) is the penalization of  $\|\nabla u\|_{2,1,1}$ , and the vectorial total variation with global coupling [3]

$$\sqrt{\sum_{m=1}^M \left( \int_{\Omega} \sqrt{(\partial_{x_1} u_m(x))^2 + (\partial_{x_2} u_m(x))^2} dx \right)^2}, \tag{5}$$

i.e. the penalization of  $\|\nabla u\|_{2,1,2}$ , both with the dimensions of  $\nabla u$  being ordered by colors, pixels and derivatives. The local vectorial TV (a special case of [25] and further studied in [5]) is

$$\int_{\Omega} \sqrt{\sum_{m=1}^M (\partial_{x_1} u_m(x))^2 + (\partial_{x_2} u_m(x))^2} dx, \quad (6)$$

which is nothing but the penalization of  $\|\nabla u\|_{2,2,1}$ . In [20] Miyata and Sakai recently proposed a vectorial TV regularization that couples different channels with the supremum norm,

$$\int_{\Omega} \left( \max_{1 \leq m \leq M} \{\partial_{x_1} u_m(x)\} + \max_{1 \leq m \leq M} \{\partial_{x_2} u_m(x)\} \right) dx, \quad (7)$$

which is the same as  $\|\nabla u\|_{\infty,1,1}$  (with the dimensions ordered by pixels, derivatives and colors) in our notation. We expect the supremum norm to couple the channels more than an  $\ell^1$  or an  $\ell^2$  norm which is why – different from [20] – we propose not to apply an additional color transform that decouples the channels.

Further versions of the local total variation in literature are based on penalizing singular values of the submatrices one obtains by fixing a pixel location (cf. [15, 17, 25]). While these approaches could be incorporated in our framework by not only allowing  $\ell^p$  norms but also Schatten- $p$  norms, we decided not to include this class of regularizations for the sake of simplicity.

## 2.2 Nonlocal Neighborhood Filters

All classical TV techniques for image processing describe regularity in terms of local derivative features so that only relationships between adjacent pixels are considered. The main assumption underlying TV functionals is that an image consists of connected smooth regions (objects) surrounded by sharp contours (edges). Accordingly, TV regularization is optimal to reduce noise and reconstruct the main geometrical shape (i.e. piecewise constant regions) in an image, but it fails to preserve fine structures, details and texture because they cannot be distinguished from noise.

In order to overcome the above drawbacks, the so-called neighborhood filters extend classical TV to nonlocal regularizations in which any point in an image can interact directly with any other point in the whole domain. In contrast to the local case, neighborhood filters use not only the spatial closeness between points but also closeness of intensity values in the image.

In this setting, a general description of a nonlocal filter for a grayscale image  $u$  at a point  $x \in \Omega$  is given by

$$NF[u](x) = \frac{1}{C(x)} \int_{\Omega} \omega_{u_0}(x, y) u(y) dy, \quad (8)$$

with  $u_0$  being a reference image on which the weight distribution  $\omega_{u_0}$  is computed, and  $C(x) = \int_{\Omega} \omega_{u_0}(x, y) dy$  being the normalization factor. The value  $\omega_{u_0}(x, y)$  represents the similarity of points  $x$  and  $y$  with respect to an appropriate measure. For image denoising tasks,  $u_0$  is usually chosen as the noisy image  $f$ . Generally, nonlocal approaches try to recover both, shapes and textures, within the same framework by identifying recurring structures in the whole image.

A special case of (8) is the nonlocal means (NL-means) algorithm by Buades, Coll and Morel [6], which restores a pixel  $x \in \Omega$  by averaging the intensity values of all pixels whose neighborhood looks like the neighborhood of  $x$ , i.e. by computing (8) with

$$\omega_{u_0}(x, y) = e^{-\frac{d_\rho(u_0(x), u_0(y))}{h^2}}, \tag{9}$$

where the distance  $d_\rho$  is defined by

$$d_\rho(u_0(x), u_0(y)) = \int_\Omega G_\rho(t) |u(x+t) - u(y+t)|^2 dt. \tag{10}$$

In this framework,  $G_\rho$  is a Gaussian kernel and  $h$  acts as a filtering parameter that controls the decay of the weights as a function of the Euclidean distances between patches. This method takes advantage of the fact that most natural images are self similar. The weight function  $\omega_{u_0}$  usually satisfies the conditions  $0 < \omega_{u_0} \leq 1$  and  $\int_\Omega \omega_{u_0}(x, y) dy = 1$ .

Defining the nonlocal gradient as

$$\nabla_\omega u(x, y) = (u(y) - u(x)) \sqrt{\omega(x, y)}, \quad \forall y \in \Omega, \tag{11}$$

for a nonnegative (symmetric) weight function  $\omega : \Omega \times \Omega \rightarrow \mathbb{R}$  and an image  $u : \Omega \rightarrow \mathbb{R}$ , it was shown by Gilboa and Osher in [13] that the NL-means algorithm can also be written in a variational framework. Being able to state nonlocal regularizations in a variational setting further motivated Gilboa and Osher to extend the nonlocal filtering model to a definition of the nonlocal total variation.

### 2.3 Nonlocal Total Variation

In [14] the quadratic nonlocal variational regularization was extended to the one-homogeneous nonlocal total variation. For grayscale images, there exist two variants of the NLTV approach depending on the question if the inner norm is chosen to be  $L^2(\Omega)$  or  $L^1(\Omega)$ , giving rise to

$$\int_\Omega \sqrt{\int_\Omega (u(y) - u(x))^2 \omega_{u_0}(x, y) dy} dx \tag{12}$$

which corresponds to the isotropic TV in the local case, and

$$\int_\Omega \int_\Omega |u(x) - u(y)| \sqrt{\omega_{u_0}(x, y)} dy dx, \tag{13}$$

which is related to the anisotropic TV in the local case.

A problem of nonlocal regularization strategies in inverse problems is the estimation of the weight function. In some image processing problems like denoising or deconvolution, the weight can be directly estimated from the noisy image. For many other problems, the observation  $f$  cannot be used directly and a first approximate solution is necessary for the computation of the weights (see [18] for more details).

### 3 Color Image Denoising Using NLTV– $\ell^{p,q,r}$

In the literature that uses neighborhood filters for color image denoising, it is common to first compute the similarity weights using all channels at the same time, but then solve the NLTV regularized problem

$$\min_{u_i \in \text{NL-BV}(\Omega)} \int_{\Omega} |\nabla_{\omega} u_i(x)| \, dx + \frac{\lambda}{2} \|u_i - f_i\|_2^2,$$

on each channel separately.

As pointed out in the introduction, our novel idea is to use the fact that the discretization of the nonlocal gradient or Jacobian  $\nabla_{\omega}$  on a color image  $u$  is nothing but a linear operator  $K$  which returns a three dimensional structure  $Ku$ . The three dimensions of  $Ku$  correspond to the spatial extend of the image, the weighted differences to all other pixels (which we will refer to as the *nonlocal derivatives*), and the color channels. In the local TV case, the weighted differences to all other pixels reduces to two components, i.e. the  $x$ - and the  $y$ -derivative, such that  $Ku$  typically is an  $N \times 2 \times 3$  matrix for color images, where  $N$  is the number of pixels. Motivated by the local TV regularization, we propose to apply the  $\ell^{p,q,r}$  regularization scheme (3) for coupling the different dimensions of  $Ku$  to the NLTV case as well. Note that the  $\ell^{p,q,r}$  norm is not invariant to permutation of the dimensions and, thus, it is important to make clear the order of the dimensions.

For example, similar to the local TV, we can see that the channel-by-channel regularizations (12) and (13) can be written as  $\ell^{2,1,1}$  and  $\ell^{1,1,1}$  respectively, where for the  $\ell^{2,1,1}$  case the dimensions are sorted according to pixels, color channels and derivatives. In [10] it was proposed to couple the dimensions in analogy to (5), by considering a regularization of the form

$$\sqrt{\sum_{m=1}^M \left( \int_{\Omega} \sqrt{\int_{\Omega} (u(y) - u(x))^2 \omega_{u_0}(x, y) \, dy} \, dx \right)^2} \tag{14}$$

such that their approach can be denoted by a nonlocal  $\ell^{2,1,2}$  regularization in our framework whenever the dimensions are ordered by channels, pixels and nonlocal derivatives. Interestingly, [10] is the only work on coupling color channels in an NLTV approach the authors are aware of. In this paper, we will show that improvements similar to the ones made by using more sophisticated coupling schemes for the local TV can be achieved for the NLTV as well.

For discussing the question what kind of matrix norm is the best candidate for the NLTV regularization we have to understand what kind of properties they try to impose on the reconstructed image. The  $\ell^{1,1,1}$  regularization penalizes each channel independent of the others, which makes it only appropriate when the channel correlation is insignificant. On the contrary, the  $\ell^{\infty,1,1}$  norm introduces a strong inter-channel coupling with the help of the supremum norm. In between the above-mentioned norms,  $\ell^{2,1,1}$  and  $\ell^{2,2,1}$  use a channel coupling in terms of the Euclidean norm. The question whether a strong or a weak coupling leads

to better results depends on the type of correlation in the data. As we will see in the numerical results, natural images have a rather strong inter channel correlation such that we found the  $\ell^\infty$  coupling of the color channels to be the most successful approach, since it worked well for suppressing color artifacts.

## 4 Numerical Implementation

### 4.1 Computation of the Weights

For computational purposes, the nonlocal regularization term is limited to interact only between pixels at a certain distance (the so-called *search window*), i.e., the weight function  $\omega(x, y)$  is zero for all points  $x$  and  $y$  with  $\|x - y\|_\infty > K$ , for a certain parameter  $K > 0$ . More precisely, the similarity weight between  $x$  and  $y$  is determined as

$$\omega(x, y) = \begin{cases} \frac{1}{C(x)} e^{-\frac{1}{h^2} \sum_{t \in \mathcal{N}_0} \|u_0(x+t) - u_0(y+t)\|^2} & \text{if } \|x - y\|_\infty \leq K, \\ 0 & \text{otherwise,} \end{cases}$$

where  $\mathcal{N}_0$  is a discrete window centered at 0 (the *comparison window*). The normalizing factor  $C(x)$  is defined by

$$C(x) = \sum_{\{y: \|x-y\|_\infty \leq K\}} e^{-\frac{1}{h^2} \sum_{t \in \mathcal{N}_0} \|u_0(x+t) - u_0(y+t)\|^2}.$$

Therefore, the matrix obtained by fixing a color channel and looking at the remaining structure in the pixel and nonlocal derivative dimensions is sparse since only a few weights are nonzero.

Note that the Gaussian kernel  $G_\rho$  introduced in the mathematical formulation (9)-(10) was omitted here since according to our numerical experiments it is only necessary when the size of the comparison window is considerably larger. Due to the fast decay of the exponential kernel, large Euclidean distances lead to nearly zero weights, acting as an automatic threshold.

### 4.2 Minimization Algorithm and Proximity Operators

It is remarkable that all variants of different matrix norms imposed on the (possibly nonlocal) gradient of the color channels can be solved very efficiently by using the same splitting technique. In this paper, we use the primal-dual hybrid gradient method [9, 12, 22, 30] for solving minimization problems of the form

$$\min_u G(u) + F(Ku),$$

where  $G$  and  $F$  are proper convex functionals and  $K$  is a linear operator, which in our case is the nonlocal gradient. The algorithm iteratively computes

$$\begin{aligned}
 u^{n+1} &= \text{prox}_{\tau G}(u^n - \tau K^T q^n), \\
 \bar{u}^{n+1} &= 2u^{n+1} - u^n, \\
 g^{n+1} &= \text{prox}_{\frac{1}{\sigma} F} \left( K\bar{u}^{n+1} + \frac{q^n}{\sigma} \right), \\
 q^{n+1} &= q^n + \sigma(K\bar{u}^{n+1} - g^{n+1}),
 \end{aligned} \tag{15}$$

where  $\tau > 0$  and  $\sigma > 0$  are chosen such that  $\tau\sigma\|K\|^2 \leq 1$ . The only thing that changes when varying the regularizer  $F$  is the proximity operator, a generalization of the projection that is defined as

$$\text{prox}_{\alpha F}(v) = \arg \min_g \frac{1}{2} \|g - v\|_2^2 + \alpha F(g).$$

The key to obtaining a fast algorithm based on (15) is the fast evaluation of the proximity operators. In the following we will discuss the implementation of  $\text{prox}_{\alpha F}(v)$  for the matrix norms discussed previously.

- $\ell^{1,1,1}$ : The proximity operator decouples for all pixels such that

$$(\text{prox}_{\alpha \ell^{1,1,1}}(v))_{i,j,k} = \text{sign}(v_{i,j,k}) \max(|v_{i,j,k}| - \alpha, 0),$$

which is known as *shrinkage* or *soft thresholding* operator.

- $\ell^{2,1,1}$ : As for instance known from the isotropic total variation, the  $\ell^2$  norm leads to the generalized shrinkage formula

$$(\text{prox}_{\alpha \ell^{2,1,1}}(v))_{i,j,k} = \frac{v_{i,j,k}}{\|v_{i,j,:}\|_2} \max(\|v_{i,j,:}\|_2 - \alpha, 0),$$

where recall that the colon denotes all elements along that dimension.

- $\ell^{2,2,1}$ : Extending the previous result by one additional dimension is straightforward. Therefore, the corresponding proximity operator is simply given by

$$(\text{prox}_{\alpha \ell^{2,2,1}}(v))_{i,j,k} = \frac{v_{i,j,k}}{\|v_{i,:,:}\|_{2,2}} \max(\|v_{i,:,:}\|_{2,2} - \alpha, 0).$$

- $\ell^{\infty,1,1}$ : Due to the outer  $\ell^1$  norms, the  $\ell^{\infty,1,1}$  problem decouples in the first and second dimensions. We are left with an  $\ell^\infty$  regularized problem at each component with a dual formulation of the form

$$\frac{1}{\alpha} \left\| \frac{v_{i,j,:}}{\alpha} - p_{i,j,:} \right\|_2^2 \text{ such that } \|p_{i,j,:}\|_1 \leq 1.$$

The primal variable can be recovered by

$$(\text{prox}_{\alpha \ell^{\infty,1,1}}(v))_{i,j,k} = v_{i,j,k} - \alpha p_{i,j,k}$$

as one can see by Moreau's identity (cf. [23]). Several efficient algorithms for projecting onto the  $\ell^1$  ball exist [11].

## 5 Experimental Results

In this section, we present a performance comparison of different NLTV- $\ell^{p,q,r}$  regularizations for image denoising and deblurring. In all cases, the dimensions of the nonlocal gradient are ordered by pixels, weighted differences and colors.

In all our numerical experiments we used the degraded image  $f$  for computing the weight function  $\omega$ , with a search window of  $11 \times 11$  pixels and a comparison window of  $3 \times 3$  pixels.

### 5.1 Image Denoising

In this section we provide a detailed comparison of the different matrix NLTV approaches for image denoising. We minimize the energy  $\frac{\lambda}{2}\|u - f\|^2 + F(\nabla_{\omega}u)$  using all matrix norms discussed throughout the paper as regularizations. For the minimization we used the primal-dual hybrid gradient algorithm (15) along with the proximity operators described in Section 4. We used all images from the Kodak database<sup>1</sup> with values being relative to the intensity range  $[0, 255]$ , and artificially added zero mean Gaussian noise with standard deviation 12.75. On the first image, we ran the minimization for each regularization for ten different data fidelity parameters  $\lambda > 0$  and ten different filtering parameters  $h > 0$ , and determined the value at which the highest peak signal to noise ratio (PSNR) is reached. These optimal parameters were then used to run each of the regularization methods on the other 23 Kodak images.

All PSNR values of the reconstructions using all kinds of regularizations are shown in Table 1. By and large, all methods led to an improvement between six and seven PSNR points. We can also see that the strong channel-coupling  $\ell^{\infty,1,1}$  regularization was superior for 22 out of the 24 images. This indicates that natural images typically have a high inter-channel correlation. Only on images number 7 and 12 the  $\ell^{2,1,1}$  norm gave the best result. As expected,  $\ell^{1,1,1}$  regularization shows one of the worst performances since it does not couple the colors. Interestingly, the  $\ell^{2,1,1}$  has outperformed the  $\ell^{2,2,1}$  regularization. Apparently, decoupling the nonlocal derivatives was better suited than coupling them with an  $\ell^2$  norm for the Kodak data set. Anyway,  $\ell^{2,1,1}$  regularization provides results close to the best ones but with the advantage of not requiring a projection as the infinity norm does and, thus, reducing the computational cost.

Since the PSNR values do not always correlate well with the visually perceived image quality, Figure 1 shows an example for the optimal results each method obtained on parts of Kodak image 23 (i.e., we computed the best  $\lambda$  and  $h$  values for this particular image in terms of the PSNR). We can see that the  $\ell^{\infty,1,1}$  result has less color artifacts than the three other methods thanks to a stronger channel coupling.

By updating the weighting function at each iteration, one would expect to increase the quality of the results at some computational cost. Hence, we tried to iterate the NLTV denoising with a recomputation of the weights on the current estimated reconstruction. Surprisingly, the results did not improve those

<sup>1</sup> See <http://r0k.us/graphics/kodak/>

**Table 1.** Comparison of the best PSNR values achieved by each matrix NLTV- $\ell^{p,q,r}$  method on each of the 24 Kodak images for denoising

	1	2	3	4	5	6	7	8	9	10	11	12
Noisy	26.03	26.13	26.10	26.06	26.14	26.17	26.08	26.15	26.03	26.04	26.13	26.07
$\ell^{1,1,1}$	30.57	33.57	35.02	33.50	31.23	31.78	34.38	31.01	34.46	34.20	32.33	34.19
$\ell^{2,1,1}$	30.66	33.78	35.69	33.75	31.36	31.88	<b>35.06</b>	31.14	35.03	34.68	32.45	<b>34.51</b>
$\ell^{2,2,1}$	30.60	33.67	35.35	33.50	31.21	31.76	34.66	31.07	34.77	34.40	32.32	34.28
$\ell^{\infty,1,1}$	<b>30.81</b>	<b>33.82</b>	<b>35.73</b>	<b>33.76</b>	<b>31.44</b>	<b>31.99</b>	35.03	<b>31.20</b>	<b>35.05</b>	<b>34.72</b>	<b>32.52</b>	34.50
	13	14	15	16	17	18	19	20	21	22	23	24
Noisy	26.11	26.08	26.31	26.04	26.24	26.15	26.07	26.98	26.06	26.06	26.09	26.14
$\ell^{1,1,1}$	29.28	31.64	33.88	33.13	33.79	31.41	32.74	34.40	32.21	32.31	35.17	31.78
$\ell^{2,1,1}$	29.37	31.70	34.24	33.29	34.18	31.54	32.83	34.74	32.50	32.39	35.93	32.03
$\ell^{2,2,1}$	29.30	31.58	34.02	33.10	33.81	31.39	32.78	34.67	32.36	32.27	35.33	31.83
$\ell^{\infty,1,1}$	<b>29.60</b>	<b>31.77</b>	<b>34.29</b>	<b>33.35</b>	<b>34.20</b>	<b>31.66</b>	<b>32.89</b>	<b>34.78</b>	<b>32.61</b>	<b>32.45</b>	<b>35.94</b>	<b>32.12</b>

displayed in Table 1. This is due to, first, the noise level we used for the experiments that makes the input data a valid image for computing the weights and, second, because of the weighting function being computed on a more and more smoothed version of the underlying true image.

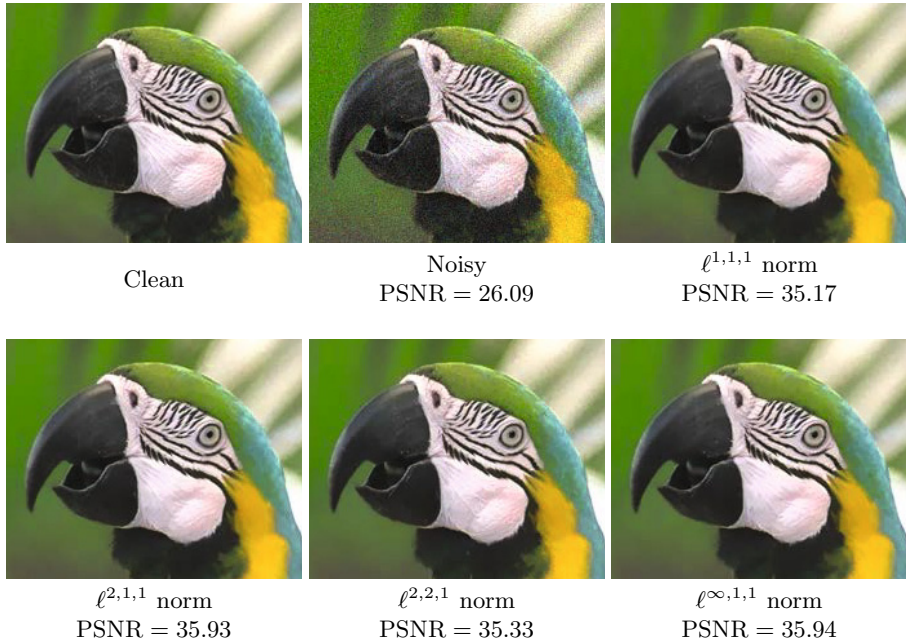
In addition to the comparison between all proposed  $\ell^{p,q,r}$  norms, it is interesting to compare the performance of local and nonlocal methods. For this purpose, Figure 2 displays the optimal results provided by each method on parts of Kodak image 3. As one expects, the nonlocal regularizations perform better than the local ones for removing noise and color artifacts while preserving fine structures and textures. For instance, note that the wood pattern has almost disappeared in all images provided by local regularizations, which does not happen in such considerable way in the results obtained with NLTV, although of course, some information from the true image is lost during the nonlocal filtering process as well. Interestingly, the gain in image quality by going from an  $\ell^{1,1,1}$  coupling to an  $\ell^{\infty,1,1}$  coupling is much bigger for the local TV than for the NLTV.

## 5.2 Image Deblurring

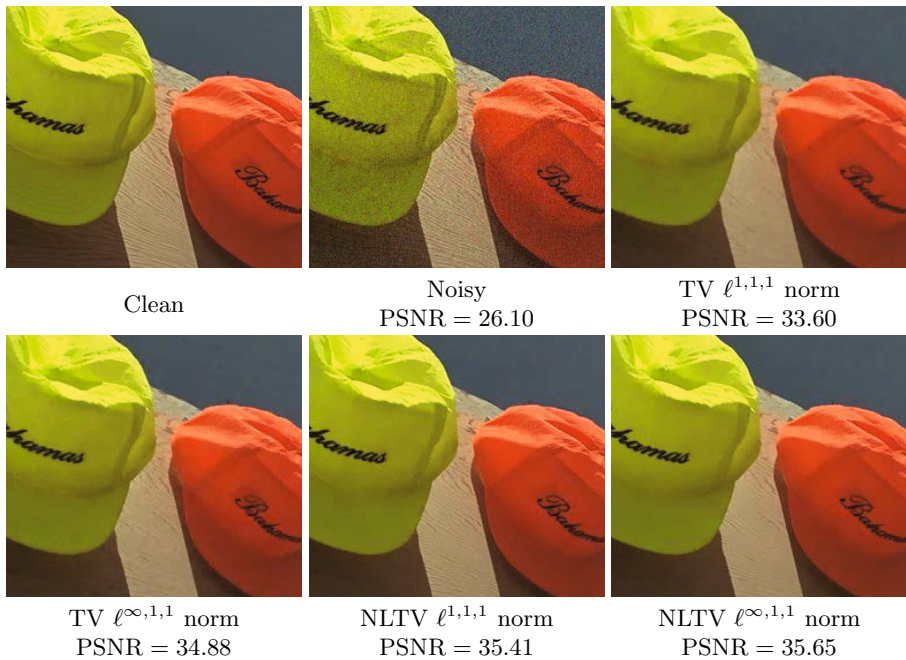
In general, the variational formulation of deblurring involves the minimization of the energy  $\frac{\lambda}{2}\|Au - f\|^2 + F(\nabla_{\omega}u)$ , where  $A$  is a linear operator modeling the degradation of  $u$  caused by the blur. For the following experiment we focus on deconvolution, which refers to the case where  $Au = \varphi * u$ , and  $\varphi$  is a Gaussian convolution kernel. In this case, the proximity operator of the fidelity term is given by  $\text{prox}_{\tau G}(v) = (I + \tau A^*A)^{-1}(v + \tau A^*f)$ . The computation of  $(I + \tau A^*A)^{-1}$  can be implemented very efficiently using the Fourier theorem based on which the convolution becomes a multiplication in the Fourier domain.

We used Kodak image 23 and optimized  $\lambda$  and  $h$  to yield the highest PSNR values. The corrupted data was simulated by convolving the ground truth with



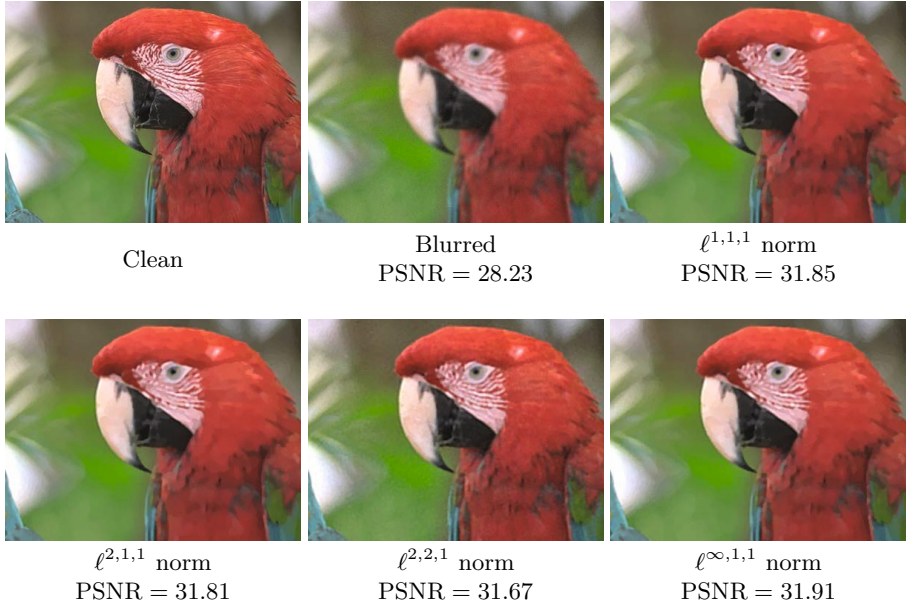


**Fig. 1.** Visual comparison of the denoising results obtained by different NLTV- $\ell^{p,q,r}$  methods on Kodak image 23



**Fig. 2.** Visual comparison of the denoising results obtained by local and nonlocal TV with  $\ell^{1,1,1}$  and  $\ell^{\infty,1,1}$  norms on Kodak image 3

a Gaussian kernel of standard deviation 1.75 and adding white Gaussian noise of standard deviation 5. Figure 3 shows the results both visually and in terms of the PSNR values. Note that in all images the blur has been reduced although some details appearing in the original image cannot be recovered from the corrupted data. As in the denoising case, coupling the channels with the help of the  $\ell^\infty$  norm leads to a higher PSNR value, although the visual difference between all restored images is small for this particular case.



**Fig. 3.** Visual comparison of the deblurring results obtained by different  $\ell^{p,q,r}$  NLTV methods on Kodak image 23

## 6 Conclusions

In this paper we proposed a general framework for local and nonlocal TV regularizations of color images by phrasing it as the penalization with a mixed  $\ell^{p,q,r}$  matrix norm. For the latter, the gradient of a color image was interpreted as a three-dimensional tensor using pixels, (local or nonlocal) derivatives and color channels as the three dimensions. We considered then several  $\ell^{p,q,r}$  norms for regularizing the gradient matrix, which led to novel regularizations in the non-local case. We discussed the numerical implementation of our framework with an efficient primal-dual algorithm and particularly focused on the evaluation of different  $\ell^{p,q,r}$  proximity operators.

We showed a detailed performance comparison of different matrix NLTV approaches for denoising, as well as an extension to image deblurring. Based on

our experiments, we exhibited the superiority of using  $\ell^\infty$  inter-channel coupling for a stronger suppression of color artifacts in natural images. Future work will mainly concentrate on the development of more  $\ell^{p,q,r}$  norms for vectorial TV and NLTV regularizations, the study of permutations in the dimensions of the data structure, a deeper insight on the mathematical properties of these matrix norms, and the extension of our approach to other image reconstruction problems.

**Acknowledgements.** J.D. and C.S. were supported by the Ministerio de Ciencia e Innovación under grant TIN2011-27539. M.M. and D.C. were supported by ERC starting grant “Convex Vision”. Additionally, J.D. acknowledges the fellowship of the Conselleria d’Educació, Cultura i Universitats of the Govern de les Illes Balears for the realization of his Ph.D. thesis, which has been selected under an operational program co-financed by the European Social Fund.

## References

1. Attouch, H., Buttazzo, G., Michaille, G.: Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization. SIAM Series on Optimization, vol. 6. SIAM (2005)
2. Benning, M., Brune, C., Burger, M., Mueller, J.: Higher-order TV methods—enhancement via Bregman iteration. *J. Sci. Comput.* 54, 269–310 (2013)
3. Blomgren, P., Chan, T.F.: Color TV: Total variation methods for restoration of vector valued images. *IEEE Trans. Image Proc.* 7, 304–309 (1998)
4. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* 3(3), 492–526 (2010)
5. Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Probl. Imag.* 2(4), 255–284 (2008)
6. Buades, A., Coll, B., Morel, J.-M.: A review of image denoising algorithms, with a new one. *SIAM Multiscale Model. Simul.* 4(2), 490–530 (2005)
7. Buades, A., Le, T.M., Morel, J.-M., Vese, L.A.: Fast cartoon + texture image filters. *IEEE Trans. Image Proc.* 19(8), 1978–1986 (2010)
8. Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numer. Math.* 76, 167–188 (1997)
9. 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)
10. Duan, J., Pan, Z., Tai, X.C.: Color texture image inpainting using the non local CTV model. *J. Signal Information Process.* 4(3B), 43–51 (2013)
11. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the  $\ell^1$ -ball for learning in high dimensions. In: Proc. of the 25th International Conference on Machine Learning, pp. 272–279. ACM, New York (2008)
12. Esser, E., Zhang, X., Chan, T.: 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)
13. Gilboa, G., Osher, S.: Nonlocal image regularization and supervised segmentation. *SIAM Multiscale Model. Simul.* 6(2), 595–630 (2007)

14. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *SIAM Multiscale Model. Simul.* 7(3), 1005–1028 (2008)
15. Goldluecke, B., Cremers, D.: An approach to vectorial total variation based on geometric measure theory. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 237–333 (2010)
16. Kindermann, S., Osher, S., Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. *SIAM Multiscale Model. Simul.* 4(4), 1091–1115 (2005)
17. Lefkimmatis, S., Roussos, A., Unser, M., Maragos, P.: Convex generalizations of total variation based on the structure tensor with applications to inverse problems. In: Pack, T. (ed.) *SSVM 2013. LNCS*, vol. 7893, pp. 48–60. Springer, Heidelberg (2013)
18. Lou, Y., Zhang, X., Osher, S., Bertozzi, A.: Image recovery via nonlocal operators. *J. Sci. Comput.* 42, 185–197 (2010)
19. Meyer, Y.: *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Soc. (2001)
20. Miyata, T., Sakai, Y.: Vectorized total variation defined by weighted L infinity norm for utilizing inter channel dependency. In: *Proc. of the 19th IEEE International Conference on Image Processing (ICIP)*, pp. 3057–3060 (2012)
21. Peyré, G., Bougleux, S., Cohen, L.: Non-local regularization of inverse problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III. LNCS*, vol. 5304, pp. 57–68. Springer, Heidelberg (2008)
22. Pock, T., Chambolle, A., Bischof, H., Cremers, D.: A convex relaxation approach for computing minimal partitions. In: *Proc. of International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 810–817 (2009)
23. Rockafellar, R.T.: *Convex Analysis, Princeton Landmarks in Mathematics*. Reprint of the 1970 original, Princeton Paperbacks (1997)
24. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60, 259–268 (1992)
25. Sapiro, G., Ringach, D.L.: Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Image Proc.* 5(11), 1582–1586 (1996)
26. Smith, S.M., Brady, J.M.: SUSAN - A new approach to low level image processing. *Int. J. Comput. Vis.* 23, 45–78 (1997)
27. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proc. of the 6th International Conference on Computer Vision*, pp. 839–846 (1998)
28. Yaroslavsky, L.: *Digital Picture Processing*. Springer, New York (1985)
29. 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)
30. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical Report 08-34, UCLA Cam Report (2008)

# Inpainting of Cyclic Data Using First and Second Order Differences

Ronny Bergmann<sup>1,\*</sup> and Andreas Weinmann<sup>2</sup>

<sup>1</sup> Department of Mathematics, Technische Universität Kaiserslautern,  
Kaiserslautern, Germany

`bergmann@mathematik.uni-kl.de`

<sup>2</sup> Department of Mathematics, Technische Universität München and Fast Algorithms  
for Biomedical Imaging Group, Helmholtz-Zentrum München, München, Germany  
`andreas.weinmann@tum.de`

**Abstract.** Cyclic data arise in various image and signal processing applications such as interferometric synthetic aperture radar, electroencephalogram data analysis, and color image restoration in HSV or LCh spaces. In this paper we introduce a variational inpainting model for cyclic data which utilizes our definition of absolute cyclic second order differences. Based on analytical expressions for the proximal mappings of these differences we propose a cyclic proximal point algorithm (CPPA) for minimizing the corresponding functional. We choose appropriate cycles to implement this algorithm in an efficient way. We further introduce a simple strategy to initialize the unknown inpainting region. Numerical results both for synthetic and real-world data demonstrate the performance of our algorithm.

**Keywords:** Inpainting, variational models with higher order differences, cyclic data, phase-valued data, cyclic proximal point algorithm.

## 1 Introduction

Image inpainting is a frequently arising problem in image processing. Examples are restoring scratches in photographs, removal of superimposed objects, dealing with areas removed by a user, digital zooming, edge decoding, restoration of defects in audio/video recordings or in seismic data. The term ‘inpainting’ first appeared in [6], but earlier work on disocclusions was already done, e.g., in [13,38]. In this respect also interpolation, approximation, and extrapolation problems may be viewed as inpainting problems. Inpainting is a very active field of research which has been tackled by various approaches. For a good overview we refer to the (tutorial) papers [10,12,17,30]. While exemplar-based and sparsity-based (dictionary/frame/tensor) methods are in general better suited for filling large texture areas, diffusion-based and corresponding variational techniques show good results for natural images. The total variation (TV) regularized model proposed in [45] for denoising was first applied to inpainting in [4,15]. It was later also

---

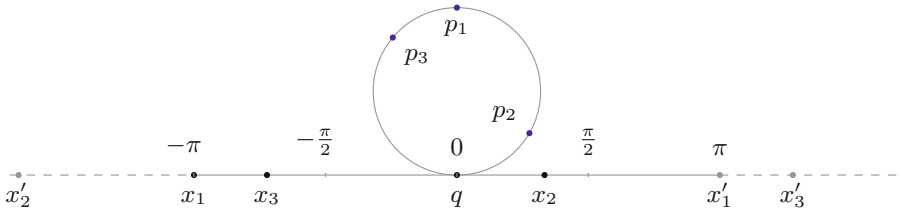
\* Corresponding author.

used in combination with other methods, however, the TV regularizer typically introduces a staircasing effect in the corresponding minimizer. A simple method to avoid these artifacts consists in the incorporation of second order derivatives into the model. Indeed, starting with [14] various approaches with higher order derivatives have been proposed, see, e.g., [9,16,19,31,33,35,40,46,47,48,51]. In this paper, we address the problem of inpainting cyclic data using a variational model with second order cyclic differences. In general, manifold-valued data processing has recently gained a lot of interest. Examples are wavelet-type multiscale transforms for manifold data [29,42,52] and manifold-valued partial differential equations [18,28]. Also statistical issues on Riemannian manifolds have been considered [22,23,41], in particular the statistics of circular data [21,32].

*Related work.* Although very popular for processing images with scalar and vector-valued data, TV minimization has only very recently been applied to cyclic structures. From a theoretical point of view TV functionals for manifold-valued functions have been studied in [26,27]. These papers extend the previous work [25] on  $\mathbb{S}^1$ -valued functions where, in particular, the existence of minimizers of certain energies is shown in the space of functions with bounded total cyclic variation. First order TV minimization for cyclic data in image processing has been investigated in [49,50]. The authors unwrap the data to the real line and propose an algorithm based on functional lifting which takes the periodicity into account. In particular, they also consider cyclic inpainting. An algorithm for TV minimization on Riemannian manifolds was proposed in [34]. The approach is based on a reformulation as a multilabel optimization problem with an infinite number of labels. Using convex relaxation techniques, the resulting hard optimization problem is approximated which also requires the discretization of the manifold. Another approach for denoising manifold-valued data via first order TV minimization was given in [53]. The authors propose cyclic and parallel proximal point algorithms which will also be our method of choice.

*Contributions.* We propose two models for inpainting of cyclic data using first and second order absolute cyclic differences. In our preprint [5] we introduced absolute second order differences for cyclic data in a sound way. We further deduced analytical expressions for the proximal mappings of these differences. Here, our first model considers the noise free inpainting situation, whereas the second one handles simultaneously inpainting and denoising. The variational formulations allow for the decomposition of the whole functionals into simpler ones, for each of which the proximal mappings are given explicitly. Thus, the minimizers can be computed efficiently by a cyclic proximal point method. We propose a suitable initialization of the inpainting area. We demonstrate by numerical examples the strength of our algorithm. Compared to [49,50] we neither have to employ Fréchet means nor to discretize the manifold.

*Organization.* In Sec. 2 we introduce our absolute second order cyclic differences and provide analytical expressions for their proximal mappings. Then, in Sec. 3, we introduce our inpainting model and propose a procedure to initialize the



**Fig. 1.** Three points  $p_j$ ,  $j = 1, 2, 3$ , on the circle and their possible unwrappings  $x_1, x_2, x_3 \in [-\pi, \pi)$  with respect to the origin  $q$  and other possibilities  $x'_2, x'_1, x'_3$  and  $x_2, x_3, x'_1$  that correspond to the same situation on  $\mathbb{S}^1$ . These are taken into account for  $d(x; w)$ .

unknown inpainting region. Sec. 4 describes the cyclic proximal point algorithm. Finally, Sec. 5 contains numerical examples. Conclusions and directions of future work are given in Sec. 6.

## 2 Absolute First and Second Order Cyclic Differences and Their Proximal Mappings

Let  $\mathbb{S}^1 := \{p_1^2 + p_2^2 = 1 : p = (p_1, p_2)^T \in \mathbb{R}^2\}$  be the unit circle endowed with the *geodesic distance*  $d_{\mathbb{S}^1}(p, q) := \arccos(\langle p, q \rangle)$ . Given a base point  $q \in \mathbb{S}^1$ , the *exponential map*  $\exp_q : \mathbb{R} \rightarrow \mathbb{S}^1$  from the tangent space  $T_q\mathbb{S}^1 \simeq \mathbb{R}$  of  $\mathbb{S}^1$  at  $q$  onto  $\mathbb{S}^1$  is defined by

$$\exp_q(x) = R_x q, \quad R_x := \begin{pmatrix} \cos x & -\sin x \\ \sin x & \cos x \end{pmatrix}.$$

This map is  $2\pi$ -periodic, i.e.,  $\exp_q(x) = \exp_q((x)_{2\pi})$  for any  $x \in \mathbb{R}$ , where  $(x)_{2\pi}$  denotes the unique point in  $[-\pi, \pi)$  such that  $x = 2\pi k + (x)_{2\pi}$ ,  $k \in \mathbb{Z}$ . For  $p, q \in \mathbb{S}^1$  with  $\exp_q(0) = q$ , there is a unique  $x \in [-\pi, \pi)$  satisfying  $\exp_q(x) = p$ . Given such representants  $x_j \in [-\pi, \pi)$  of  $p_j \in \mathbb{S}^1$ ,  $j = 1, 2$  centered at an arbitrary base point  $q \in \mathbb{S}^1$  the geodesic distance becomes

$$d_{\mathbb{S}^1}(p_1, p_2) = d(x_1, x_2) = \min_{k \in \mathbb{Z}} |x_2 - x_1 + 2\pi k| = |(x_2 - x_1)_{2\pi}|$$

which is of course independent of  $q$ . We want to define higher order differences for points  $(p_j)_{j=1}^d \in (\mathbb{S}^1)^d$  using their representants  $x := (x_j)_{j=1}^d \in [-\pi, \pi)^d$ . To achieve independence of the base point the differences must be shift invariant modulo  $2\pi$ , see Fig. 1. Let  $1_d$  denote the vector with  $d$  entries 1. We define the *absolute cyclic difference* of  $x \in [-\pi, \pi)^d$  with respect to a difference filter  $w \in \mathbb{R}^d$  with  $\langle w, 1_d \rangle = 0$  by

$$d(x; w) := \min_{\alpha \in \mathbb{R}} \langle [x + \alpha 1_d]_{2\pi}, w \rangle, \tag{1}$$

where  $[x]_{2\pi}$  denotes the componentwise application of  $(t)_{2\pi}$  if  $t \neq (2k + 1)\pi$ ,  $k \in \mathbb{Z}$  and  $[(2k + 1)\pi]_{2\pi} = \pm\pi$ ,  $k \in \mathbb{Z}$ . Let

$$b_1 := (-1, 1)^T \quad \text{and} \quad b_2 := (1, -2, 1)^T, \quad b_{1,1} := (-1, 1, 1, -1)^T$$

be a first order (forward) difference filter, and two second order difference filters, respectively. For  $w \in \mathcal{B} := \{b_1, b_2, b_{1,1}\}$  we have shown in our accompanying preprint [5] that the absolute cyclic differences can be rewritten as

$$d(x; w) = (\langle x, w \rangle)_{2\pi}. \tag{2}$$

Clearly, we have  $d(x; b_1) = d(x_1, x_2)$ . Interestingly, the definition (1) and (2) do not coincide, e.g., for third order cyclic differences [5].

Next we are interested in proximal mappings of absolute cyclic differences. Recall that for a proper, closed, convex function  $\varphi : \mathbb{R}^N \rightarrow (-\infty, +\infty]$  and  $\lambda > 0$  the *proximal mapping*  $\text{prox}_{\lambda\varphi} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  is well defined by

$$\text{prox}_{\lambda\varphi}(f) := \arg \min_{x \in \mathbb{R}^N} \frac{1}{2} \|f - x\|_2^2 + \lambda\varphi(x).$$

We introduce the proximal mapping  $\text{prox}_{\lambda d(\cdot; w)} : (\mathbb{S}^1)^d \rightarrow (\mathbb{S}^1)^d$  by

$$\text{prox}_{\lambda d(\cdot; w)}(f) := \arg \min_{x \in [-\pi, \pi]^d} \frac{1}{2} \sum_{j=1}^d d(x_j, f_j)^2 + \lambda d(x; w), \quad \lambda > 0.$$

The following theorem determines the proximal mapping analytically for  $w \in \mathcal{B}$ . In particular, the mapping is single-valued for  $f \in [-\pi, \pi]^d$  with  $|(\langle f, w \rangle)_{2\pi}| < \pi$  and two-valued for  $|(\langle f, w \rangle)_{2\pi}| = \pi$ . Note that for  $w = b_1$  the second case appears exactly if  $f_1$  and  $f_2$  are antipodal points. For a proof we refer to our preprint [5].

**Theorem 1.** For  $w \in \mathcal{B}$  set  $s := \text{sgn}(\langle f, w \rangle)_{2\pi}$ . Let  $f \in [-\pi, \pi]^d$ , where  $d$  is adapted to the respective length of  $w$ ,  $\lambda > 0$ , and  $m := \min \left\{ \lambda, \frac{|(\langle f, w \rangle)_{2\pi}|}{\|w\|_2^2} \right\}$ .

- i) If  $|(\langle f, w \rangle)_{2\pi}| < \pi$ , then  $\text{prox}_{\lambda d(\cdot; w)}(f) = (f - s m w)_{2\pi}$ .
- ii) If  $|(\langle f, w \rangle)_{2\pi}| = \pi$ , then  $\text{prox}_{\lambda d(\cdot; w)}(f) = \{(f + s m w)_{2\pi}, (f - s m w)_{2\pi}\}$ .

For handling noisy data we will further need the following proximal mapping:

**Theorem 2.** For  $f, g \in [-\pi, \pi]^N$  we have

$$\begin{aligned} \text{prox}_{\lambda d(\cdot, f)}(g) &:= \arg \min_x \sum_{j=1}^N (d(g_j, x_j)^2 + \lambda d(f_j, x_j)^2) \\ &= \left( \frac{g + \lambda f}{1 + \lambda} + \frac{\lambda}{1 + \lambda} 2\pi v \right)_{2\pi}, \end{aligned}$$

where  $d(g, f) := \sum_{j=1}^N d(g_j, f_j)$  and  $v = (v_j)_{j=1}^N \in \mathbb{R}^N$  is defined by

$$v_j := \begin{cases} 0 & \text{if } |g_j - f_j| \leq \pi, \\ \text{sgn}(g_j - f_j) & \text{if } |g_j - f_j| > \pi. \end{cases}$$



### 3 Inpainting Models for Cyclic Data

Given an image domain  $\Omega_0 = \{1, \dots, N\} \times \{1, \dots, M\}$ , the inpainting region  $\Omega \subset \Omega_0$  is the subset where the pixel values  $f_{i,j}$ ,  $(i, j) \in \Omega$  are unknown. The (noiseless) inpainting problem consists of finding a function  $x$  on  $\Omega_0$  from data  $f$  given on  $\bar{\Omega} = \Omega_0 \setminus \Omega$  such that  $x$  is a suitable extension of  $f$  to  $\Omega_0$ . Let  $d_2(x) := d(x; b_2)$  and  $d_{1,1}(x) := d(x; b_{1,1})$ . Our functional for inpainting of noiseless cyclic data reads

$$\begin{aligned} \arg \min_{x \in [-\pi, \pi]^{N, M}} \quad & \alpha \text{TV}_1^\Omega(x) + \beta \text{TV}_2^\Omega(x) + \gamma \text{TV}_{1,1}^\Omega(x), \\ \text{s.t.} \quad & x_{i,j} = f_{i,j} \quad \text{for all } (i, j) \in \bar{\Omega}, \end{aligned} \quad (3)$$

where  $\alpha := (\alpha_1, \alpha_2, \alpha_2, \alpha_4)$ ,  $\beta := (\beta_1, \beta_2)$  and the restricted first and second order difference terms given by

$$\begin{aligned} \alpha \text{TV}_1^\Omega(x) = & \alpha_1 \sum_{(i,j)} d(x_{i,j}, x_{i+1,j}) + \alpha_2 \sum_{(i,j)} d(x_{i,j}, x_{i,j+1}) \\ & + \frac{1}{\sqrt{2}} \left( \alpha_3 \sum_{(i,j)} d(x_{i,j}, x_{i+1,j+1}) + \alpha_4 \sum_{(i,j)} d(x_{i,j+1}, x_{i+1,j}) \right), \end{aligned}$$

$$\beta \text{TV}_2^\Omega(x) = \beta_1 \sum_{(i,j)} d_2(x_{i-1,j}, x_{i,j}, x_{i+1,j}) + \beta_2 \sum_{(i,j)} d_2(x_{i,j-1}, x_{i,j}, x_{i,j+1}),$$

and

$$\gamma \text{TV}_{1,1}^\Omega(x) = \gamma \sum_{(i,j)} d_{1,1}(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1}),$$

where the sums are taken only for those  $(i, j)$  for which at least one entry  $x_{a,b}$  in the corresponding differences is contained in  $\Omega$ . We use the notation TV since the model of the first order differences resembles an anisotropic TV model.

For the inpainting problem in the presence of noise the requirement of equality on  $\bar{\Omega}$  is replaced by  $x$  being an approximation of  $f$ :

$$\arg \min_{x \in [-\pi, \pi]^{N, M}} F_{\bar{\Omega}}(x; f) + \alpha \text{TV}_1(x) + \beta \text{TV}_2(x) + \gamma \text{TV}_{1,1}(x), \quad (4)$$

where

$$F_{\bar{\Omega}}(x; f) := \sum_{(i,j) \in \bar{\Omega}} d(x_{i,j}, f_{i,j})^2.$$

and the first and second order difference terms sum over all indices in  $\Omega_0$  now.

*Initialization of the inpainting region.* Since the inpainting problem does not possess a unique minimizer the initialization of the inpainting area is crucial. We present a method which is related to the idea of unknown boundary conditions used by Almeida and Figueiredo in [1]. It can also be viewed as an implicit version of the ordering method of pixels by adapted distance functions used by März in [36,37]. To this end, we initialize  $x_{i,j} = f_{i,j}$  for  $(i, j) \in \bar{\Omega}$ . The other ones are considered as not initialized. We use first, second and mixed order differences  $d = d_1, d_2$  and  $d_{1,1}$  and let  $t \in \{1, 2, (1, 1)\}$ . Let  $x := (x_{k_1}, \dots, x_{k_l})^T$  be a set of points corresponding to a stencil of such a difference term  $d_t$ . If  $k_i \in \Omega$ ,  $i \in \{1, \dots, l\}$ , is the unique index such that  $x_{k_i}$  is not yet initialized, i.e., there is exactly one unknown point at  $k_i \in \Omega$  in  $x$ , we can initialize this value as follows. The minimal value for the absolute cyclic finite difference is  $0 = (\langle x, b_t \rangle)_{2\pi}$  and this equation provides an initial value for  $x_{k_i}$ . Such a situation of exactly one unknown index  $k_i$  always exists at the boundary of the initialized area.

## 4 Cyclic Proximal Point Algorithm

Since the proximal mappings of our absolute cyclic differences can be efficiently computed using their analytical expressions in Theorem 1 and Theorem 2, we suggest to apply a cyclic proximal point algorithm to find a minimizer for the inpainting problem. Recently, the proximal point algorithm (PPA) on the Euclidean space [44] was extended to Riemannian manifolds of non-positive sectional curvature [20] and also to Hadamard spaces [2]. A cyclic PPA (CPPA) on the Euclidean space was given in [7,8] and on Hadamard spaces in [3]. Unfortunately, one of the simplest manifolds that is not of Hadamard type is the circle  $\mathbb{S}^1$ . However, under certain assumptions we were able to prove the convergence of the CPPA to a minimizer of the denoising problem for cyclic data, see [5]. A similar proof can also be given for the inpainting problem. Indeed, we have observed convergence of our algorithm in all numerical tests.

In the CPPA the original function  $J$  is split into a sum  $J = \sum_{l=1}^c J_l$  and the proximal mappings of the functions  $J_l$  are applied in each iteration cycle, i.e.,

$$x^{(k+1)} = \text{prox}_{\lambda_k J_c} \left( \text{prox}_{\lambda_k J_{c-1}} \left( \dots \left( \text{prox}_{\lambda_k J_1} (x^{(k)}) \right) \right) \right).$$

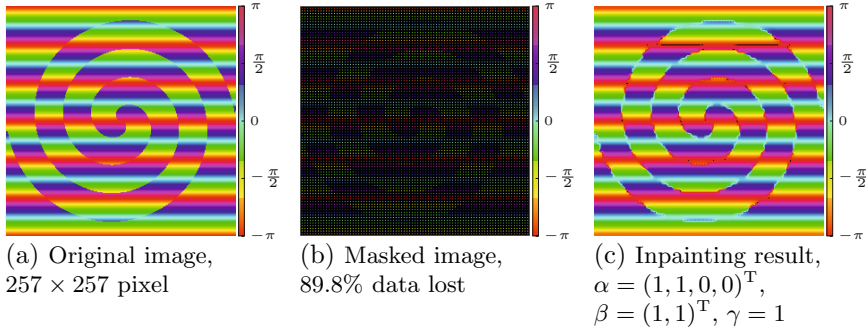
For  $J = J_1 + J_2$ , where  $J_1, J_2 : \mathbb{R}^N \rightarrow (-\infty, +\infty]$  are proper, closed convex functions, it is well known that the nested PPA

$$x^{(k+1)} = \text{prox}_{\lambda J_2} \left( \text{prox}_{\lambda J_1} (x^{(k)}) \right)$$

converges for any fixed parameter  $\lambda > 0$  to a fixed point of  $\text{prox}_{\lambda J_2} \circ \text{prox}_{\lambda J_1}$ . Unfortunately this fixed point is not a minimizer of  $J$  but of  $J_2 + \lambda J_1$ , where  $\lambda J_1$  denotes the Moreau envelope of  $J_1$ . Convergence to the correct minimizer can be achieved by choosing an iteration dependent sequence  $\{\lambda_k\}_k$  fulfilling

$$\sum_{k=0}^{\infty} \lambda_k = \infty, \quad \text{and} \quad \sum_{k=0}^{\infty} \lambda_k^2 < \infty,$$

see [3,8]. A specific splitting of our inpainting model (3) for the CPPA is given in the appendix.



**Fig. 2.** The synthetic SAR data in (a) is reduced by a factor of nine by removing two thirds of all rows and columns. They are indicated as black pixels in (b), the right image (c) shows the reconstruction based on first and second order cyclic differences.

## 5 Numerical Results

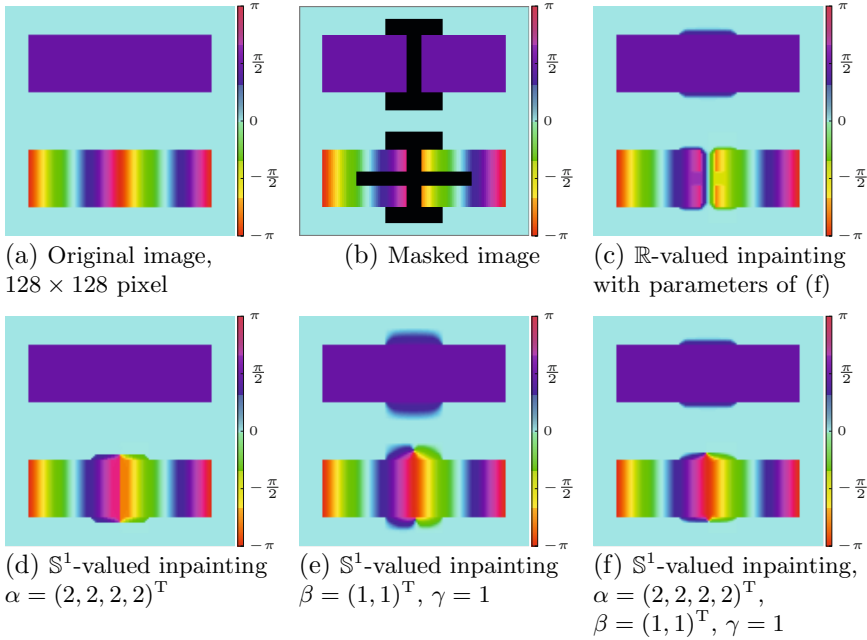
For the numerical computations of the following examples, the presented algorithms were implemented in MATLAB. The experiments were performed on a MacBook Pro with an Intel Quad Core i5, 2.6 Ghz and 8 GB of RAM on OS X 10.9.2.

*Interpolation and Approximation.* As a first example we consider an synthetic SAR data sample taken from [24]<sup>1</sup>, see Fig. 2 (a). We destroy about 89.8% of the data by removing all but the rows and columns that are not divisible by 3, see Fig. 2 (b). This is taken as input for the CPPA in order to minimize (3) using the parameters  $\alpha = (1, 1, 0, 0)^T$ ,  $\beta = (1, 1)^T$  and  $\gamma = 1$ . The result is shown in Fig. 2 (c), where the linear parts are reconstructed perfectly, while the edges are interpolated and hence suffer from linearization of the original circular edge path. The runtime is about 80 seconds for the image of size  $257 \times 257$  pixel when using  $k = 700$  iterations as a stopping criterion for the CPPA from Sec. 4.

*Inpainting for Restoring Image Regions.* A main application of inpainting is to restore destroyed image regions in noiseless images. We use the first model (3) and consider an example adapted from [40], where a similar image was used to demonstrate regularization with a second order model for real valued images. We extend their experiment by including a region with linear increase that is wrapped twice, cf. Fig. 3 (a). We remove a vertical strip in the middle of both regions and stripes between the fore- and background. Furthermore for the second, linearly increasing region we mask a small band in the middle, cf. Fig. 3 (b).

<sup>1</sup> Online available at

[ftp://ftp.wiley.com/public/sci\\_tech\\_med/phase\\_unwrapping/data.zip](ftp://ftp.wiley.com/public/sci_tech_med/phase_unwrapping/data.zip).

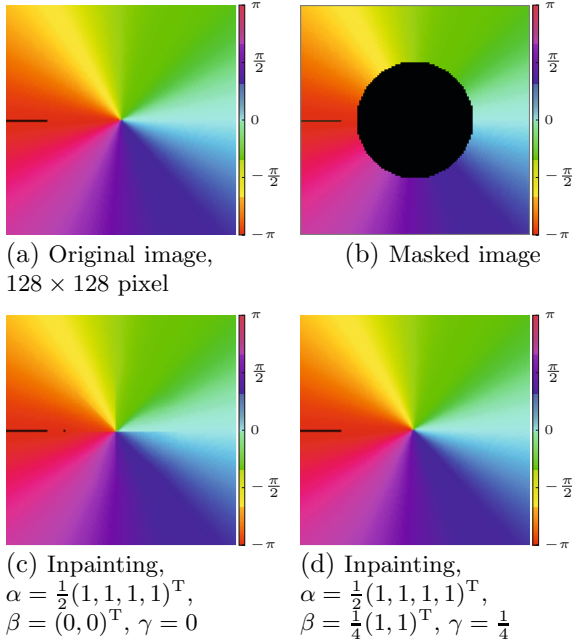


**Fig. 3.** Inpainting with first and second order differences: (a) from the original image (b) some parts (black) are lost. (c) A real-valued inpainting fails; (d) a first order model reconstructs the constant region perfectly; (e) a pure second order model has linear artifacts; (f) a first and second order model performs best.

We then employ a real valued inpainting using first and second order differences, cf. Fig. 3(c). The constant rectangle shows a similar behavior to [40], where the smoothing at the top and bottom is reduced here. This is due to employment of both first and second order real valued differences. Most noticeably, the linearly increasing region is not reconstructed.

The Figs. 3(d)–(f) illustrate the effects of first and second order absolute cyclic differences. Fig. 3(d) uses only the first order model, (e) only the second order differences. and (f) combines both. The first order absolute cyclic differences reconstruct the constant region perfectly, but also produce the well known staircasing in the lower part. The second order cyclic model introduces a smooth transition between fore- and background. However, it perfectly reconstructs the linear increase. Combining both the first and second order cyclic models yields a perfect reconstruction of the linearly increasing region while reducing the smooth transition, cf. Fig. 3(f).

As a second reconstruction example we consider the function  $\text{atan2}(y, x)$  sampled on a regular grid in  $[-\frac{1}{2}, \frac{1}{2}]^2$  having 128 sampling points in each dimension,



**Fig. 4.** We mask a circular region at the center of (a), see (b). The reconstruction used in (c) employs only first order cyclic differences and produces staircasing. Combining first and second order cyclic differences in (d) we obtain a nearly perfect reconstruction.

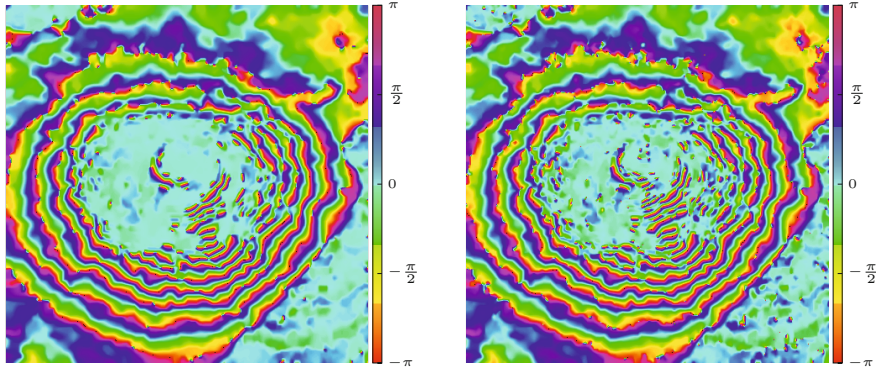
cf. Fig. 4 (a). We take a circular mask in the center of the image, see Figure 4 (b), where the mask is shown in black. In this experiment we compare the results using only first order absolute cyclic differences with a combined approach of first and second order cyclic model.

When only using first order differences, we obtain a result that again reveals staircasing, cf. Fig. 4 (c). It prefers  $x$ - and  $y$ -axis, and both diagonals, which can be seen by the crosses created in the middle. By also including second order differences we obtain almost the original image, cf. Fig. 4 (d).

For both examples the computation takes about 43 seconds for first order differences and 55 seconds for the combined cases, respectively. We used  $k = 2000$  iterations as a stopping criterion for the CPPA from Sec. 4.

*Inpainting in the presence of noise.* In real world measurements data are often noisy. If these data are also partially lost, we employ the model (4). As an example we consider the measurement of elevation using InSAR [11,39]. In particular, we consider phase valued measured data of Mount Vesuvius [43]<sup>2</sup>.

<sup>2</sup> Online available at <https://earth.esa.int/workshops/ers97/program-details/speeches/rocca-et-al/>



(a) Denoised Mt. Vesuvius image,  
432 × 426 pixels

(b) Inpainted and denoised version,  
where 20% of data was lost

**Fig. 5.** Real data of Mount Vesuvius. We compare a pure denoising approach in (a) with a combined inpainting and denoising approach in (b), where 20% of the data was lost before the inpainting and denoising process.

We compare denoising with simultaneously inpainting and denoising. To this end, we randomly destroyed 20% of the data items. The results without and with lost data are shown in Fig. 5 (a) and (b), respectively. For the inpainting version the parameters used in Fig. 5 (a),  $\alpha = \frac{1}{4}(1, 1, 1, 1)^T$ ,  $\beta = \frac{3}{4}(1, 1)^T$  and  $\gamma = \frac{3}{4}$ , were multiplied by 2. The combined approach of simultaneously inpainting and denoising introduces a few more artifacts than pure denoising; cf. the middle and top right area. However, both results are of comparable quality in smooth regions, e.g., the plateau in the bottom left.

## 6 Conclusions

We proposed an inpainting model for cyclic data which involves our recently established second order cyclic differences. Since there are analytical expressions for the proximal mappings of these differences we suggested a CPP algorithm together with a strategy for choosing the cycles to compute a minimizer of the corresponding functionals efficiently. There is large room for improvements and future work.

We want to apply our second order cyclic differences to other image restoration tasks such as, e.g., deblurring and investigate other couplings of first and second order differences. It is possible to generalize our geometrically driven definition of second order differences to higher dimensional spheres and also to general manifolds. We want to use such generalization for image processing tasks of general manifold-valued data.

## Appendix

The proximal mappings from Theorem 1 can be efficiently applied in parallel if they act on distinct data. This reduces the cycle length  $c$  of the CPPA from Sec. 4 tremendously and provides an efficient, parallel implementation. Especially, the cycle length is independent of the inpainting area  $\Omega$  or the image domain  $\Omega_0$  and depends only on the number of dissimilar differences used. We present a specific splitting in the CPPA of our inpainting model (3) given by

$$J(x) = \alpha \text{TV}_1^\Omega(x) + \beta \text{TV}_2^\Omega(x) + \gamma \text{TV}_{1,1}^\Omega(x)$$

with the constraints  $x_{i,j} = f_{i,j}$  on  $\bar{\Omega}$ . We write

$$J = \sum_{l=1}^{18} J_l$$

with summands  $J_l$  given by the subsequent explanation. We start with the  $\alpha \text{TV}_1^\Omega(x)$  term and first consider the horizontal summand  $\alpha_1 \sum_{(i,j)} d(x_{i,j}, x_{i+1,j})$ . We split this sum into an even and an odd part  $J_1$  and  $J_2$ , more precisely

$$\alpha_1 \sum_{(i,j)} d(x_{i,j}, x_{i+1,j}) = J_1 + J_2,$$

where

$$J_1 + J_2 := \alpha_1 \sum_{(i,j)} d(x_{2i,j}, x_{2i+1,j}) + \alpha_1 \sum_{(i,j)} d(x_{2i+1,j}, x_{2i+2,j}),$$

with the restriction to the summands as in Sec. 3. This means, that for each item  $(i, j)$  in the sum, the corresponding index of at least one of the arguments  $x_{2i,j}$ ,  $x_{2i+1,j}$  is in  $\Omega$ . For the vertical as well as for the diagonal summands in  $\alpha \text{TV}_1^\Omega(x)$  we proceed analogously to obtain the splitting functionals  $J_3, \dots, J_8$ .

Next, we consider the  $\beta \text{TV}_2^\Omega(x)$  term with its first (horizontal) summand given by  $\beta_1 \sum_{(i,j)} d_2(x_{i-1,j}, x_{i,j}, x_{i+1,j})$ . We decompose this summand into three sums  $J_9, J_{10}, J_{11}$  given by

$$\begin{aligned} J_9 &= \beta_1 \sum_{(i,j)} d_2(x_{3i-1,j}, x_{3i,j}, x_{3i+1,j}), \\ J_{10} &= \beta_1 \sum_{(i,j)} d_2(x_{3i,j}, x_{3i+1,j}, x_{3i+2,j}), \\ J_{11} &= \beta_1 \sum_{(i,j)} d_2(x_{3i+1,j}, x_{3i+2,j}, x_{3i+3,j}), \end{aligned}$$

again, with the restriction to the summands as in Sec. 3. For the vertical summand in  $\beta \text{TV}_2^\Omega(x)$  we proceed analogously to obtain  $J_{12}, \dots, J_{14}$ .

It remains to split the term  $\gamma \text{TV}_{1,1}^\Omega(x)$  into four functionals  $J_{15}, \dots, J_{18}$  as follows

$$J_{15} = \gamma \sum_{(i,j)} d_{1,1}(x_{2i,2j}, x_{2i+1,2j}, x_{2i,2j+1}, x_{2i+1,2j+1}),$$

$$J_{16} = \gamma \sum_{(i,j)} d_{1,1}(x_{2i+1,2j}, x_{2i+2,2j}, x_{2i+1,2j+1}, x_{2i+2,2j+1}),$$

$$J_{17} = \gamma \sum_{(i,j)} d_{1,1}(x_{2i,2j+1}, x_{2i+1,2j+1}, x_{2i,2j+2}, x_{2i+1,2j+2}),$$

$$J_{18} = \gamma \sum_{(i,j)} d_{1,1}(x_{2i+1,2j+1}, x_{2i+2,2j+1}, x_{2i+1,2j+2}, x_{2i+2,2j+2}).$$

Each summation is again restricted to those terms where at least one index of an argument of  $d_{1,1}$  is in  $\Omega$ . For  $J_1, \dots, J_{18}$  the corresponding proximal mapping can be explicitly computed and the cycle length  $c = 18$  is independent of the cardinality of  $\Omega$  or  $\Omega_0$ . After application of the proximal mapping of each  $J_i$  we set  $x_{i,j} = f_{i,j}$  on  $\hat{\Omega}$  to fulfill the respective constraint, which is the same as performing a projection.

## References

1. Almeida, M., Figueiredo, M.: Deconvolving images with unknown boundaries using the alternating direction method of multipliers. *IEEE Trans. on Image Process.* 22(8), 3074–3086 (2013)
2. Bačák, M.: The proximal point algorithm in metric spaces. *Isr. J. Math.* 194(2), 689–701 (2013)
3. Bačák, M.: Computing medians and means in Hadamard spaces. *SIAM J. Optim.* (to appear, 2014)
4. Ballester, C., Bertalmio, 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)
5. Bergmann, R., Laus, F., Steidl, G., Weinmann, A.: Second order differences of cyclic data and applications in variational denoising (Preprint, 2014)
6. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proceedings of SIGGRAPH, New Orleans, USA*, pp. 417–424 (2000)
7. Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. Technical Report LIDS-P-2848, Laboratory for Information and Decision Systems, MIT, Cambridge, MA (2010)
8. Bertsekas, D.P.: Incremental proximal methods for large scale convex optimization. *Math. Program., Ser. B* 129(2), 163–195 (2011)
9. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* 3(3), 1–42 (2009)
10. Bugeau, A., Bertalmío, M., Caselles, V., Sapiro, G.: A comprehensive framework for image inpainting. *IEEE Trans. Signal Process.* 19, 2634–2645 (2010)
11. Bürgmann, R., Rosen, P.A., Fielding, E.J.: Synthetic aperture radar interferometry to measure earth’s surface topography and its deformation. *Annu. Rev. Earth Planet. Sci.* 28(1), 169–209 (2000)



12. Cai, J.-F., Dong, B., Osher, S., Shen, Z.: Image restoration: Total variation, wavelet frames, and beyond. *J. Amer. Math. Soc.* 25(4), 1033–1089 (2012)
13. Caselles, V., Morel, J.-M., Sbert, C.: An axiomatic approach to image interpolation. *IEEE Trans. on Image Process.* 7(3), 376–386 (1998)
14. Chambolle, A., Lions, P.-L.: Image recovery via total variation minimization and related problems. *Numer. Math.* 76(2), 167–188 (1997)
15. Chan, T., Shen, J.: Local inpainting models and TV inpainting. *SIAM J. Appl. Math.* 62(3), 1019–1043 (2001)
16. Chan, T.F., Marquina, A., Mulet, P.: High-order total variation-based image restoration. *SIAM J. Sci. Comput.* 22(2), 503–516 (2000)
17. Chan, T.F., Shen, J.: *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods.* SIAM (2005)
18. Chedf'Hotel, C., Tschumperlé, D., Deriche, R., Faugeras, O.: Regularizing flows for constrained matrix-valued images. *J. Math. Imaging Vis.* 20(1-2), 147–162 (2004)
19. Didas, S., Weickert, J., Burgeth, B.: Properties of higher order nonlinear diffusion filtering. *J. Math. Imaging Vis.* 35, 208–226 (2009)
20. Ferreira, O.P., Oliveira, P.R.: Proximal point algorithm on Riemannian manifolds. *Optimization* 51(2), 257–270 (2002)
21. Fisher, N.I.: *Statistical Analysis of Circular Data.* Cambridge University Press (1995)
22. Fletcher, P.: Geodesic regression and the theory of least squares on Riemannian manifolds. *Int. J. Comput. Vision* 105(2), 171–185 (2013)
23. Fletcher, P., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Process.* 87(2), 250–262 (2007)
24. Ghiglia, D.C., Pritt, M.D.: *Two-dimensional phase unwrapping: theory, algorithms, and software.* Wiley (1998)
25. Giaquinta, M., Modica, G., Souček, J.: Variational problems for maps of bounded variation with values in  $S^1$ . *Calc. Var.* 1(1), 87–121 (1993)
26. Giaquinta, M., Mucci, D.: The BV-energy of maps into a manifold: relaxation and density results. *Ann. Sc. Norm. Super. Pisa Cl. Sci.* 5(4), 483–548 (2006)
27. Giaquinta, M., Mucci, D.: Maps of bounded variation with values into a manifold: total variation and relaxed energy. *Pure Appl. Math. Q.* 3(2), 513–538 (2007)
28. Grohs, P., Hardering, H., Sander, O.: Optimal a priori discretization error bounds for geodesic finite elements. Technical Report 2013-16, Seminar for Applied Mathematics, ETH Zürich, Switzerland (2013)
29. Grohs, P., Wallner, J.: Interpolatory wavelets for manifold-valued data. *Appl. Comput. Harmon. Anal.* 27(3), 325–333 (2009)
30. Guillemot, C., Le Meur, O.: Image inpainting: Overview and recent advances. *IEEE Signal Process. Mag.* 31(1), 127–144 (2014)
31. Hinterberger, W., Scherzer, O.: Variational methods on the space of functions of bounded Hessian for convexification and denoising. *Computing* 76(1), 109–133 (2006)
32. Jammalamadaka, S.R., SenGupta, A.: *Topics in Circular Statistics.* World Scientific Publishing Company (2001)
33. Lefkimmatis, S., Bourquard, A., Unser, M.: Hessian-based norm regularization for image restoration with biomedical applications. *IEEE Trans. Image Process.* 21(3), 983–995 (2012)
34. Lellmann, J., Strelakovsky, E., Koetter, S., Cremers, D.: Total variation regularization for functions with values in a manifold. In: *IEEE ICCV 2013*, pp. 2944–2951 (2013)

35. Lysaker, M., Lundervold, A., Tai, X.-C.: Noise removal using fourth-order partial differential equations with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Process.* 12(12), 1579–1590 (2003)
36. März, T.: Image inpainting based on coherence transport with adapted distance functions. *SIAM J. Imaging Sci.* 4(4), 981–1000 (2011)
37. März, T.: A well-posedness framework for inpainting based on coherence transport. *Found. Comput. Math.* (to appear, 2014)
38. Masnou, S., Morel, J.-M.: Level lines based disocclusion. In: *IEEE ICIP 1998*, pp. 259–263 (1998)
39. Massonnet, D., Feigl, K.L.: Radar interferometry and its application to changes in the Earth's surface. *Rev. Geophys.* 36(4), 441–500 (1998)
40. Papafitsoros, K., Schönlieb, C.B.: A combined first and second order variational approach for image reconstruction. *J. Math. Imaging Vis.* 2(48), 308–338 (2014)
41. Pennec, X.: Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *J. Math. Imaging Vis.* 25(1), 127–154 (2006)
42. Rahman, I.U., Drori, I., Stodden, V.C., Donoho, D.L.: Multiscale representations for manifold-valued data. *Multiscale Model. Simul.* 4(4), 1201–1232 (2005)
43. Rocca, F., Prati, C., Guarnieri, A.M.: Possibilities and limits of SAR interferometry. In: *Proc. Int. Conf. Image Process. Techn.*, pp. 15–26 (1997)
44. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* 14(5), 877–898 (1976)
45. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* 60(1), 259–268 (1992)
46. Scherzer, O.: Denoising with higher order derivatives of bounded variation and an application to parameter estimation. *Computing* 60, 1–27 (1998)
47. Setzer, S., Steidl, G.: Variational methods with higher order derivatives in image processing. In: *Approximation Theory XII: San Antonio 2007*, pp. 360–385 (2008)
48. Setzer, S., Steidl, G., Teuber, T.: Infimal convolution regularizations with discrete  $l_1$ -type functionals. *Commun. Math. Sci.* 9(3), 797–872 (2011)
49. Strelakovsky, E., Cremers, D.: Total variation for cyclic structures: Convex relaxation and efficient minimization. In: *IEEE CVPR 2011*, pp. 1905–1911 (2011)
50. Strelakovsky, E., Cremers, D.: Total cyclic variation and generalizations. *J. Math. Imaging Vis.* 47(3), 258–277 (2013)
51. Valkonen, T., Bredies, K., Knoll, F.: Total generalized variation in diffusion tensor imaging. *SIAM J. Imag. Sci.* 6(1), 487–525 (2013)
52. Weinmann, A.: Interpolatory multiscale representation for functions between manifolds. *SIAM J. Math. Anal.* 44(1), 162–191 (2012)
53. Weinmann, A., Demaret, L., Storath, M.: Total variation regularization for manifold-valued data (2013) (preprint)

# Discrete Green's Functions for Harmonic and Biharmonic Inpainting with Sparse Atoms

Sebastian Hoffmann<sup>1</sup>, Gerlind Plonka<sup>2</sup>, and Joachim Weickert<sup>1</sup>

<sup>1</sup> Mathematical Image Analysis Group  
Faculty of Mathematics and Computer Science, Campus E1.7  
Saarland University, 66041 Saarbrücken, Germany  
{hoffmann,weickert}@mia.uni-saarland.de

<sup>2</sup> Institute for Numerical and Applied Mathematics  
University of Göttingen  
Lotzestr. 16–18, 37083 Göttingen, Germany  
plonka@math.uni-goettingen.de

**Abstract.** Recent research has shown that inpainting with the Laplace or biharmonic operator has a high potential for image compression, if the stored data is optimised and sufficiently sparse. The goal of our paper is to connect these linear inpainting methods to sparsity concepts. To understand these relations, we explore the theory of Green's functions. In contrast to most work in the mathematical literature, we derive our Green's functions in a discrete setting and on a rectangular image domain with homogeneous Neumann boundary conditions. These discrete Green's functions can be interpreted as columns of the Moore–Penrose inverse of the discretised differential operator. More importantly, they serve as atoms in a dictionary that allows a sparse representation of the inpainting solution. Apart from offering novel theoretical insights, this representation is also simple to implement and computationally efficient if the inpainting data is sparse.

**Keywords:** inpainting, sparsity, discrete Green's functions, Laplace operator, biharmonic operator.

## 1 Introduction

Image inpainting with partial differential equations (PDEs) is becoming increasingly important for image compression. For this problem, nonlinear anisotropic diffusion processes have been introduced by Galić et al. in 2005 [7] and have been improved later in [8]. In the meantime, a more sophisticated variant is able to outperform JPEG2000 [19]. Even with a conceptually simpler linear process based e.g. on the Laplace equation, one can achieve remarkable results [15,12,17] and beat the quality of state-of-the-art methods for specific types of images [14,9,13]. Also the biharmonic equation has been reported to yield very good results [8,3].

In the present paper, we want to gain theoretical insights on inpainting methods with linear selfadjoint differential operators such as the Laplacian or the

biharmonic operator. In particular, we analyse their relation to a very popular idea in modern signal and image analysis, namely sparsity. To this end, we make use of the concept of discrete Green's functions [1,4]. Green's functions are mainly known from the continuous theory of partial differential equations (PDEs) as a tool to describe the solution of boundary value problems [16]. Most publications on Green's functions focus on continuous differential operators. Digital images, however, reveal a natural discretisation on a regular grid. Moreover, they are given on a rectangular image domain, and it is fairly common to extend image processing operators at the boundaries by mirroring. This motivates us to investigate discrete Green's functions for linear differential operators on a rectangular image domain with homogeneous Neumann boundary conditions. Moreover, we will give an interpretation of the obtained discrete Green's functions in terms of linear algebra. More precisely, we will elaborate the connection to the Moore–Penrose inverse of the discretised differential operator.

The discrete Green's functions that we derive will serve as atoms in a dictionary for inpainting. There is a one-to-one correspondence between each pixel and its corresponding Green's function. Hence, if only a sparse set of pixels is kept, the solution of the discrete inpainting problem can be expressed in a compact way in terms of their Green's functions. We will show that this representation does not only offer novel theoretical insights into the connections between inpainting and sparsity, but also has algorithmic benefits. The main focus of the present paper, however, will be on the theoretical aspect.

The outline of our paper is as follows. First we sketch the continuous and discrete formulations of inpainting with the Laplace and biharmonic equation in Section 2. In the subsequent section we explain the concept of discrete Green's functions and their use for a sparse representation of the solution of the inpainting problems. Numerical advantages of our Green's function framework are discussed in Section 4. Our paper is concluded with a summary in Section 5.

## 2 Laplace and Biharmonic Inpainting

### 2.1 Continuous Inpainting Models

Let  $\Omega \subset \mathbb{R}^2$  denote a rectangular image domain and  $f : \Omega \rightarrow \mathbb{R}$  a greyscale image. If this image is only known at some subset  $\Omega_K \subset \Omega$ , one can try to fill in the missing information by solving the Laplace equation

$$-\Delta u = 0 \quad \text{on } \Omega \setminus \Omega_K \quad (1)$$

with homogeneous Neumann boundary conditions:

$$\partial_n u = 0 \quad \text{on } \partial\Omega, \quad (2)$$

where  $\partial_n$  denotes the derivative normal to the boundaries. Moreover, the known data set provides Dirichlet boundary conditions:

$$u = f \quad \text{on } \Omega_K. \quad (3)$$

As an alternative to the Laplace equation, one can also consider higher-order differential operators leading e.g. to the biharmonic equation:

$$\Delta^2 u = 0 \quad \text{on } \Omega \setminus \Omega_K. \quad (4)$$

Both models have in common that they use linear selfadjoint differential operators. These properties will be useful for our later analysis. From a practical viewpoint they are attractive, since they are parameter-free and give rise to relatively easy implementations.

## 2.2 Discrete Inpainting Models

Digital images reveal a discretisation on an equispaced rectangular grid. Thus, it is natural to use finite difference discretisations of the beforementioned continuous inpainting processes. We consider a regular two-dimensional grid  $\Gamma = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$  with grid size  $h$ . The value of a discrete image  $\mathbf{f}$  at a grid point  $(i, j) \in \Gamma$  is denoted by  $f_{i,j}$ . The subset  $K \subset \Gamma$  denotes the grid points where the discrete inpainting data is known. We call them mask points. At the locations  $\Gamma \setminus K$  where the data is unknown, we seek the inpainting solution  $\mathbf{u}$  by solving a discrete problem of type

$$(\mathbf{D}\mathbf{u})_{i,j} = 0 \quad \text{for } (i, j) \in \Gamma \setminus K, \quad (5)$$

$$u_{i,j} = f_{i,j} \quad \text{for } (i, j) \in K. \quad (6)$$

Here,  $\mathbf{D}$  can be seen as an inpainting operator. We mainly focus on the following two choices. On the one hand, we consider  $\mathbf{D} = -\mathbf{L}$ , where  $\mathbf{L}$  is the discrete Laplace operator (harmonic operator) on  $\Gamma$  fulfilling homogeneous Neumann boundary conditions at the image boundaries. For the inner grid points its stencil notation is given by

$$\frac{1}{h^2} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}.$$

The homogeneous Neumann boundary conditions are incorporated by mirroring the image at the boundaries and by using the above stencil also for the boundary grid points. The resulting inpainting process is also known as homogeneous diffusion inpainting. In [14], the existence and uniqueness of the discrete inpainting solution for the Laplace operator has been shown. On the other hand, we will also consider the biharmonic operator, i.e.  $\mathbf{D} = \mathbf{B} := \mathbf{L}^2$ .

Typically, the inpainting solution is found by solving the discrete problem directly. This can be done with iterative methods such as a fast explicit diffusion (FED) scheme [11] or bidirectional multigrid approaches [14]. In the present paper we want to study how the solution can be obtained in a noniterative way by means of discrete Green's functions.

### 2.3 Eigenvalues and Eigenvectors of the Discrete Operators

For our later analysis it is useful to represent the discrete differential operators  $-\mathbf{L}$  and  $\mathbf{B}$  in terms of their eigenvalues and eigenvectors. The following theorem provides the required information. It extends 1D results that can be found for example in [20] to the two-dimensional setting.

**Theorem 1 (Eigenvalues and Eigenvectors of the Discrete Operators).**

*The orthonormal set of eigenvectors of  $-\mathbf{L}$  as well as of  $\mathbf{B}$  is given by*

$$(\mathbf{v}_{m,n})_{i,j} = \begin{cases} \sqrt{\frac{1}{MN}} & \text{if } m = n = 0, \\ \sqrt{\frac{2}{MN}} \cdot \cos(\mu\tilde{i}) \cdot \cos(\nu\tilde{j}) & \text{if either } m = 0 \text{ or } n = 0, \\ \sqrt{\frac{4}{MN}} \cdot \cos(\mu\tilde{i}) \cdot \cos(\nu\tilde{j}) & \text{if } m > 0 \text{ and } n > 0, \end{cases} \quad (7)$$

with  $(m, n) \in \Gamma$ ,  $\mu := \frac{m\pi}{M}$ ,  $\nu := \frac{n\pi}{N}$ ,  $\tilde{i} := (i + \frac{1}{2})$ , and  $\tilde{j} := (j + \frac{1}{2})$ .

*The corresponding eigenvalues for  $-\mathbf{L}$  are*

$$\lambda_{m,n}^{-L} = \frac{4}{h^2} \left( \sin^2\left(\frac{\mu}{2}\right) + \sin^2\left(\frac{\nu}{2}\right) \right). \quad (8)$$

*The eigenvalues of the discrete biharmonic operator  $\mathbf{B}$  read as*

$$\lambda_{m,n}^B = (\lambda_{m,n}^{-L})^2 \quad (9)$$

*Proof.* While this eigenstructure may not appear obvious, proving its correctness is fairly straightforward: One has to check that  $-\mathbf{L}\mathbf{v}_{m,n} = \lambda_{m,n}^{-L}\mathbf{v}_{m,n}$  and  $\mathbf{B}\mathbf{v}_{m,n} = \lambda_{m,n}^B\mathbf{v}_{m,n}$  hold true for all  $(m, n) \in \Gamma$  and that the homogeneous Neumann boundary conditions are fulfilled. Additionally, one has to show the orthonormality of the set of eigenvectors.  $\square$

We observe that both operators are singular, since the eigenvalues  $\lambda_{0,0}^{-L}$  and  $\lambda_{0,0}^B$  vanish. This will complicate some of our discussions on discrete Green's functions in the next section.

## 3 Discrete Green's Functions

After the preceding discussions we are in a position to introduce the concept of discrete Green's functions. First, we discuss the basic structure before we sketch relations to linear algebra and specific applications to our inpainting problem.

### 3.1 Basic Structure

Let us study a general discrete problem of the following type:

$$\mathbf{D}\mathbf{u} = \mathbf{a}. \quad (10)$$

Thereby  $\mathbf{u} \in \mathbb{R}^{M \times N}$  is the unknown image,  $\mathbf{a} \in \mathbb{R}^{M \times N}$  is a prescribed right hand side, and  $\mathbf{D} \in \mathbb{R}^{(M \times N) \times (M \times N)}$  a given symmetric discrete linear differential operator incorporating homogeneous Neumann boundary conditions.

The solvability of this problem can be investigated with the so called Fredholm alternative, which is known from the theory of differential equations; see e.g. [5]:

**Theorem 2 (Fredholm Alternative).** *If  $\mathbf{D}$  is invertible, then the solution  $\mathbf{u}$  of the discrete problem (10) exists and is unique. Otherwise, assuming that  $\mathbf{D}^\top$  possesses the single eigenvalue 0 with the corresponding eigenvector  $\mathbf{v} \in \mathbb{R}^{M \times N}$ , there exist infinitely many solutions if*

$$\langle \mathbf{v}, \mathbf{a} \rangle = 0, \tag{11}$$

and there exists no solution at all if

$$\langle \mathbf{v}, \mathbf{a} \rangle \neq 0. \tag{12}$$

Here, the Euclidean inner product is defined as  $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{(i,j) \in \Gamma} a_{i,j} b_{i,j}$ . Let us assume that  $\mathbf{a}$  in (10) is chosen such that there exists a solution  $\mathbf{u}$ . A standard approach to find this solution is to solve the linear system of equations directly. Instead, another promising approach is to express the solution by means of Green's functions. The Green's function can be considered as the influence of an impulse at a point  $(k, \ell)$  on the complete image. Assuming that  $\mathbf{D}$  is invertible, the discrete Green's function  $\mathbf{g}_{k,\ell}$  corresponding to a point  $(k, \ell) \in \Gamma$  for a given discrete problem is defined as the solution of

$$(\mathbf{D}\mathbf{g}_{k,\ell})_{i,j} = \delta_{(k,\ell),(i,j)} \quad \text{for } (i,j) \in \Gamma, \tag{13}$$

where the Kronecker delta function is defined as

$$\delta_{(k,\ell),(i,j)} = \begin{cases} 1 & \text{if } (i,j) = (k,\ell), \\ 0 & \text{if } (i,j) \neq (k,\ell). \end{cases} \tag{14}$$

Otherwise, if  $\mathbf{D}$  possesses the single eigenvalue 0 and if  $\mathbf{v}$  is the corresponding eigenvector of  $\mathbf{D}^\top$ , we can still obtain Green's functions by the following modification. The infinitely many discrete Green's functions for a point  $(k, \ell) \in \Gamma$  are now defined as solutions of

$$(\mathbf{D}\mathbf{g}_{k,\ell})_{i,j} = \delta_{(k,\ell),(i,j)} - \frac{v_{i,j} \cdot v_{k,\ell}}{\langle \mathbf{v}, \mathbf{v} \rangle} \quad \text{for } (i,j) \in \Gamma. \tag{15}$$

Indeed, the right hand side of (15) (in vector notation) now satisfies the solvability condition (11):

$$\left\langle \mathbf{v}, \delta_{k,\ell} - \frac{v_{k,\ell}}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v} \right\rangle = v_{k,\ell} - v_{k,\ell} = 0. \tag{16}$$

### 3.2 Interpretation as Moore–Penrose Inverse

The Fredholm alternative can also be expressed in terms of linear algebra. To this end, we reshape the image matrices  $\mathbf{u}$ ,  $\mathbf{a}$  to vectors of length  $MN$  using the operation  $\text{col} : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{MN}$ , and  $\mathbf{D}$  to a symmetric  $(MN \times MN)$ -matrix  $\mathbf{D}_{MN}$ . Then, (10) transfers to a linear system  $\mathbf{D}_{MN} \text{col}(\mathbf{u}) = \text{col}(\mathbf{a})$  of size  $MN$ . This system is uniquely solvable, if and only if  $\mathbf{D}_{MN}$  is invertible. If  $\text{rank}(\mathbf{D}_{MN}) = MN - 1$ , then (10) possesses either infinitely many solutions if  $\text{rank}(\mathbf{D}_{MN}) = \text{rank}(\mathbf{D}_{MN}, \text{col}(\mathbf{a}))$ , or no solution if  $\text{rank}(\mathbf{D}_{MN}) < \text{rank}(\mathbf{D}_{MN}, \text{col}(\mathbf{a}))$ .

Assuming that  $\mathbf{D}_{MN}$  is invertible, the discrete Green’s function defined in (13) can be expressed as the solution of

$$\mathbf{D}_{MN} \mathbf{G}_{MN} = \mathbf{I}_{MN}, \quad (17)$$

where  $\mathbf{I}_{MN}$  denotes the identity matrix of size  $MN \times MN$  and  $\mathbf{G}_{MN} \in \mathbb{R}^{MN \times MN}$  the matrix that contains the discrete Green’s functions  $\mathbf{g}_{k,\ell}$  as columns.

If  $\text{rank}(\mathbf{D}_{MN}) = MN - 1$  then there exist infinitely many Green’s functions, and (15) leads to:

$$\mathbf{D}_{MN} \mathbf{G}_{MN} = \mathbf{I}_{MN} - \frac{1}{\langle \mathbf{v}, \mathbf{v} \rangle} (\text{col}(\mathbf{v}))(\text{col}(\mathbf{v}))^\top. \quad (18)$$

In the following theorem, we introduce a useful additional constraint that creates a unique solution and allows to relate discrete Green’s functions to the Moore–Penrose inverse of their discrete differential operator. The Moore–Penrose inverse aims at generalising the inverse of a matrix such that it is also applicable to singular matrices [10].

**Theorem 3 (Discrete Green’s Functions and Moore–Penrose Inverse).**

*Let  $\text{col}(\mathbf{v})$  denote the eigenvector to the singular eigenvalue of  $\mathbf{D}_{MN}$ . If the discrete Green’s functions  $\mathbf{g}_{k,\ell}$  satisfy the additional constraint*

$$\langle \mathbf{v}, \mathbf{g}_{k,\ell} \rangle = 0 \quad \text{for all } (k, \ell) \in \Gamma, \quad (19)$$

*then they are given by the columns of the Moore–Penrose inverse of  $\mathbf{D}_{MN}$ .*

*Proof.* To verify that  $\mathbf{G}_{MN}$  is the Moore–Penrose inverse of  $\mathbf{D}_{MN}$ , we have to check the following properties (cf. [10]):

- (i)  $\mathbf{D}_{MN} \mathbf{G}_{MN} \mathbf{D}_{MN} = \mathbf{D}_{MN}$
- (ii)  $\mathbf{G}_{MN} \mathbf{D}_{MN} \mathbf{G}_{MN} = \mathbf{G}_{MN}$
- (iii)  $\mathbf{D}_{MN} \mathbf{G}_{MN}$  is symmetric.
- (iv)  $\mathbf{G}_{MN} \mathbf{D}_{MN}$  is symmetric.

Since  $\text{col}(\mathbf{v})$  is an eigenvector of  $\mathbf{D}_{MN}$  to the eigenvalue 0, we have

$$(\text{col}(\mathbf{v}))^\top \mathbf{D}_{MN} = \mathbf{0}^\top. \quad (20)$$



Thus, together with (18), it follows that

$$\mathbf{D}_{MN}\mathbf{G}_{MN}\mathbf{D}_{MN} = \left( \mathbf{I}_{MN} - \frac{1}{\langle \mathbf{v}, \mathbf{v} \rangle} (\text{col}(\mathbf{v}))(\text{col}(\mathbf{v}))^\top \right) \mathbf{D}_{MN} = \mathbf{D}_{MN} \quad (21)$$

and

$$\mathbf{G}_{MN}\mathbf{D}_{MN}\mathbf{G}_{MN} = \mathbf{G}_{MN} - \frac{1}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{G}_{MN}(\text{col}(\mathbf{v}))(\text{col}(\mathbf{v}))^\top. \quad (22)$$

The condition  $\langle \mathbf{v}, \mathbf{g}_{k,\ell} \rangle = 0$  implies  $\mathbf{G}_{MN}(\text{col}(\mathbf{v})) = \mathbf{0}$ , and hence

$$\mathbf{G}_{MN}\mathbf{D}_{MN}\mathbf{G}_{MN} = \mathbf{G}_{MN}. \quad (23)$$

From (18) it is evident that  $\mathbf{D}_{MN}\mathbf{G}_{MN}$  is symmetric. Let us now show that also  $\mathbf{G}_{MN}\mathbf{D}_{MN}$  is symmetric. Due to the symmetry of  $\mathbf{D}_{MN}$ , we can diagonalise it and write

$$\mathbf{D}_{MN} = \mathbf{V}\mathbf{S}\mathbf{V}^\top \quad (24)$$

with a diagonal matrix  $\mathbf{S}$  and an orthogonal matrix  $\mathbf{V}$ . Since  $\mathbf{D}_{MN}$  contains a singular eigenvalue, we obtain the Moore–Penrose inverse  $\mathbf{G}_{MN}$  as

$$\mathbf{G}_{MN} = \mathbf{D}_{MN}^+ = \mathbf{V}\mathbf{S}^+\mathbf{V}^\top. \quad (25)$$

The matrix  $\mathbf{S}^+$  contains the reciprocal of the eigenvalues except for the zero eigenvalue that remains 0. Furthermore, as  $\mathbf{V}$  is orthogonal, we obtain

$$\mathbf{G}_{MN}\mathbf{D}_{MN} = (\mathbf{V}\mathbf{S}^+\mathbf{V}^\top)(\mathbf{V}\mathbf{S}\mathbf{V}^\top) = \mathbf{V}\mathbf{S}^+\mathbf{S}\mathbf{V}^\top \quad (26)$$

as well as

$$(\mathbf{G}_{MN}\mathbf{D}_{MN})^\top = (\mathbf{V}\mathbf{S}\mathbf{V}^\top)^\top (\mathbf{V}\mathbf{S}^+\mathbf{V}^\top)^\top \quad (27)$$

$$= \mathbf{V}\mathbf{S}^\top (\mathbf{S}^+)^\top \mathbf{V}^\top = \mathbf{V}\mathbf{S}^+\mathbf{S}\mathbf{V}^\top \quad (28)$$

$$= \mathbf{G}_{MN}\mathbf{D}_{MN}. \quad (29)$$

Thus,  $\mathbf{G}_{MN}\mathbf{D}_{MN}$  is symmetric, too.  $\square$

### 3.3 Representing Solutions with Green's Functions

Knowing the Green's functions for all  $(k, \ell) \in \Gamma$ , the following theorem can be formulated [6]:

**Theorem 4 (Analytic Solution).** *The solution  $\mathbf{u}$  of the discrete problem (10) is given by*

$$\mathbf{u} = \sum_{(k,\ell) \in \Gamma} a_{k,\ell} \mathbf{g}_{k,\ell} \quad (30)$$

where in case of a singular operator  $\mathbf{D}$  the solvability condition (11) is assumed to be satisfied, and the solution based on the Green's functions is no longer unique.

In practice, it is often not straightforward to determine the Green’s functions, since they depend on the domain as well as on the boundary conditions. There exist some designated approaches for specific problem settings [16]. The probably most promising technique is the so-called *method of eigenfunction expansion* [16] for the continuous case. In the discrete setting, the discrete Green’s functions are expressed in terms of the eigenvectors and corresponding eigenvalues of  $\mathbf{D}$  (cf. [1]). Let us now study this approach in detail.

### 3.4 Constructing Discrete Green’s Functions for Our Operators

Let us now apply our theory on the discrete Laplace or biharmonic operator. To this end, we recall that both operators  $-\mathbf{L}$  and  $\mathbf{B}$  have a zero eigenvalue  $\lambda_{0,0}$ . It belongs to the constant eigenvector  $\mathbf{v}_{0,0}$  with entries  $1/\sqrt{MN}$ . Thus, we know from Section 3 that in a point  $(k, \ell) \in \Gamma$ , the Green’s function  $\mathbf{g}_{k,\ell}$  for both operators is not unique. It satisfies the following system of equations:

$$(\mathbf{D}\mathbf{g}_{k,\ell})_{i,j} = \delta_{(k,\ell),(i,j)} - \frac{1}{MN} \quad \text{for } (i, j) \in \Gamma \tag{31}$$

with  $\mathbf{D} = -\mathbf{L}$  or  $\mathbf{D} = \mathbf{B}$ , respectively. The theorem below states the solution in a closed form:

**Theorem 5 (Discrete Green’s Functions).** *In a point  $(k, \ell) \in \Gamma$  the discrete Green’s functions for the matrix  $\mathbf{D} = -\mathbf{L}$  or  $\mathbf{D} = \mathbf{B}$  are given by*

$$(\mathbf{g}_{k,\ell}^c)_{i,j} = \sum_{\substack{m=0 \\ (m,n) \neq (0,0)}}^{M-1} \sum_{n=0}^{N-1} \left[ \frac{1}{\lambda_{m,n}} \cdot (\mathbf{v}_{m,n})_{k,\ell} \cdot (\mathbf{v}_{m,n})_{i,j} \right] + c, \tag{32}$$

where  $\lambda_{m,n}$  are the eigenvalues corresponding to the eigenvectors  $\mathbf{v}_{m,n}$  of  $\mathbf{D}$ , and the constant  $c \in \mathbb{R}$  can be chosen arbitrarily.

*Proof.* Following [1], we express the Green’s function in terms of the orthonormal eigenvectors:

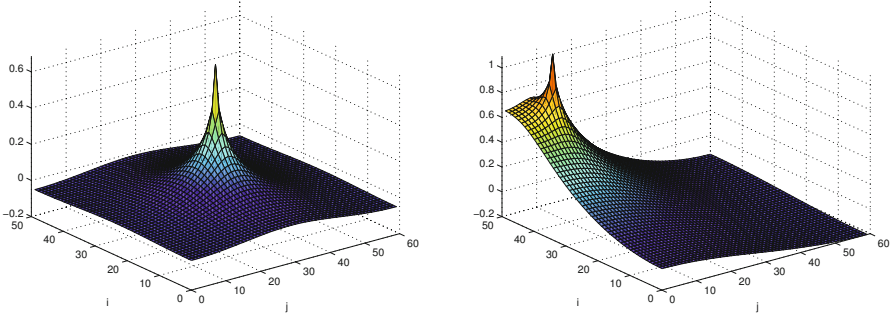
$$\mathbf{g}_{k,\ell} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_{m,n} \mathbf{v}_{m,n} \tag{33}$$

with coefficients  $c_{m,n} \in \mathbb{R}$ . Plugging this into (31) yields

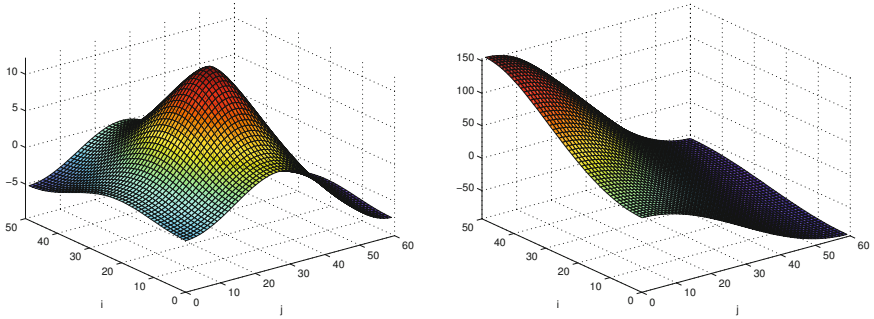
$$\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_{m,n} \lambda_{m,n} (\mathbf{v}_{m,n})_{i,j} = \delta_{(k,\ell),(i,j)} - \frac{1}{MN}. \tag{34}$$

After multiplying both sides with  $(\mathbf{v}_{m',n'})_{i,j}$  for fixed  $(m', n') \in \Gamma$ , and summing up over all pixels  $(i, j) \in \Gamma$ , we have

$$c_{m',n'} \lambda_{m',n'} = (\mathbf{v}_{m',n'})_{k,\ell} - \frac{1}{MN} \sum_{(i,j) \in \Gamma} (\mathbf{v}_{m',n'})_{i,j}. \tag{35}$$



**Fig. 1.** Example of discrete Green's functions for the negative Laplacian with homogeneous Neumann boundary conditions on an image with  $50 \times 60$  pixels. **Left:**  $\mathbf{g}_{25,30}^0$ . **Right:**  $\mathbf{g}_{45,10}^0$



**Fig. 2.** Example of discrete Green's functions for the biharmonic operator with homogeneous Neumann boundary conditions on an image with  $50 \times 60$  pixels. **Left:**  $\mathbf{g}_{25,30}^0$ . **Right:**  $\mathbf{g}_{45,10}^0$

For  $m' = n' = 0$ , the eigenvalue  $\lambda_{0,0}$  as well as the right hand side become 0 by (7). Thus,  $c_{0,0}$  can be chosen arbitrarily. This means that the Green's function is unique up to a constant  $c$ . For  $m' > 0$  or  $n' > 0$ , we obtain

$$c_{m,n} = \frac{1}{\lambda_{m,n}} (\mathbf{v}_{m,n})_{k,\ell}. \tag{36}$$

This concludes the proof. □

We specify a canonic representative  $\mathbf{g}_{k,\ell}^0$  by setting the constant  $c := 0$ . As the eigenvectors  $\mathbf{v}_{m,n}$  with  $(m,n) \neq (0,0)$  of the discrete operator are orthogonal to  $\mathbf{v}_{0,0}$ , this is equivalent to assuming  $\langle \mathbf{g}_{k,\ell}^0, \mathbf{v}_{0,0} \rangle = 0$ . This shows that the obtained Green's functions  $\mathbf{g}_{k,\ell}^0$  have mean value zero. Moreover, we can apply Theorem 3 and see that they build the Moore–Penrose inverse of  $\mathbf{D}$ . Example plots of Green's functions are depicted in Figure 1 and 2.

In practice, we can exploit the symmetry of the rectangular image domain to reduce the effort for computing all discrete Green’s functions by a factor of 4: Once the Green’s function is computed for a specific source point  $(k, \ell) \in \Gamma$ , the Green’s functions for the source points  $(M - k, \ell)$ ,  $(k, N - \ell)$ , and  $(M - k, N - \ell)$  can be obtained by mirroring  $\mathbf{g}_{k,\ell}^0$  along the  $x$  axis, the  $y$  axis and both axes.

### 3.5 Inpainting with Green’s Functions

We want to use the Green’s functions to find an exact solution of the discrete inpainting problem. Therefore, the trick is to rewrite the problem such that it has the form as in (10). We construct a right hand side  $\mathbf{a}$  such that it is zero at all non-mask points, while its values at all mask points  $(i, j) \in K$  must be determined later. As a result, the problem reads as

$$D\mathbf{u} = \mathbf{a} \tag{37}$$

subject to

$$u_{i,j} = f_{i,j} \quad \text{if } (i, j) \in K, \tag{38}$$

$$a_{i,j} = 0 \quad \text{if } (i, j) \in \Gamma \setminus K. \tag{39}$$

Assuming that  $\langle \mathbf{v}_{0,0}, \mathbf{a} \rangle = 0$  we can write the solution  $\mathbf{u}$  of (37) as

$$u_{i,j} = \sum_{(k,\ell) \in \Gamma} a_{k,\ell} \cdot (\mathbf{g}_{k,\ell}^0)_{i,j} + c \tag{40}$$

with the discrete canonic Green’s functions  $\mathbf{g}_{k,\ell}^0$  ( $(k, \ell) \in \Gamma$ ) and an unknown constant  $c$ , comprising all constants of the individual Green’s functions. As by (39) the entries of  $\mathbf{a}$  vanish at all non-mask points, (40) can be simplified to

$$u_{i,j} = \sum_{(k,\ell) \in K} a_{k,\ell} \cdot (\mathbf{g}_{k,\ell}^0)_{i,j} + c. \tag{41}$$

This representation shows that the inpainting solution can be composed by a small number of atoms, namely the discrete Green’s functions corresponding to the mask pixels. Thus, the discrete Green’s functions  $\mathbf{g}_{k,\ell}^0$  corresponding to  $(k, \ell) \in K$  can be seen as a generating system for the space of all inpainting solutions on  $\Gamma \setminus K$  (with mean value zero on  $\Gamma$ ).

It remains to find the unknown coefficients  $c$  and  $a_{k,\ell}$ ,  $(k, \ell) \in K$ . They are determined by (38). Together with the solvability condition (11) within the Fredholm alternative,

$$\langle \mathbf{v}_{0,0}, \mathbf{a} \rangle = 0 \quad \iff \quad \sum_{(k,\ell) \in K} a_{k,\ell} = 0, \tag{42}$$

---

**Algorithm 1.** Inpainting with Green’s functions

---

**Input:** Image  $f$  at specified mask  $K$ .

1. For all  $(k, \ell) \in K$ , compute the corresponding canonic Green’s function  $\mathbf{g}_{k,\ell}^0$  using Theorem 5.
2. Compute the unknown coefficients of  $\mathbf{a}$  and  $c$  by solving (43).
3. Obtain the solution  $\mathbf{u}$  as the superposition given in (41).

**Output:** Inpainting solution  $\mathbf{u}$ .

---

we can specify the inpainting result uniquely. Denoting the 2D pixel indices of the mask points by  $m_1, \dots, m_L$ , with  $L := |K|$ , we can formulate the linear system of equations for finding the unknown values of  $\mathbf{a}$  and  $c$ :

$$\begin{pmatrix} (\mathbf{g}_{m_1}^0)_{m_1} & (\mathbf{g}_{m_2}^0)_{m_1} & \dots & (\mathbf{g}_{m_L}^0)_{m_1} & 1 \\ (\mathbf{g}_{m_1}^0)_{m_2} & (\mathbf{g}_{m_2}^0)_{m_2} & \dots & (\mathbf{g}_{m_L}^0)_{m_2} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (\mathbf{g}_{m_1}^0)_{m_L} & (\mathbf{g}_{m_2}^0)_{m_L} & \dots & (\mathbf{g}_{m_L}^0)_{m_L} & 1 \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} a_{m_1} \\ a_{m_2} \\ \vdots \\ a_{m_L} \\ c \end{pmatrix} = \begin{pmatrix} f_{m_1} \\ f_{m_2} \\ \vdots \\ f_{m_L} \\ 0 \end{pmatrix}. \tag{43}$$

For solving this system of equations, we recommend the QR algorithm since it does not create error accumulations [18]. Once the values for  $c$  and  $a_{m_1}, \dots, a_{m_L}$  are computed, the inpainting solution  $\mathbf{u}$  is represented exactly with (41). For the reader’s convenience, Algorithm 1 summarises the full workflow.

A decisive advantage of our inpainting algorithm with Green’s functions is that it reveals the influence of each mask point on the overall inpainting result: This influence is described by the respective Green’s function. It is clear that the complexity for finding a solution increases with the number of mask points. Interestingly, this is different to the standard approach of solving the discrete inpainting problem iteratively, where it is computationally more expensive to find a solution for a sparse mask: In the latter case, it typically takes more time to diffuse the information at the mask points over the complete image. In contrast, our new approach can compute the solution much faster if the specified data is sparse. For image compression applications this can be a relevant scenario.

## 4 Experiments

Although the main goal of our paper is to emphasise the theoretical advantages of Green’s functions as a tool to understand the connections between PDE-based inpainting and sparsity, our framework can also offer practical advantages. This shall be illustrated by an application in the context of image compression with PDEs. In order to reconstruct an image in the decoding step, we have to solve inpainting problems. If they use the Laplacian or biharmonic operator, we propose to refrain from storing the greyvalues at all mask pixels and rather store the coefficients  $c$  and  $a_{m_1}, \dots, a_{m_{L-1}}$  instead. Note that the missing coefficient  $a_{m_L}$  can be recovered from these coefficients with the help of the solvability

**Table 1.** Runtime comparison for inpainting with the Laplace operator. The CPU time is given in seconds.

mask density	0.01%	0.5%	1%	2%	4%	8%	16%
multigrid (max. error 0.5)	0.425	0.306	0.305	0.305	0.264	0.263	0.216
multigrid (max. error 0.05)	0.777	0.855	0.581	0.579	0.263	0.263	0.216
multigrid (max. error 0.005)	11.331	2.238	1.685	0.857	0.742	0.502	0.216
our approach	0.001	0.037	0.073	0.143	0.293	0.585	1.179

**Table 2.** Runtime comparison for inpainting with the biharmonic operator. The CPU time is given in seconds.

mask density	0.01%	0.5%	1%	2%	4%	8%	16%
multigrid (max. error 0.5)	0.691	0.463	0.464	0.462	0.382	0.382	0.305
multigrid (max. error 0.05)	0.688	0.876	0.875	0.874	0.382	0.383	0.305
multigrid (max. error 0.005)	5.312	2.114	1.287	1.306	0.725	0.382	0.305
our approach	0.001	0.037	0.074	0.148	0.298	0.597	1.181

condition (42). The computation of the Green’s functions can be performed offline before storing them on the hard disk. This has the advantage that they do not have to be recomputed every time they are needed. As a result, we obtain a very efficient decoding for sparse masks where the inpainting result is computed by a simple superposition of Green’s functions.

To evaluate this algorithm for inpainting with the Laplace or biharmonic operator, we compare it with bidirectional multigrid methods. These sophisticated numerical algorithms belong to the most efficient techniques that are used for this purpose; see e.g. [14]. As a model problem, we consider an image of size  $256 \times 256$  pixels with greyvalues in the range between 0 and 255. Moreover, we use randomly sampled mask points with varying density. Table 1 juxtaposes the runtimes of our Green’s function algorithm and bidirectional multigrid methods with two different accuracy levels for the Laplace operator. Corresponding comparisons for the biharmonic operator are presented in Table 2. We use C implementations on an Intel Xeon quadcore architecture with 3.2 GHz and 24 GB memory. For more details on the multigrid implementation, we refer to [14].

We observe that our Green’s function approach gives favourable results if the mask density is low and high accuracy is needed. In the context of depth map compression for example, one usually deals with very sparse masks as only few data points suffice to represent smooth transitions [13]. This shows the practical relevance of the presented algorithm. Note that in contrast to the bidirectional multigrid approach, the Green’s function algorithm solves the discrete inpainting problem exactly (up to machine precision). Thus, there is no need for devising appropriate stopping criteria and making decisions on the numerous parameters that are characteristic for multigrid methods. Last but not least, it should be emphasised that the runtime of the Green’s function method does not deteriorate when one replaces the Laplace operator by the biharmonic operator (or even higher order linear differential operators). It remains a simple superposition of Green’s functions.

## 5 Conclusion

Since one decade, the paradigms of sparse signal processing and inpainting methods for compact image representations have been enjoying a successful development. Although they often pursue similar goals, it is surprising that this has happened without any interaction. With our paper, we have paved the way for a mutual exchange of ideas.

The key concept for understanding this relation was the notion of discrete Green's functions. They serve as atoms in a dictionary. Only a single atom is needed to describe the global influence of one mask pixel. This allows to reinterpret successful inpainting methods with linear differential operators in terms of sparsity. Moreover, discrete Green's functions also offer an interesting interpretation as columns of the Moore–Penrose pseudoinverse of the discretised (singular) differential operator.

Our framework is fairly general: It is directly applicable to any linear selfadjoint differential operator with a known eigendecomposition. We have illustrated this by means of the Laplace operator with homogeneous Neumann boundary conditions and its biharmonic counterpart.

One important result of our Green's function research is the fact that it allows us to have direct access to the exact solution of the discrete inpainting problem. This may also have practical advantages for PDE-based decoding with sparse inpainting masks. In our ongoing research, we are also exploring applications of Green's functions within the encoding step.

It is worth mentioning that our representation of PDE-based inpainting in terms of Green's functions also connects PDE-based image compression to scattered data interpolation with radial basis functions [2]. Many of these basis functions are given as continuous Green's functions on an unbounded domain. With our research we have taken into account the discreteness of digital images and have incorporated image boundaries in a natural way.

**Acknowledgements.** We gratefully acknowledge the partial funding by the Deutsche Forschungsgemeinschaft (DFG) through a Gottfried Wilhelm Leibniz Prize for Joachim Weickert.

## References

1. Berger, J.M., Lasher, G.J.: The use of discrete Green's functions in the numerical solution of Poisson's equation. *Illinois Journal of Mathematics* 2(4A), 593–607 (1958)
2. Buhmann, M.D.: *Radial Basis Functions*. Cambridge University Press, Cambridge (2003)
3. Chen, Y., Ranftl, R., Pock, T.: A bi-level view of inpainting-based image compression. In: Kúkelová, Z., Heller, J. (eds.) *Proc. 19th Computer Vision Winter Workshop, Křtiny, Czech Republic (February 2014)*
4. Chung, F., Yau, S.T.: Discrete Green's functions. *Journal of Combinatorial Theory, Series A* 91(12), 191–214 (2000)

5. Colton, D.: *Partial Differential Equations: An Introduction*. Dover, New York (2004)
6. Evans, G., Blackledge, J., Yardley, P.: *Analytic Methods for Partial Differential Equations*. Springer, London (2000)
7. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.-P.: Towards PDE-based image compression. In: Paragios, N., Faugeras, O., Chan, T., Schnörr, C. (eds.) *VLSM 2005*. LNCS, vol. 3752, pp. 37–48. Springer, Heidelberg (2005)
8. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision* 31(2-3), 255–269 (2008)
9. Gautier, J., Meur, O.L., Guillemot, C.: Efficient depth map compression based on lossless edge coding and diffusion. In: *Picture Coding Symposium, Kraków, Poland*, pp. 81–84 (May 2012)
10. Golub, G.H., Van Loan, C.F.: *Matrix computations*. The John Hopkins University Press, New York (1996)
11. Grewenig, S., Weickert, J., Bruhn, A.: From box filtering to fast explicit diffusion. In: Gesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) *DAGM 2010*. LNCS, vol. 6376, pp. 533–542. Springer, Heidelberg (2010)
12. Hoeltgen, L., Setzer, S., Weickert, J.: An optimal control approach to find sparse data for Laplace interpolation. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMMCVPR 2013*. LNCS, vol. 8081, pp. 151–164. Springer, Heidelberg (2013)
13. Hoffmann, S., Mainberger, M., Weickert, J., Puhl, M.: Compression of depth maps with segment-based homogeneous diffusion. In: Pack, T. (ed.) *SSVM 2013*. LNCS, vol. 7893, pp. 319–330. Springer, Heidelberg (2013)
14. Mainberger, M., Bruhn, A., Weickert, J., Forchhammer, S.: Edge-based image compression of cartoon-like images with homogeneous diffusion. *Pattern Recognition* 44(9), 1859–1873 (2011)
15. Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for homogeneous diffusion inpainting. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) *SSVM 2011*. LNCS, vol. 6667, pp. 26–37. Springer, Heidelberg (2012)
16. Melnikov, Y., Melnikov, M.: *Green’s Functions: Construction and Applications*. De Gruyter, Berlin (2012)
17. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences* 7(2), 1388–1419 (2014)
18. Sauer, T.: *Numerical Analysis*. Pearson Addison Wesley, Boston (2006)
19. Schmaltz, C., Peter, P., Mainberger, M., Ebel, F., Weickert, J., Bruhn, A.: Understanding, optimising, and extending data compression with anisotropic diffusion. *International Journal of Computer Vision* 108(3), 222–240 (2014)
20. Strang, G., MacNamara, S.: Functions of difference matrices are Toeplitz plus Hankel. *SIAM Review* 56(3), 525–546 (2014)



# A Fast Projection Method for Connectivity Constraints in Image Segmentation

Jan Stühmer and Daniel Cremers

Department of Computer Science,  
Technische Universität München, Germany

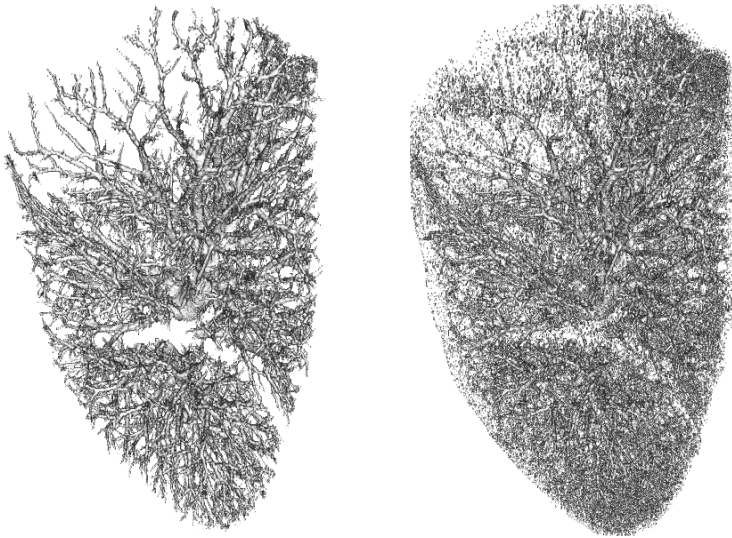
**Abstract.** We propose to solve an image segmentation problem with connectivity constraints via projection onto the constraint set. The constraints form a convex set and the convex image segmentation problem with a total variation regularizer can be solved to global optimality in a primal-dual framework. Efficiency is achieved by directly computing the update of the primal variable via a projection onto the constraint set, which results in a special quadratic programming problem similar to the problems studied as isotonic regression methods in statistics, which can be solved with  $O(n \log n)$  complexity. We show that especially for segmentation problems with long range connections this method is by orders of magnitudes more efficient, both in iteration number and runtime, than solving the dual of the constrained optimization problem. Experiments validate the usefulness of connectivity constraints for segmenting thin structures such as veins and arteries in medical image analysis.

## 1 Introduction

To allow to preserve thin structures, topological constraints, and especially those that preserve connectivity [16,15], have been introduced into image segmentation methods.

These constraints have a great advantage in several application areas, including the segmentation of arteries and veins in medical imaging but also in a user interactive setting for general image segmentation. They are very useful when thin structures should be extracted from image data, allowing to extract the whole branching tree of blood vessels in the lung, as shown on the left in Fig. 1. For comparison, a total variation regularized segmentation of the dataset without connectivity constraints is shown on the right. In order to preserve the thin structures, only a very small weight of the regularizer can be chosen. Therefore a lot of noise is still present in the final segmentation.

Including these constraints in the segmentation model either leads to a higher algorithmic complexity [16,6] or slow convergence when solving the dual of the constrained optimization problem [15].



Result with connectivity constraint Without connectivity constraint

**Fig. 1.** Connectivity constraints allow to extract the whole branching tree of blood vessels in the lung, as shown on the left<sup>1</sup>. For comparison, a total variation regularized segmentation without connectivity constraints is shown on the right. In order to preserve the thin structures, only a very small weight of the regularizer can be chosen, therefore a lot of noise is still present in the final segmentation.

## 1.1 Related Work

Topology preserving constraints have been recently proposed for different algorithmic frameworks. For the graph cut [4] framework, Zeng *et al.* [17] present an extension, that allows to preserve the topology of the result with respect to an initial segmentation. Beginning on a coarse scale, their method preserves the topology of the initial segmentation during refinement. A similar approach was proposed by Han *et al.* [11] for the level set framework. The drawback of both methods is that they depend on the initialization and therefore only reach a local optimum.

Vicente *et al.* [16] introduce connectivity priors into interactive segmentation in a Markov random field framework and enforce connectivity to user given seed points. The authors show that the original problem is NP-hard and propose a greedy approximation scheme consisting of a Dijkstra algorithm where in every expansion step a graph cut needs to be solved. Their method also only reaches a local optimum.

Chen *et al.* [6] propose to alternately solve a graph cut and modify the unary terms based on a level-set representation until predefined topological constraints are fulfilled. The runtime complexity of the method prevents to use it for large scale problems.

<sup>1</sup> CT dataset from the *Vessel Segmentation in the Lung 2012 Grand Challenge*.

Recently, three different methods were proposed, that aim to reach a global optimum. First, Nowozin and Lampert [12] propose to formulate the image segmentation problem with topological constraints as a linear program relaxation. However, even for small image sizes the runtime complexity of the method does not scale well and the relaxation is not tight. In contrast to the method presented in this publication, their method is not suitable for large scale problems in 3D segmentation.

Gulshan *et al.* [10] introduce geodesic star shape priors into the graph cut framework. The solution of the segmentation is restricted to the shape of a geodesic star around an input seed, while the geodesic distance depends on the image gradient. If multiple input seeds are given, the foreground segment takes the form of a geodesic forest, the union of the geodesic stars for every seed. A drawback of their method is that the boundary length regularizer is affected by the discretization of the pixel neighborhood.

In a previous work [15] we propose a global optimal segmentation method with connectivity constraints in a convex optimization framework. The combination of a total variation regularizer with a connectivity constraint allows to segment thin structures even in very noisy image data. Compared to the work of Gulshan *et al.* [10] our method uses a continuous segmentation framework and therefore the boundary length regularizer is not biased by discretization artifacts. The constrained optimization problem in [15] is solved by computing a solution of the dual problem. In this work, we propose an efficient projection scheme to directly compute a solution for the update of the primal variable.

## 1.2 Contribution

We propose to solve an image segmentation problem with connectivity constraints via projection onto the constraint set. We show that the constraints form a convex set and derive a projection algorithm from isotonic regression methods in statistics. We show that especially for segmentation problems with long range connections this method is by orders of magnitudes more efficient, both in iteration number and runtime, than solving the dual of the constrained optimization problem.

## 2 Connectivity Constraints in Image Segmentation

First lets review the results from [15] where image segmentation with connectivity constraints is formalized as the constrained optimization problem

$$\min_{u \in BV(\Omega; [0,1])} \int_{\Omega} f(x) u(x) + |\nabla u| dx \quad (1)$$

s.t.

$$\forall x \in \Omega, u(x) = 1 : \exists C_s^x \in \mathcal{G}_s : u(C_s^x(t)) = 1. \quad \mathbf{C1}$$

where  $I$  is an image with the domain  $\Omega$ , a bounded connected subset of  $\mathbb{R}^m$ ,  $BV(\Omega; [0, 1])$  is the space of functions with bounded variation and  $f : \Omega \rightarrow \mathbb{R}$  depends on the image data. The data term  $f$  is chosen in such a way that it is negative for image values which are more likely to be foreground and negative in regions which should be regarded as background, e.g. the log ratio  $f(x) = \log \frac{P(I(x)|l(x)=0)}{P(I(x)|l(x)=1)}$ . The discrete label assignment  $l : \Omega \rightarrow \{0, 1\}$ , that describes if an image region belongs to the object of interest  $l(x) = 1$  or the image background  $l(x) = 0$ , is relaxed by introducing the continuous indicator function  $u : \Omega \rightarrow [0, 1]$ . The total variation regularizer  $|\nabla u|$  measures the boundary length of the foreground segment. With  $C_s^x$  we formalize the shortest geodesic path from a given starting point  $s$ , for example defined by user input, to a terminal point  $x$  which is part of the geodesic shortest path tree  $\mathcal{G}_s$ .

The solution of the optimization problem should satisfy the connectivity constraint **C1**:

*For each  $x \in \Omega$  that belongs to the foreground there must exist a connected shortest geodesic path from a given  $s \in \Omega$  to  $x$  such that all  $p \in \Omega$  in the path between  $x$  and  $s$  belong to the foreground.*

This constraint not only ensures the connection of every labeled foreground region to  $s$  but also ensures that the whole foreground segment is connected.

## 2.1 Geodesic Distances

Recently, shortest geodesic distance measures have been successfully applied to image segmentation problems including medical image segmentation [3] as well as general image segmentation [1, 7].

In order to define the geodesic shortest path tree  $\mathcal{G}_s$ , first we have to choose an appropriate local geodesic metric. If  $\lambda = 0$  the labeling function  $u(x)$  takes the value 1 for  $f(x) < 0$  and 0 for  $f(x) > 0$ . We leave out the special case  $f(x) = 0$  as it does not occur in practice. For all  $x_p \in \Omega$  that do not belong to the foreground but need to be added to the foreground to satisfy the connectivity constraint obviously  $u(x_p) = 0$  and therefore  $f(x_p) \geq 0$ . The optimal cost of the connecting path between a fixed  $s$  and any  $x$  in the region that should be connected on  $\mathcal{G}_s$  is then given by

$$\min_{C_s^x} \int_0^T f^+(C(t)) dt, \quad (2)$$

with  $f^+ = \max(0, f(x))$ . Thus, we choose the non negative cost function  $f^+$  as metric for the construction of  $\mathcal{G}_s$ . Thus the shortest path tree can be computed using Dijkstra's algorithm [8].

More complex prior models for the geodesic path are possible. In [15] we could show that a bending energy prior for the construction of the geodesic shortest path tree can improve the segmentation performance on a retinal blood vessel dataset to some extent.

### 3 Constrained Convex Optimization

The geodesic shortest path tree forms a directed acyclic graph  $\mathcal{G}_s = \{V, E\}$  with the set of vertices  $V$  with  $|V| = n$  and the set of directed edges  $E \subset V \times V$  with  $|E| = m$ . We follow [15] and formulate the global connectivity constraint as a monotonicity constraint over each edge of this graph. To satisfy the connectivity constraint we observe that the value of the discretized value function  $u_i$  of a node  $i$  with distance to the root node  $d_i$  should always be greater or equal than the labels of its neighbors with a larger distance  $d_j > d_i$  to the root node. This implies that the *directional derivative*

$$\partial_i u_j := (du)(e_{ij}) = (u(j) - u(i))$$

of  $u$  at vertex  $i$  along the edge to vertex  $j$  should always be less or equal to zero.

The image segmentation problem Eq. (1) thus can be written as the constrained optimization problem

$$\begin{aligned} \min_{u_i \in [0,1]} & \int_{\Omega} f(x) u(x) + \lambda |\nabla u| dx & (3) \\ \text{s.t.} & \\ \partial_i u_j & \leq 0, \forall (i, j) \in E. \end{aligned}$$

This image segmentation problem can be optimized using the Primal-Dual framework of [14,5] which can be applied to convex optimization problems with a saddle-point structure

$$\min_{u \in U} \max_{p \in P} \langle Ku, p \rangle + G(u) - F^*(p), \tag{4}$$

where  $U$  and  $P$  are finite-dimensional vector spaces,  $K : U \rightarrow P$  is a continuous linear operator and  $G : U \rightarrow [0, +\infty)$  and  $F^* : P \rightarrow [0, +\infty)$  are proper, convex, lower semicontinuous functions. The update steps in [5] are computed using the **prox**-operator, which is defined as

$$v = (I + \tau \partial G)^{-1}(u) = \arg \min_v \left\{ \frac{\|u - v\|^2}{2\tau} + G(v) \right\}. \tag{5}$$

Using this **prox**-operator, the updates in the primal variable  $u$  and the dual variable  $p$  are computed as

$$u^{k+1} = (I + \tau \partial G)^{-1}(u^k - \tau K^* p^{k+1}) \tag{6}$$

$$p^{k+1} = (I + \sigma \partial F^*)^{-1}(p^k + \sigma K(u^{k+1} + \theta(u^{k+1} - u^k))). \tag{7}$$

To formulate the image segmentation problem Eq. (3) in the Primal-Dual framework we reformulate the total variation regularizer by introducing a dual variable  $p \in R^2$  [14] and after discretization arrive at the saddle point problem

$$\min_{u_i \in [0,1]} \max_{|p| \leq 1} \lambda \langle \nabla u, p \rangle + \langle f, u \rangle + \delta_{\leq 0}(\nabla_i u), \tag{8}$$

where  $\nabla_i u$  is the stacked vector of the directional derivatives  $\partial_i u_j$  and the connectivity constraint is included by adding its indicator function<sup>1</sup>. We identify the function  $G(u)$  in Eq. (4) with  $G(u) = \langle f, u \rangle + \delta_{\leq 0}(\nabla_i u)$ .

While the constraints over the domains of  $u$  and  $p$  can be solved by simple projections, the optimization with respect to the connectivity constraint is more involved. In the following, we will investigate two different strategies to incorporate the connectivity constraint.

### 3.1 Optimization via Fenchel Duality

In [15] we propose to optimize the dual of the constrained optimization problem

$$\min_{u_i \in [0,1]} \max_{\substack{|p| \leq 1 \\ \alpha \geq 0}} \lambda \langle \nabla u, p \rangle + \langle f, u \rangle + \langle \alpha, \nabla_i u \rangle. \quad (9)$$

The connectivity constraint is ensured by introducing an additional dual variable  $\alpha_{ij}$  for each edge  $(i, j) \in E$ . Especially for long range connections the convergence of these multipliers is very slow as we show in our experiments in section 4.

### 3.2 Projection onto the Constraint Set

In this section we describe how the connectivity constraint can be included by directly computing the update of the primal variable subject to this constraint. Therefore we propose an efficient projection scheme to solve the constrained quadratic programming problem, which results from the definition of the **prox**-operator.

According to [5] the update in the primal variable  $u$  is defined as

$$u^{k+1} = (I + \tau \partial G)^{-1}(u^k + \tau \operatorname{div} p^{k+1}) \quad (10)$$

$$= \arg \min_{v \in [0,1]} \left\{ \frac{\|v - (u^k + \tau \operatorname{div} p^{k+1})\|^2}{2\tau} + \langle f, v \rangle + \delta_{\leq 0}(\nabla_i v) \right\}. \quad (11)$$

By completing the square and omitting terms independent of  $v$  we arrive at

$$u^{k+1} = \arg \min_{v \in [0,1]} \{ \|v - (u^k + \tau \operatorname{div} p^{k+1} - \tau f)\|^2 + \delta_{\leq 0}(\nabla_i v) \} \quad (12)$$

which is of the general form

$$\arg \min_{v_i \in [0,1]} \|v - \tilde{u}\|^2 \quad (13)$$

s.t.

$$v_i \geq v_j, \quad \forall (i, j) \in E,$$

with  $\tilde{u} = (u^k + \tau \operatorname{div} p^{k+1} - \tau f)$ .

<sup>1</sup> Note that while  $\nabla_i u$  is defined on the graph  $\mathcal{G}_s$ , the gradient  $\nabla u$  used in the total variaton regularizer is computed using standard forward operators on the image grid.

**Proposition 1.** *The feasible set  $C$  determined by the constraints of the optimization problem Eq. (13) is a convex set.*

*Proof.* Let  $C_1$  be the feasible set determined by the inequality constraints and  $C_2$  the constraint on the range of  $v$ . The feasible set of Eq. (13) then is  $C = C_1 \cap C_2$ . First we show that  $C_1$  is convex. If for every  $a, b \in C_1$  and  $\alpha, \beta > 0$  it holds that  $\alpha a + \beta b \in C_1$  then  $C_1$  is a convex cone. Because  $a, b \in C_1$  it holds that

$$a_i \geq a_j, b_i \geq b_j, \quad \forall (i, j) \in E, \tag{14}$$

and because  $\alpha, \beta > 0$  it follows

$$\alpha a_i \geq \alpha a_j, \beta b_i \geq \beta b_j, \quad \forall (i, j) \in E, \tag{15}$$

$$\alpha a_i + \beta b_i \geq \alpha a_j + \beta b_j, \quad \forall (i, j) \in E. \tag{16}$$

Hence the set  $C_1$  is a convex cone. In addition to the inequality constraints we also have the constraint on the range of  $v$ . We call the feasible set of this constraint  $C_2 = [0, 1]$ . This set is convex, so  $C = C_1 \cap C_2$ , the intersection of two convex sets, is convex.  $\square$

Thus the optimization problem Eq. (13) is strictly convex subject to convex constraints. Its solution is an Euclidean projection of  $\tilde{u}$  onto the set  $C$  and can be solved to global optimality. Furthermore the inequality constraints describe a partial order on the values of  $v$ . A quadratic programming problem with this structure is known in statistics as isotonic regression [2].

### 3.3 Isotonic Regression on a Tree

In Pardalos *et al.* [13] the authors investigate a class of algorithms for isotonic regression where the constraints define a partial order which can be represented by a directed graph. In particular the authors propose an  $O(n \log n)$  algorithm for the case when the directed graph is a directed tree with  $n$  vertices. For convenience we present the algorithm IRT-BIN here as Algorithm 1.

We call the isotonic regression problem subject to partial order constraints *IRT*. This problem does not include the range constraints of Eq. (13). In the following, we will show that a projection of the optimal solution of *IRT* on the range constraint yields the optimal solution of Eq. (13).

First we follow the presentation of Pardalos *et al.* [13] and describe the algorithm for isotonic regression with partial order constraints, using the concept of *upper sets*, *lower sets* and *level sets*:

**Definition 1.** *Let  $X$  be a nonempty finite set. Let  $\preceq$  be a partial order on  $X$ . Let  $Y$  be a nonempty subset of  $X$ . We define the average of  $Y$  as  $A_v(Y) = \frac{1}{|Y|} \sum_{i \in Y} \tilde{u}_i$ . We call a subset  $L \subset X$  a lower set of  $X$  with respect to  $\preceq$  if  $i \in X, j \in L$  and  $i \preceq j$  implies  $i \in L$ . Consequently a subset  $U \subset X$  is an upper set if  $i \in U, j \in X$  and  $i \preceq j$  implies  $j \in U$ . We call a subset  $S \subset X$  a level*

set if there are an upper set  $U$  and a lower set  $L$  such that  $S = L \cup U$ . A block  $B$  of  $X$  is a nonempty level set such that for each upper set  $U \subset X$  for which  $U \cap B \neq \emptyset$  it holds that  $Av(B) \geq Av(U \cap B)$ .

Furthermore the authors of [13] introduce the concept of a *block class*:

**Definition 2.** A collection  $\Delta$  of blocks of  $X$  is called a block class of  $X$  if

1. the blocks in  $\Delta$  are pairwise disjoint and their union is the set  $X$ .
2. the collection  $\Delta$  can be ordered by a partial-order  $\preceq$  such that  $A \preceq B$  for  $A, B \in \Delta$  if there exist  $i \in A$  and  $j \in B$  such that  $i \preceq j$ .

Note that the collection of all singleton subsets  $\{x\}$  with  $x \in X$  is a block class.

The authors prove that the optimal solution of *IRT* on a block  $B$  is  $v_i = Av(B)$  for every  $i \in B$ . Furthermore they show that if a block class  $\Delta$  has no adjacent violators, then the optimal solution of the isotonic regression is given by  $v_i^* = Av(B(i))$ , where  $B(i)$  is the block which contains  $i$ , for each element  $i$  of  $X$ .

---

**Algorithm 1.** IRT-BIN from Pardalos *et al.* [13]

---

- 1: Let  $\Delta$  be the singleton block class and let  $\mathcal{T}$  be a copy of the underlying rooted tree.
  - 2: Mark each leaf node of  $\mathcal{T}$  as solved and all other nodes as unsolved.
  - 3: **for** each node  $x_i$  of  $\mathcal{T}$  **do**
  - 4:     Create a block  $B(x_i) = \{x_i\}$  and a binomial heap  $H_i$ .
  - 5: **end for**
  - 6: **if** all nodes of  $\mathcal{T}$  are marked as solved **then**
  - 7:     output the blocks corresponding to the nodes in  $\mathcal{T}$  as the final block class and **stop**;
  - 8: **end if**
  - 9: Let  $x_i$  be an unsolved node of  $\mathcal{T}$  such that all the children nodes of  $x_i$  are solved.
  - 10: Let  $B(x_i)$  (resp.  $H_i$ ) be the block (resp. binomial heap) corresponding to node  $x_i$ .
  - 11: **while**  $Av(B(x_i)) < Maximum(H_i)$  **do**
  - 12:      $ExtractMax(H_i)$  and let  $B(x_k)$  be the corresponding block
  - 13:     Shrink the edge connecting  $x_i$  to  $x_k$       $\triangleright$  the new vertex is still called  $v_i$
  - 14:     Create a new block  $B(x_i) \leftarrow B(x_i) \cup B(x_k)$       $\triangleright$  the new block is still called  $B(x_i)$
  - 15:     Calculate the  $Av(B(x_i))$  for the new block  $B(x_i)$
  - 16:      $H_i \leftarrow Union(H_i, H_k)$       $\triangleright$  this is the binomial heap for the new block  $B(x_i)$
  - 17: **end while**
  - 18: Mark the node  $x_i$  of  $\mathcal{T}$  as solved.
  - 19: Let  $x_p$  be the parent node of  $x_i$  in  $\mathcal{T}$ . Let  $H_p$  be the binomial heap corresponding to  $B(x_p)$  and let  $a_i$  be the node in  $H_p$  which corresponds to  $B(x_i)$ .  
        $ChangeKey(a_i, Av(B(x_i)), H_p)$ .
  - 20: **go to** 6.
- 

We will show with the proof of the following proposition that given a solution  $v^*$  of *IRT* the optimal solution to Eq. (13) is achieved by projecting  $v^*$  on  $C_2$ .



Thus, we can directly project onto the constraints of the optimization problem Eq. (13) by first projecting onto the isotonicity constraint and then onto the  $[0, 1]$ -box constraint.

Obviously, projecting first onto the  $[0, 1]$ -box constraint and then onto the isotonicity constraint will not lead to a valid projection. When the averaging step is performed after the  $[0, 1]$  clipping, in case that the isotonicity constraint is violated and some values are smaller 1, only block average values well below 1 can be achieved, even when the average of the block before projection was larger than 1.

**Proposition 2. Direct Projection onto the Constraint Set**

Let  $B$  be a block of  $X$ . Let  $v_i^* = Av(B)$  for every  $i \in B$  be the solution of IRT. Let  $\pi_{[0,1]} : \mathbb{R} \rightarrow [0, 1]$  be a projection that projects negative values to 0 and values larger 1 to 1. Then  $\{\pi_{[0,1]}(v_i^*) : i \in B\}$  is the optimal solution to the optimization problem (13) on  $B$ .

*Proof.* Let us assume that  $B$  has  $m$  elements  $x_1, x_2, \dots, x_m$ . We look at the three cases  $Av(B) > 1$ ,  $Av(B) \in [0, 1]$  and  $Av(B) < 0$ . Obviously these three cases are exhaustive. If  $Av(B) \in [0, 1]$  then the solution  $v^*$  of IRT also fulfills the range constraint and the solution of Eq. (13) for the set  $B$  is identical to the solution of IRT on  $B$ .

If  $Av(B) > 1$  we follow a similar proof as in [13] and show that the point

$$\begin{aligned} \{\pi_{[0,1]}(v_i^*) : i \in B\} &= (\pi_{[0,1]}(Av(B)), \pi_{[0,1]}(Av(B)), \dots, \pi_{[0,1]}(Av(B))) \in \mathbb{R}^m \\ &= (1, 1, \dots, 1) \in \mathbb{R}^m \end{aligned} \tag{17}$$

is the optimal solution to Eq. (13) by showing that the inner product of the gradient of Eq. (13) with any feasible direction  $d \in \mathbb{R}^m$  at that point is a non-negative number.

Let  $d = (d_1, d_2, \dots, d_m)$  be a feasible direction of the isotonic regression problem on  $B$ . Then, in order to preserve isotonicity, feasibility of the direction  $d$  implies  $d_i \leq d_j$  when  $x_i \preceq x_j$ .

Therefore there exists a permutation  $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(m))$  such that

$$d_{\sigma(1)} \geq d_{\sigma(2)} \geq \dots \geq d_{\sigma(m)} \tag{18}$$

and

$$x_{\sigma(i)} \preceq x_{\sigma(j)} \implies i \leq j. \tag{19}$$

To prove that for  $Av(B) > 1$  the point in (29) is the optimal solution of the optimization problem (13) on the set  $B$  it is sufficient show that

$$\sum_{i \in B} (1 - \tilde{u}_{\sigma(i)}) \times d_{\sigma(i)} \geq 0. \tag{20}$$

From Eq. (18) and from the definition of a block it follows that

$$\frac{1}{m - k + 1} \sum_{i=k}^m u_{\sigma(i)} \geq Av(B) > 1 \text{ for all } 1 < k \leq m. \tag{21}$$

This implies that

$$\sum_{i=k}^m (1 - u_{\sigma(i)}) \leq 0 \text{ for all } 1 < k \leq m. \tag{22}$$

Equations (22) and (18) imply that for all  $1 < k \leq m$  that the following inequality holds

$$\sum_{i=k}^m (1 - u_{\sigma(i)}) \times d_{\sigma(k-1)} \geq \sum_{i=k}^m (1 - u_{\sigma(i)}) \times d_{\sigma(k)}. \tag{23}$$

Because  $Av(B) > 1$  the feasibility of  $d$  implies that  $d_{\sigma(i)} \leq 0$  for all  $i \in \{1, \dots, m\}$ . Combining everything together we get

$$\begin{aligned} & \sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(1)} && (24) \\ &= \sum_{i=1}^1 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=2}^m (1 - u_{\sigma(1)}) \times d_{\sigma(1)} \\ &\leq \sum_{i=1}^1 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=2}^m (1 - u_{\sigma(2)}) \times d_{\sigma(2)} \\ &= \sum_{i=1}^2 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=3}^m (1 - u_{\sigma(2)}) \times d_{\sigma(2)} \\ &\leq \sum_{i=1}^2 (1 - u_{\sigma(i)}) \times d_{\sigma(i)} + \sum_{i=3}^m (1 - u_{\sigma(3)}) \times d_{\sigma(3)} \\ &\dots \\ &\leq \sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(i)} && (25) \end{aligned}$$

From  $Av(B) > 1$  it follows that

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) < 0. \tag{26}$$

Together with  $d_{\sigma(i)} \leq 0$  for all  $i \in \{1, \dots, m\}$  it follows for Eq. (24)

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(1)} \geq 0. \tag{27}$$

Therefore from Eq. (24) to Eq. (25) we have proven that if  $Av(B) > 1$

$$\sum_{i=1}^m (1 - u_{\sigma(i)}) \times d_{\sigma(i)} \geq 0. \tag{28}$$

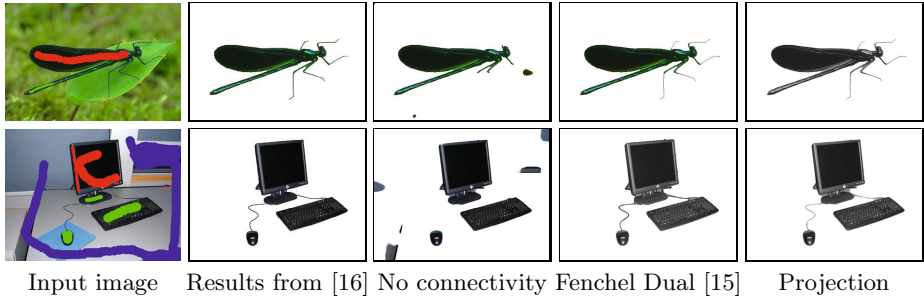
If  $Av(B) < 0$  we have to show that the inner product of the gradient of Eq. (13) with any feasible direction  $d = (d_1, d_2, \dots, d_m) \in \mathbb{R}^m$  at the point

$$\{\pi_{[0,1]}(v_i^*) : i \in B\} = (0, 0, \dots, 0) \in \mathbb{R}^m$$

is a positive number. This proof is equivalent to the proof for  $Av(B) > 1$ .  $\square$

### 4 Experimental Results

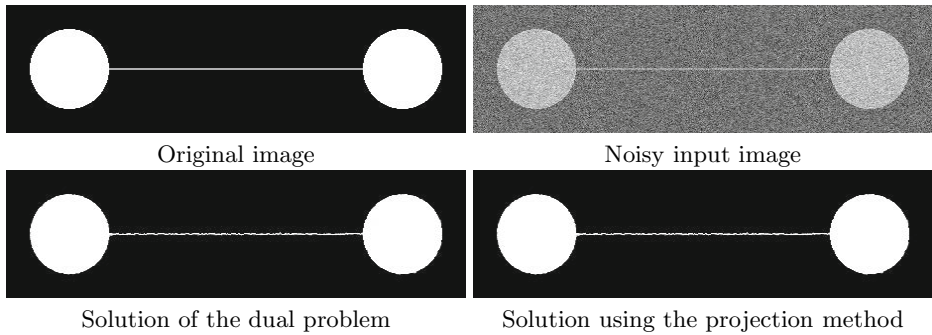
For comparison we performed experiments for interactive segmentation on images from [16] that also have been used in other publications, e.g. [9,15]. As depicted in Fig. 2, the segmentations acquired with the projection method are not different from the results of the algorithm based on Fenchel duality [15].



**Fig. 2.** Connectivity priors for interactive segmentation. First column: Input image with user scribbles. The red scribbles are the source of the geodesic shortest path tree, green scribbles are foreground regions that should be connected and blue scribbles are background regions. Second column: Results from [16]. Third column: Segmentation without connectivity constraints. Fourth column: Segmentation with connectivity constraints by solving the dual problem [15]. Fourth column: Segmentation with connectivity constraints using the proposed projection scheme.

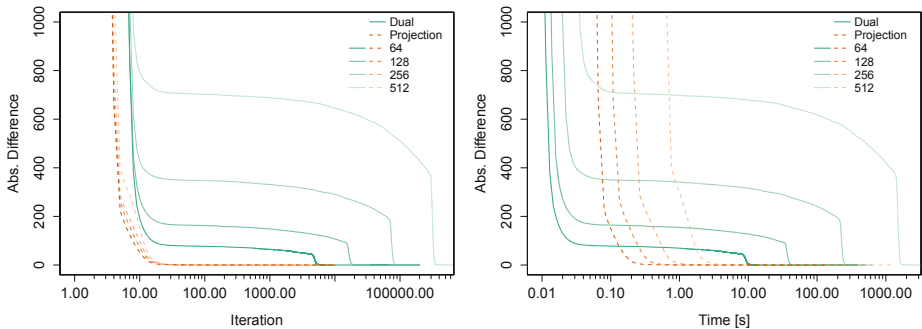
We provide convergence results of the two different methods on a set of synthetic test images. The set contains images of two circles that are connected by a 2 pixel wide faint path of a length of 64, 128, 256 and 512 pixels. As an example, the image for the path length of 256 pixels is shown in Fig. 3.

Plots of the convergence of the two methods with respect to runtime are shown in Fig. 4. The projection method clearly outperforms the method based on Fenchel duality. The longer the connection, the higher the runtime difference of both methods. Convergence of the dual method takes from 10.12 seconds for the 64 pixel connection, over 41.11 seconds for 128, 251.17 seconds for 256 to 1639.15 seconds for the 512 pixel connection, whereas the projection method converges within less than 3 seconds for all different images. Although solving the isotonic regression problem results in a higher complexity of each iteration, by



**Fig. 3.** Synthetic test image. Upper row: The input image with added Gaussian noise. Lower row: Identical results of the two different methods to include the connectivity constraint.

magnitudes fewer iterations are required for the projection method to converge. The needed runtime and number of iterations until convergence for both methods are also shown in Table 1. To measure the speed of convergence we first compute a segmentation result that is reached after a large number of iterations (10000). Then we restart the algorithm and stop when the absolute difference between the current result and the converged result is below 0.1 % of the number of pixels of the image. All Experiments were performed on a single threaded 2.27 GHZ Intel Xeon architecture.



**Fig. 4.** Convergence of the two different methods to include the connectivity constraint on a set of test images as shown in Fig. 3. The set contains images with two circles that are connected by a 2 pixel width path of a length of 64, 128, 256 and 512 pixels. Note that the plots have a logarithmic scale at the x axes. When using the projection method (dashed line), by order of magnitudes fewer iterations are needed than for solving the dual problem (solid). This results in a by order of magnitudes better runtime performance.

**Table 1.** Comparison of runtime and number of iterations until convergence. Especially when the images contain long range connections, the projection method is by magnitudes more efficient than solving the dual problem.

Image	Fenchel Duality		Projection Method	
	Iterations	Runtime	Iterations	Runtime
Test Circle 64	5396	10.12 s	<b>19</b>	<b>0.29 s</b>
Test Circle 128	18318	41.11 s	<b>20</b>	<b>0.52 s</b>
Test Circle 256	81987	251.17 s	<b>20</b>	<b>1.06 s</b>
Test Circle 512	344030	1639.15 s	<b>20</b>	<b>2.89 s</b>
Fly	1226	9.13 s	<b>54</b>	<b>3.66 s</b>
Desk	3440	42.00 s	<b>109</b>	<b>13.40 s</b>

## 5 Conclusion

We presented a very efficient projection scheme to include connectivity constraints in a convex image segmentation framework. The method outperforms commonly used approaches that are based on Fenchel duality by orders of magnitudes. Instead of using the common approach to solve the dual problem of the constrained optimization problem we directly project onto the constraint set thus significantly fewer iterations are needed until a sufficient convergence is reached. This enables to use connectivity constraints for large segmentation problems as they arise for example in medical image segmentation of three dimensional CT angiography.

**Acknowledgements.** We thank Michael McCoy, Michael Möller and Konstantin Pieper for fruitful discussions. This research was supported by the ERC Starting Grant "ConvexVision" and the Technische Universität München - Institute for Advanced Study, funded by the German Excellence Initiative.

## References

1. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8. IEEE (2007)
2. Barlow, R., Brunk, H.: The isotonic regression problem and its dual. *Journal of the American Statistical Association* 67(337), 140–147 (1972)
3. Benmansour, F., Cohen, L.: Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision* 92, 192–210 (2011), <http://dx.doi.org/10.1007/s11263-010-0331-0>
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(11), 1222–1239 (2001)
5. 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), <http://dx.doi.org/10.1007/s10851-010-0251-1>

6. Chen, C., Freedman, D., Lampert, C.H.: Enforcing topological constraints in random field image segmentation. In: Proc. International Conference on Computer Vision and Pattern Recognition, pp. 2089–2096 (2011)
7. Criminisi, A., Sharp, T., Blake, A.: GeoS: Geodesic image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 99–112. Springer, Heidelberg (2008)
8. Dijkstra, E.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959), <http://dx.doi.org/10.1007/BF01386390>
9. El-Zehiry, N.Y., Grady, L.: Fast global optimization of curvature. In: CVPR, pp. 3257–3264. IEEE (2010)
10. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: Proc. International Conference on Computer Vision and Pattern Recognition, pp. 3129–3136. IEEE (2010)
11. Han, X., Xu, C., Prince, J.L.: A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(6), 755–768 (2003)
12. Nowozin, S., Lampert, C.H.: Global connectivity potentials for random field models. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 818–825. IEEE (2009)
13. Pardalos, P.M., Xue, G.: Algorithms for a class of isotonic regression problems. *Algorithmica* 23(3), 211–222 (1999)
14. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth mumford-shah functional. In: IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan (2009)
15. Stühmer, J., Schröder, P., Cremers, D.: Tree shape priors with connectivity constraints using convex relaxation on general graphs. In: IEEE International Conference on Computer Vision (ICCV), Sydney, Australia (December 2013)
16. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: Proc. International Conference on Computer Vision and Pattern Recognition (2008)
17. Zeng, Y., Samaras, D., Chen, W., Peng, Q.: Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding* 112(1), 81–90 (2008)

# Two-Dimensional Variational Mode Decomposition<sup>\*</sup>

Konstantin Dragomiretskiy and Dominique Zosso

Department of Mathematics, University of California, Los Angeles  
520 Portola Plaza, Box 951555, Los Angeles, CA 90095-1555, USA  
{konstantin,zosso}@math.ucla.edu

**Abstract.** In this paper we propose a variational method to adaptively decompose an image into few different modes of separate spectral bands, which are unknown before. A popular method for recursive one dimensional signal decomposition is the Empirical Mode Decomposition algorithm, introduced by Huang in the nineties. This algorithm, as well as its 2D extension, though extensively used, suffers from a lack of exact mathematical model, interpolation choice, and sensitivity to both noise and sampling. Other state-of-the-art models include synchrosqueezing, the empirical wavelet transform, and recursive variational decomposition into smooth signals and residuals. Here, we have created an entirely non-recursive 2D variational mode decomposition (2D-VMD) model, where the modes are extracted concurrently. The model looks for a number of 2D modes and their respective center frequencies, such that the bandlimited modes reproduce the input image (exactly or in least-squares sense). Preliminary results show excellent performance on both synthetic and real images. Running this algorithm on a peptide microscopy image yields accurate, timely, and autonomous segmentation - pertinent in the fields of biochemistry and nanoscience.

## 1 Introduction

In this paper we are interested in decomposing images into ensembles of constituent modes (components) that have specific directional and oscillatory characteristics. Simply put, the goal is to retrieve a small number of modes, that each have a very limited bandwidth around their characteristic center frequency. These modes are called intrinsic mode functions (IMF) and can be seen as amplitude- and frequency-modulated (AM-FM) 2D signals. Such a mode can have limited spatial support, its local (instantaneous) frequency and amplitude vary smoothly, several modes can overlap in space, and together the ensemble of modes should reconstruct the given input image up to noise and singular features.

---

<sup>\*</sup> This work is supported by the National Science Foundation (NSF) under grant DMS-1118971, UC Lab Fees Research grant 12-LR-236660, the Swiss National Science Foundation (SNF) grant P300P2\_147778, and the W. M. Keck Foundation.

The problem is inspired by the one-dimensional Empirical Mode Decomposition (EMD) algorithm [16] and its two-dimensional extensions [20,23,22] for recursive sifting of 2D spatial signals by means of interpolating upper and lower envelopes, median envelopes, and thus extracting image components in different “frequency” bands. This 2D-EMD, however, suffers from the same drawbacks in robustness as the original EMD in extremal point finding, interpolation of envelopes, and stopping criteria imposed.

Other methods of directional image decomposition work by mostly rigid frames, decomposing the Fourier spectrum into fixed, mostly or strictly disjoint, (quasi-)orthogonal basis elements. Examples include Gabor filters [24], wavelets [7,18,9], curvelets [4], shearlets [17,14], etc. These methods are not adaptive relative to the signal, and can attribute principle components of the image to different bands, as well as contain several different image components in the same band. Adaptivity and tuned sparsity concerns have been addressed through synchrosqueezed wavelet transforms [8,6], where unimportant wavelet coefficients are removed by thresholding based on energy content. In pursuit of the same goal, the 2D Empirical Wavelet Transform (EWT) [12,13] decomposes an image by creating an adaptive wavelet basis.

A variational solution for the related decomposition problem in one dimension was recently presented [10]. The so-called *variational mode decomposition* in 1D is essentially based on well established concepts such as Wiener filtering, the 1D Hilbert transform and the analytic signal, and heterodyne demodulation. The goal of 1D-VMD is to decompose an input signal into a discrete number of sub-signals (modes), where each mode has limited bandwidth in spectral domain. In other words, one requires each mode  $k$  to be mostly compact around a center pulsation  $\omega_k$ , which is to be determined along with the decomposition. In order to assess the bandwidth of a mode, the following scheme was proposed [10]: 1) for each mode  $u_k$ , compute the associated analytic signal by means of the Hilbert transform in order to obtain a unilateral frequency spectrum. 2) For each mode, shift the mode’s frequency spectrum to “baseband”, by mixing with an exponential tuned to the respective estimated center frequency. 3) The bandwidth is now estimated through the  $H^1$  smoothness of the demodulated signal, i.e. the squared  $L^2$ -norm of the gradient. The resulting constrained variational problem is the following:

$$\begin{aligned} \min_{u_k, \omega_k} & \left\{ \sum_k \alpha_k \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \\ \text{s.t. } & \forall t: \sum_k u_k(t) = f(t), \end{aligned} \quad (1)$$

where  $*$  denotes convolution.

In this paper we propose a natural two-dimensional extension of the (1D) Variational Mode Decomposition (VMD) algorithm [10] in the context of image segmentation and directional decomposition. 2D-VMD is a non-recursive, fully adaptive, variational method which sparsely decomposes images in a mathematically



well-founded manner with minimal parameters and no explicit interpolation. Practicable applications include the decomposition of images into (possibly overlapping) regions of essentially wave-like nature, in order to make these components accessible for further analysis, such as space-frequency analysis and demodulation.

The rest of this paper is organized as follows: Section 2 presents and explains the two dimensional model. 2D generalizations of the Hilbert transform and the analytic signal are discussed. Once all of the tools from the 1D model are made analogous, we first present the constrained 2D model, and then its unconstrained formulation. The details of the optimization with respect to each unknown are shown, leading to the algorithmic updates of the variables. Section 3 contains our experiments and results, namely of two images, one synthetic multi-mode image and one real  $\beta$ -sheet microscopy image. We present the 2D-VMD algorithm's decomposition of these images, along with the reconstruction. Section 4 concludes on our proposed 2D-VMD method, summarizes again the main assumptions and limitations, and includes some future directions and expected improvements.

## 2 2D Variational Mode Decomposition

We design the 2D model relatively analogous to its 1D predecessor, minimizing the constituent sub-signals bandwidth while maintaining data fidelity. While derivatives in higher dimensions are simply generalized by gradients, and modulation is also straightforward, the generalization of the analytic signal is less obvious. To complete the analogy, we must first define the Hilbert transform along with the analytic signal in the 2D context.

### 2.1 2D Analytic Signal

In 1D, in the time domain, the analytic signal was achieved by adding the Hilbert transformed copy of the original signal as imaginary part:

$$f_{AS}(t) = f(t) + j\mathcal{H}\{f\}(t), \quad (2)$$

where the 1D Hilbert transform is defined as:

$$\mathcal{H}\{f\}(t) := \left\{ \frac{1}{\pi} * f(\cdot) \right\} (t) = \frac{1}{\pi} \text{p.v.} \int_{\mathbb{R}} \frac{f(v)}{t-v} dv. \quad (3)$$

We take note that the real signal is recovered simply by taking the real component of the analytic signal.

In the spectral domain, the analytic signal is obtained by suppressing the negative frequencies, thus giving it a unilateral spectrum:

$$\hat{f}_{AS}(\omega) = \begin{cases} 2\hat{f}(\omega), & \text{if } \omega > 0 \\ \hat{f}(\omega), & \text{if } \omega = 0 \\ 0, & \text{if } \omega < 0. \end{cases} \quad (4)$$

Single-sidedness of the analytic signal spectrum was the key-property motivating its use in the 1D case, since this property allowed for easy frequency shifting to base-band by complex exponential mixing. Therefore, to mimic this spectral property in 2D, one half-plane of the frequency domain must effectively be set to zero; this half-plane is chosen relative to a vector, in our case to  $\boldsymbol{\omega}_k$ . Thus the 2D analytic signal of interest can first be defined in the frequency domain.

$$\hat{u}_{AS,k}(\boldsymbol{\omega}) = \begin{cases} 2\hat{u}_k(\boldsymbol{\omega}), & \text{if } \langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle > 0 \\ \hat{u}_k(\boldsymbol{\omega}), & \text{if } \langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle = 0 \\ 0, & \text{if } \langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle < 0 \end{cases} \quad (5)$$

$$= (1 + \text{sgn}(\langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle))\hat{u}_k(\boldsymbol{\omega})$$

The 2D analytic signal with the aforementioned Fourier property is [3]:

$$u_{AS,k}(\boldsymbol{x}) = u_k(\boldsymbol{x}) * \left( \delta(\langle \boldsymbol{x}, \boldsymbol{\omega}_k \rangle) + \frac{j}{\pi \langle \boldsymbol{x}, \boldsymbol{\omega}_k \rangle} \right) \delta(\langle \boldsymbol{x}, \boldsymbol{\omega}_{k,\perp} \rangle), \quad (6)$$

where  $*$  denotes convolution and the transform is separable. Here, the analytic signal is calculated line-wise along the direction of reference,  $\boldsymbol{\omega}_k$ . These lines are processed independently, hence this definition is intrinsically 1D, but has the desired 2D Fourier property.

## 2.2 2D-VMD Functional

We are now able to put all the generalized VMD-ingredient together to define the two-dimensional extension of variational mode decomposition. The functional to be minimized, stemming from this definition of 2D analytic signal, is:

$$\min_{u_k, \boldsymbol{\omega}_k} \left\{ \sum_k \alpha_k \left\| \nabla \left[ u_{AS,k}(\boldsymbol{x}) e^{-j\langle \boldsymbol{\omega}_k, \boldsymbol{x} \rangle} \right] \right\|_2^2 \right\} \quad \text{s.t.} \quad \forall \boldsymbol{x}: \sum_k u_k(\boldsymbol{x}) = f(\boldsymbol{x}) \quad (7)$$

The objective function is an assessment of the sum of the modes' bandwidths as the squared  $H^1$  norm of its directional 2D analytic signal with only half-space frequencies, shifted to baseband by mixing with a complex exponential of the current center frequency estimate, while maintaining reconstructive signal fidelity.

The reconstruction constraint is addressed through quadratic penalty and Lagrangian multiplier, and we proceed by ADMM for optimization [10,1,19].

## 2.3 ADMM Optimization of 2D-VMD

To render the problem unconstrained, we include both a quadratic penalty and a Lagrangian multiplier to enforce the constraint fidelity; the augmented Lagrangian is now:

$$\begin{aligned} \mathcal{L}(\{u_k\}, \{\omega_k\}, \lambda) := & \sum_k \alpha_k \left\| \nabla \left[ u_{AS,k}(\mathbf{x}) e^{-j\langle \omega_k, \mathbf{x} \rangle} \right] \right\|_2^2 \\ & + \left\| f(\mathbf{x}) - \sum_k u_k(\mathbf{x}) \right\|_2^2 + \left\langle \lambda(\mathbf{x}), f(\mathbf{x}) - \sum_k u_k(\mathbf{x}) \right\rangle. \end{aligned} \quad (8)$$

Our unconstrained problem is then:

$$\min_{u_k, \omega_k} \max_{\lambda} \mathcal{L}(\{u_k\}, \{\omega_k\}, \lambda) \quad (9)$$

The solution to the original constrained minimization problem (7) is now found as the saddle point of the augmented Lagrangian  $\mathcal{L}$  in a sequence of iterative sub-optimizations called alternate direction method of multipliers (ADMM) [15,21,2], see Algorithm 1.

For simplified notation, we incorporate the Lagrangian multiplier term  $\lambda$  into the quadratic penalty term, and rewrite the objective expression slightly different:

$$\sum_k \alpha_k \left\| \nabla \left[ u_{AS,k}(\mathbf{x}) e^{-j\langle \omega_k, \mathbf{x} \rangle} \right] \right\|_2^2 + \left\| f(\mathbf{x}) - \sum_k u_k(\mathbf{x}) + \frac{\lambda(\mathbf{x})}{2} \right\|_2^2 - \left\| \frac{\lambda(\mathbf{x})}{4} \right\|_2^2 \quad (10)$$

**Minimization w.r.t. The Modes  $u_k$ .** The relevant update problem derived from (10) is

$$u_k^{n+1} = \arg \min_{u_k} \left\{ \alpha_k \left\| \nabla [u_{AS,k}(\mathbf{x}) e^{-j\langle \omega_k, \mathbf{x} \rangle}] \right\|_2^2 + \left\| f(\mathbf{x}) - \sum_i u_i(\mathbf{x}) + \frac{\lambda(\mathbf{x})}{2} \right\|_2^2 \right\}. \quad (11)$$

Since we are dealing with  $L^2$ -norms, the functional, including the augmented Lagrangian, can be written in Fourier domain utilizing the  $L^2$  Fourier isometry:

$$\hat{u}_k^{n+1} = \arg \min_{\hat{u}_k} \left\{ \alpha_k \left\| j\omega [\hat{u}_{AS,k}(\omega + \omega_k)] \right\|_2^2 + \left\| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right\|_2^2 \right\} \quad (12)$$

$$= \arg \min_{\hat{u}_k} \left\{ \alpha_k \left\| j(\omega - \omega_k) \hat{u}_{AS,k}(\omega) \right\|_2^2 + \left\| \hat{f}(\omega) - \sum_i \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2} \right\|_2^2 \right\}. \quad (13)$$

These equalities are justified by the well-known transform pair:

$$f(\mathbf{x}) e^{-j\langle \omega_0, \mathbf{x} \rangle} \xleftrightarrow{\mathcal{F}} \hat{f}(\omega) * \delta(\omega + \omega_0) = \hat{f}(\omega + \omega_0), \quad (14)$$

where  $\delta$  is the Dirac distribution and  $*$  denotes convolution. Thus, multiplying an analytic signal with a pure exponential results in simple frequency shifting.

Taking the first variation with respect to  $\hat{u}_k$  and setting it to 0 yields<sup>1</sup>:

$$2\alpha_k |\boldsymbol{\omega} - \boldsymbol{\omega}_k|^2 \hat{u}_k + (-1) \left( \hat{f}(\boldsymbol{\omega}) - \sum_i \hat{u}_i(\boldsymbol{\omega}) + \frac{\hat{\lambda}(\boldsymbol{\omega})}{2} \right) = 0, \quad \forall \boldsymbol{\omega} \in \Omega_k: \Omega_k = \{\boldsymbol{\omega} \mid \langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle \geq 0\}. \quad (15)$$

With this optimality condition, solving for  $\hat{u}_k$  yields the following Wiener-filter update:

$$\hat{u}_k^{n+1}(\boldsymbol{\omega}) = \left( \hat{f}(\boldsymbol{\omega}) - \sum_{i \neq k} \hat{u}_i(\boldsymbol{\omega}) + \frac{\hat{\lambda}(\boldsymbol{\omega})}{2} \right) \frac{1}{1 + 2\alpha_k |\boldsymbol{\omega} - \boldsymbol{\omega}_k|^2} \quad \forall \boldsymbol{\omega} \in \Omega_k: \Omega_k = \{\boldsymbol{\omega} \mid \langle \boldsymbol{\omega}, \boldsymbol{\omega}_k \rangle \geq 0\}, \quad (16)$$

and the other half-plane is completed through Hermitian symmetry. The term in parentheses is the signal's  $k$ th residual, where

$$\hat{f}(\boldsymbol{\omega}) - \sum_{i \neq k} \hat{u}_i(\boldsymbol{\omega})$$

is the explicit current residual, and  $\hat{\lambda}_k$  accumulates the residual in the form of the Lagrangian multiplier.

**Minimization w.r.t. The Center Frequencies  $\boldsymbol{\omega}_k$ .** Optimizing for  $\boldsymbol{\omega}_k$  is even simpler. Indeed, the update goal is

$$\boldsymbol{\omega}_k^{n+1} = \arg \min_{\boldsymbol{\omega}_k} \left\{ \alpha_k \left\| \nabla \left[ u_{AS,k}(\mathbf{x}) e^{-j \langle \boldsymbol{\omega}_k, \mathbf{x} \rangle} \right] \right\|_2^2 \right\}. \quad (17)$$

Or, again in the Fourier domain:

$$\boldsymbol{\omega}_k^{n+1} = \arg \min_{\boldsymbol{\omega}_k} \left\{ \alpha_k \left\| j(\boldsymbol{\omega} - \boldsymbol{\omega}_k) [(1 + \operatorname{sgn}(\langle \boldsymbol{\omega}_k, \boldsymbol{\omega} \rangle)) \hat{u}_k(\boldsymbol{\omega})] \right\|_2^2 \right\} \quad (18)$$

$$= \arg \min_{\boldsymbol{\omega}_k} \left\{ 4\alpha_k \left\| (\boldsymbol{\omega} - \boldsymbol{\omega}_k) \hat{u}_k(\boldsymbol{\omega}) \right\|_{\Omega_k}^2 \right\}. \quad (19)$$

The minimization is solved by letting the first variation w.r.t.  $\boldsymbol{\omega}_k$  vanish (on the frequency halfplane  $\Omega_k$ ):

$$8\alpha_k \int_{\Omega_k} (\boldsymbol{\omega} - \boldsymbol{\omega}_k) |\hat{u}_k|^2 d\boldsymbol{\omega} = 0. \quad (20)$$

The resulting solutions are the first moments of the mode's power spectrum  $|\hat{u}_k(\boldsymbol{\omega})|^2$  on the half-plane  $\Omega_k$ :

$$\boldsymbol{\omega}_k^{n+1} = \frac{\int_{\Omega_k} \boldsymbol{\omega} |\hat{u}_k(\boldsymbol{\omega})|^2 d\boldsymbol{\omega}}{\int_{\Omega_k} |\hat{u}_k(\boldsymbol{\omega})|^2 d\boldsymbol{\omega}}. \quad (21)$$

<sup>1</sup> Note that the functional is complex valued so the process of "taking the first variation" is not self-evident. However, the functional is analytic and complex-valued equivalents to the standard derivatives do indeed apply.

**Maximization w.r.t. the Lagrangian Multiplier  $\lambda$ .** Maximizing the  $\lambda$  is the simplest step in the algorithm. The first variation for  $\lambda$  is just the data residual,  $f(\mathbf{x}) - \sum_k u_k^{n+1}(\mathbf{x})$ . We use a standard gradient ascent with fixed time step  $\tau$  to achieve this maximization:

$$\lambda^{n+1}(\mathbf{x}) = \lambda^n(\mathbf{x}) + \tau \left( f(\mathbf{x}) - \sum_k u_k^{n+1}(\mathbf{x}) \right). \quad (22)$$

Note that the linearity of the Euler-Lagrange equation allows an impartial choice in which space to update the Lagrangian multiplier, either in the time domain or in the frequency domain. In our implementation, we perform our dual ascent update in the frequency domain, see Algorithm 1.

## 2.4 Optimization Algorithm

The full algorithm is detailed in Algorithm 1.

---

### Algorithm 1. Complete ADMM optimization of 2D-VMD

---

Initialize  $\{\hat{u}_k^0\}$ ,  $\{\hat{\omega}_k^0\}$ ,  $\hat{\lambda}^0$ ,  $n \leftarrow 0$

**repeat**

$n \leftarrow n + 1$

**for**  $k = 1 : K$  **do**

Create 2D Hilbert mask for Fourier multiplier

$$\mathcal{H}_k^{n+1}(\omega) \leftarrow 1 + \text{sgn}(\langle \omega_k^n, \omega \rangle) \quad (23)$$

Update  $\hat{u}_{AS,k}$  for all  $\omega$  such that  $\langle \omega_k^n, \omega \rangle \geq 0$ :

$$\hat{u}_{AS,k}^{n+1}(\omega) \leftarrow \mathcal{H}_k^{n+1}(\omega) \left[ \frac{\hat{f}(\omega) - \sum_{i < k} \hat{u}_i^{n+1}(\omega) - \sum_{i > k} \hat{u}_i^n(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + 2\alpha|\omega - \omega_k^n|^2} \right] \quad (24)$$

Update  $\omega_k$ :

$$\omega_k^{n+1} \leftarrow \frac{\int_{\mathbb{R}^2} \omega |\hat{u}_{AS,k}^{n+1}(\omega)|^2 d\omega}{\int_{\mathbb{R}^2} |\hat{u}_{AS,k}^{n+1}(\omega)|^2 d\omega} \quad (25)$$

Retrieve  $u_k$ :

$$u_k^{n+1}(\mathbf{x}) \leftarrow \Re\{\mathcal{F}^{-1} \hat{u}_{AS,k}^{n+1}(\omega)\} \quad (26)$$

**end for**

Dual ascent:

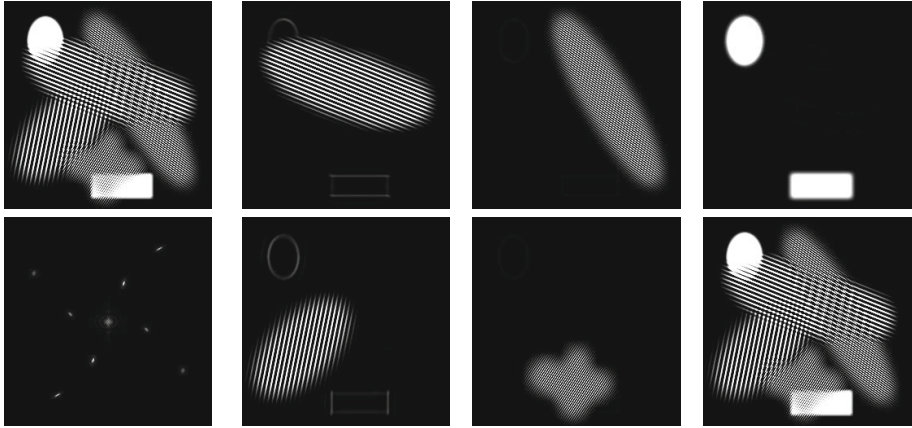
$$\hat{\lambda}^{n+1}(\omega) \leftarrow \hat{\lambda}^n(\omega) + \tau \left( \hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega) \right) \quad (27)$$

**until** convergence:  $\sum_k \|\hat{u}_k^{n+1} - \hat{u}_k^n\|_2^2 / \|\hat{u}_k^n\|_2^2 < K\epsilon$ .

---

### 3 Experiments and Results

We have implemented the 2D-VMD method in MATLAB<sup>®</sup><sup>2</sup>, and test the algorithm on both a synthetic and a real image.



**Fig. 1.** Results of 2D-VMD on  $f_{\text{Synth}}$ . **Top left:** The synthetic image. **Bottom left:** Spectrum. **Right:** 5 recovered modes, and **Bottom right:** their mode superposition.

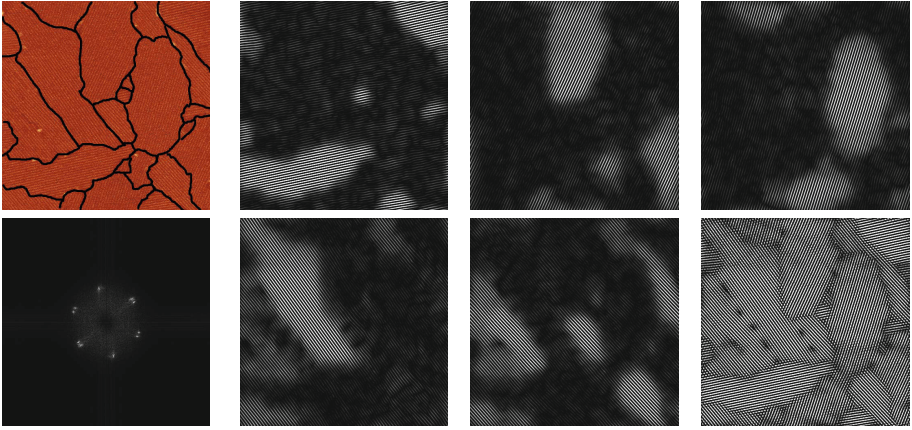
#### 3.1 Synthetic Image

The first, synthetic image is a composition of spatially overlapping basic shapes, more precisely six ellipses and a rectangle, with frequency patterns varying in both periodicity and direction, courtesy of J. Gilles [11]. The spectrum is ideal for segmentation due to modes being deliberately both well isolated and narrow-banded. The resolution of the synthetic image is 256x256 and the experiment was run with parameters  $\alpha = 1000$  and  $K = 5$ . This experiment converged in 520 iterations which took 45 seconds on a standard PC. The algorithm has no problems in accuracy nor timeliness in segmenting the image into its five constituent sub-images. The first is the DC component of the image - a solid ellipse and rectangle, while the four remaining decompositions in Fig. 1 show clear separation of the patterned ellipses. Due to the solid pieces having sharp edges, their spectra are not bandlimited and only smoothed versions are recovered. This is naturally paired with the two lower frequency modes picking up residual boundary artifacts of the DC component.

#### 3.2 Peptide $\beta$ -sheet Microscopy Image

The second test image is a scanning tunneling microscopy (STM) image of peptide  $\beta$ -sheets bonding on a graphite base, courtesy of the Weiss-group at the

<sup>2</sup> Code is available at <http://www.math.ucla.edu/~zosso/code.html>



**Fig. 2.** Results of 2D-VMD on peptide  $\beta$ -sheet image. **Top left:** Peptide  $\beta$ -sheet (with manual boundaries), and **Bottom left:** its power spectrum. **Right:** Five recovered constituent modes, and **Bottom right:** their mode superposition.

California NanoScience Institute (CNSI), [5]. The peptide sheets grow in regions of directional homogeneity and form natural spatial boundaries where the regions meet. It is important to scientists to have accurate segmentation for their dual interests in the homogeneous regions and their boundaries. Identifying regions of homogeneity allows for the subsequent study of isolated peptide sheets of one particular bonding class. For these types of scans, manually finding the boundaries is a tedious problem that demands the attention of a skilled scientist on a rote task. In addition to speed and automation, the proposed 2D-VMD is superior in accuracy to manual boundary identification due to regions potentially having very similar patterns, varying by only a few degrees, that are difficult to discern to even the trained eye. The success of the 1D-VMD algorithm in tone separation carries over to its 2D counterpart in separating patterns that are very close, yet distinct, in spectrum.

As a common pre-processing step in image analysis, here we apply a difference-of-Gaussians (DoG) band-pass filter to the image in order to remove both noise and the DC component. Subsequently, the 2D-VMD algorithm decomposes the piecewise homogeneous peptide sheet image into its five principle components with the purpose of segmentation. Fig. 2 illustrates these individual components, and then compares their superposition to the original peptide sheet with manual boundaries added. The resolution of this peptide image is 512x512 and the experiment was run with parameters  $\alpha = 5000$  and  $K = 5$ . This experiment converged in 210 iterations which took 140 seconds on a standard PC.

### 3.3 $\alpha_k$ Analysis and $\omega_k$ Initialization

An important degree of freedom in this algorithm is the initialization of the variables. While the  $u_k$  have a natural initialization of  $u_k \equiv 0$ , the  $\omega_k$  are

somewhat more sensitive. Though one could manually initialize the frequencies via the image’s power spectrum visualization with high accuracy, in the pursuit of full automation, we discuss the robustness of a fully unsupervised method of initialization. Qualitatively, a high  $\alpha$  leads to finer separation of constituent subsignals due to the Wiener filter being more narrowly concentrated around its center frequency. However, this same narrow filter, if centered away from a principle frequency, may fail to capture the relevant principle frequencies. Conversely, a low  $\alpha$  creates a wider filter, allowing the algorithm to “see and travel” to the correct frequencies, but yields worse separation. Given that we know the correct frequencies about which to initialize, a high  $\alpha$  will produce accurate results. If we do not know estimates of the principle frequencies of the subcomponents *a priori*, it seems that we are forced to use a lower  $\alpha$ , where the  $\omega_k$  gains freedom of mobility to the appropriate modes at the expense of proper separation. Instead of sacrificing accuracy for mobility, we keep both at the expense of computation time in the following way:

Initialize the  $\omega_k^0$  for  $k = 1, \dots, K$  randomly on any half-plane, such as  $\{\omega = (\omega_1, \omega_2) \mid \omega_1 \geq 0\}$ . Using a high  $\alpha$ , run the VMD algorithm and record the final values of  $\omega_k^N$ . Perform this repeatedly for a number of times and create a histogram of the convergent  $\omega_k^N$ , then observe the top  $K$  values. Individual iterations may converge to local minimizers, where qualitatively non-principle modes such as noise will be found, or where multiple  $\omega_k$  converge to the same principle mode while others are not picked up. The silver lining is that the non-principle convergent modes will be mostly uniformly spread across the spectrum, while the principle ones will show up with much higher consistency. This histogram of convergent modes captures the location of the consistent modes, from which we may get an excellent initialization for a final “clean” iteration. In the above peptide sheet image, we used 200 such iterations to unambiguously determine a proper initialization, though as few as 25 iterations were needed for the histogram to begin to resemble the power spectrum. Keeping this in mind, for practical applications, one can discount the ideal pursuit of automation, and do an accurate and simple graphical (semi-supervised) initialization of the frequencies to avoid multiple iterations in order to preserve timeliness.

## 4 Conclusions and Outlooks

In this paper, we have presented a 2D variational method for decomposing an image into an ensemble of band-limited intrinsic mode functions.

Our decomposition model solves the inverse problem as follows: decompose an image into a given number of modes such that each individual mode has limited bandwidth. We assess the mode’s bandwidth as the squared  $H^1$  norm of its directional 2D analytic signal with only half-space frequencies, shifted to baseband by mixing with a complex exponential of the current center frequency estimate. The modes are updated by simple Wiener filtering, directly in Fourier domain with a filter tuned to the current center frequency. Then the center frequencies are updated as the center of gravity of the mode’s power spectrum. We apply our model to both synthetic and real image data and can show successful decomposition.



The most important limitation of the proposed 2D-VMD is with boundary effects, and sudden signal onset in general. Despite the mostly successful decomposition, there is an issue of components' boundaries being overly smooth due to the narrow-band violation caused by discontinuous envelopes in such AM-FM signals, and a quadratic functional disfavoring and overly penalizing sharp amplitude changes, as with domain boundaries and piecewise regions. Conversely, this is also reflected by implicit periodicity assumptions when optimizing in Fourier domain. Another point is the required explicit selection of the number of active modes in the decomposition. A way to handle this issue is through the histogram method mentioned above. With many random initialization iterations, the dominant modes would show up with highest frequency, and this appropriate number of bins would become obvious by looking at an ordered distribution of the convergent frequency bins, exhibiting a gap between true frequencies and random noise. Rather than relying on iterations, both of these issues are being addressed in current work on an extended mathematical model.

We thank Paul S. Weiss and group members for sample images, and the collaborators of the W. M. Keck Foundation project "Leveraging sparsity" for inspiring discussion.

## References

1. Bertsekas, D.P.: Multiplier methods: A survey. *Automatica* 12(2), 133–145 (1976)
2. Bertsekas, D.P.: *Constrained optimization and Lagrange Multiplier methods*, vol. 1. Academic Press, Boston (1982)
3. Bülow, T., Sommer, G.: A Novel Approach to the 2D Analytic Signal. In: Solina, F., Leonardis, A. (eds.) CAIP 1999. LNCS, vol. 1689, pp. 25–32. Springer, Heidelberg (1999)
4. Candes, E.J., Donoho, D.L.: Curvelets: A Surprisingly Effective Nonadaptive Representation for Objects with Edges. In: *Curve and Surface Fitting*, pp. 105–120 (1999)
5. Claridge, S.A., Thomas, J.C., Silverman, M.A., Schwartz, J.J., Yang, Y., Wang, C., Weiss, P.S.: Differentiating Amino Acid Residues and Side Chain Orientations in Peptides Using Scanning Tunneling Microscopy. *Journal of the American Chemical Society* (November 2013)
6. Clausel, M., Oberlin, T., Perrier, V.: The Monogenic Synchrosqueezed Wavelet Transform: A tool for the Decomposition/Demodulation of AM-FM images (November 2012), <http://arxiv.org/abs/1211.5082>
7. Daubechies, I.: Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics* 41(7), 909–996 (1988)
8. Daubechies, I., Lu, J., Wu, H.T.: Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and Computational Harmonic Analysis* 30(2), 243–261 (2011)
9. Do, M., Vetterli, M.: Pyramidal directional filter banks and curvelets. In: *Proceedings of the 2001 International Conference on Image Processing*, vol. 2, pp. 158–161. IEEE (2001)
10. Dragomireskiy, K., Zosso, D.: Variational Mode Decomposition. *IEEE Transactions on Signal Processing* 62(3), 531–544 (2014)

11. Gilles, J.: Multiscale Texture Separation. *Multiscale Modeling & Simulation* 10(4), 1409–1427 (2012)
12. Gilles, J.: Empirical Wavelet Transform. *IEEE Transactions on Signal Processing* 61(16), 3999–4010 (2013)
13. Gilles, J., Tran, G., Osher, S.: 2D Empirical Transforms. *Wavelets, Ridgelets, and Curvelets Revisited*. *SIAM Journal on Imaging Sciences* 7(1), 157–186 (2014)
14. Guo, K., Labate, D.: Optimally Sparse Multidimensional Representation Using Shearlets. *SIAM Journal on Mathematical Analysis* 39(1), 298–318 (2007)
15. Hestenes, M.R.: Multiplier and Gradient Methods. *Journal of Optimization Theory and Applications* 4(5), 303–320 (1969)
16. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 454, 903–995 (1971)
17. Labate, D., Lim, W.Q., Kutyniok, G., Weiss, G.: Sparse Multidimensional Representation using Shearlets. In: Papadakis, M., Laine, A.F., Unser, M.A. (eds.) *Optics & Photonics*, pp. 1–9. International Society for Optics and Photonics (August 2005)
18. Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7), 674–693 (1989)
19. Nocedal, J., Wright, S.J.: *Numerical optimization*, 2nd edn. Springer, Berlin (2006)
20. Nunes, J., Bouaoune, Y., Delechelle, E., Niang, O., Bunel, P.: Image analysis by bidimensional empirical mode decomposition. *Image and Vision Computing* 21(12), 1019–1026 (2003)
21. Rockafellar, R.T.: A dual approach to solving nonlinear programming problems by unconstrained optimization. *Mathematical Programming* 5(1), 354–373 (1973)
22. Schmitt, J., Pustelnik, N., Borgnat, P., Flandrin, P.: 2D Hilbert-Huang Transform. In: *Proc. Int. Conf. Acoust., Speech Signal Process.* (2014)
23. Schmitt, J., Pustelnik, N., Borgnat, P., Flandrin, P., Condat, L.: 2-D Prony-Huang Transform: A New Tool for 2-D Spectral Analysis, p. 24 (April 2014), <http://arxiv.org/abs/1404.7680>
24. Lee, T.S.: Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(10), 959–971 (1996)

# Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme<sup>\*</sup>

Huiyi Hu<sup>1</sup>, Justin Sunu<sup>2</sup>, and Andrea L. Bertozzi<sup>1</sup>

<sup>1</sup> Department of Mathematics, University of California, Los Angeles  
huiyihu@math.ucla.edu, bertozzi@math.ucla.edu

<sup>2</sup> Institute of Mathematical Science, Claremont Graduate University  
justinsunu@gmail.com

**Abstract.** We focus on the multi-class segmentation problem using the piecewise constant Mumford-Shah model in a graph setting. After formulating a graph version of the Mumford-Shah energy, we propose an efficient algorithm called the MBO scheme using threshold dynamics. Theoretical analysis is developed and a Lyapunov functional is proven to decrease as the algorithm proceeds. Furthermore, to reduce the computational cost for large datasets, we incorporate the Nyström extension method which efficiently approximates eigenvectors of the graph Laplacian based on a small portion of the weight matrix. Finally, we implement the proposed method on the problem of chemical plume detection in hyper-spectral video data.

**Keywords:** graph, segmentation, Mumford-Shah, total variation, MBO, Nyström, hyper-spectral image.

## 1 Introduction

Multi-class segmentation has been studied as an important problem for many years in various areas, such as computer science and machine learning. For imagery data in particular, the Mumford-Shah model [18] is one of the most extensively used model in the past decade. This model approximates the true image by an optimal piecewise smooth function through solving a energy minimization problem. More detailed review of the work on Mumford-Shah model can be found in the references of [4]. A simplified version of Mumford-Shah is the piecewise constant model (also known as the “minimal partition problem”), which is widely used due to its reduced complexity compared to the original one. For a given contour  $\Phi$  which segments an image region  $\Omega$  into  $\hat{n}$  many disjoint sub-regions  $\Omega = \cup_{r=1}^{\hat{n}} \Omega_r$ , the *piecewise constant Mumford-Shah* energy is defined as:

$$E^{\text{MS}}(\Phi, \{c_r\}_{r=1}^{\hat{n}}) = |\Phi| + \lambda \sum_{r=1}^{\hat{n}} \int_{\Omega_r} (u_0 - c_r)^2, \quad (1)$$

---

<sup>\*</sup> This work was supported by NSF grants DMS-1118971, DMS-1045536, and DMS-1417674, UC Lab Fees Research grant 12-LR-236660, ONR grants N000141210040 and N000141210838, and the W. M. Keck Foundation.

where  $u_0$  is the observed image data,  $\{c_r\}_{r=1}^{\hat{n}}$  is a set of constant values, and  $|\Phi|$  denotes the length of the contour  $\Phi$ . By minimizing the energy  $E^{\text{MS}}$  over  $\Phi$  and  $\{c_r\}_{r=1}^{\hat{n}}$ , one obtains an optimal function which is constant within each sub-region to approximate  $u_0$ , along with a segmentation given by the optimal  $\Phi$ . In [5], a method of active contours without edges is proposed to solve for the two-class piecewise constant Mumford-Shah model ( $\hat{n} = 2$ ), using a level set method introduced in [19]. The work in [5] is further generalized to a multi-class scenario in [24]. The method developed in [5,24] is well known as the Chan-Vese model, which is a popular and representative method for image segmentation. The Chan-Vese method has been widely used due to the model's flexibility and the great success it achieves in performance.

In this work, we formulate the piecewise constant MS problem in a graph setting instead of a continuous one, and propose an efficient algorithm to solve it. Recently the authors of [2] introduced a binary semi-supervised segmentation method based on minimizing the Ginzburg-Landau functional on a graph. Inspired by [2], a collection of work has been done on graph-based high-dimensional data clustering problems posed as energy minimization problems, such as semi-supervised methods studied in [14,11] and an unsupervised network clustering method [13] known as modularity optimization. These methods make use of graph tools [6] and efficient graph algorithms, and our work pursues similar ideas. Note that unlike the Chan-Vese model which uses  $\log_2(\hat{n})$  many level set functions and binary representations to denote multiple classes, our model uses simplex constrained vectors for class assignments representation (details explained below).

To solve the multi-class piecewise constant MS variational problem in the graph setting, we propose an efficient algorithm using threshold dynamics. This algorithm is a variant of the one presented in the work of Merriman, Bence and Osher (MBO) [16,17], which was introduced to approximate the motion of an interface by its mean curvature in a continuous space. The idea of the MBO scheme is used on the continuous MS model [8,21] motivated by level set methods. The authors of [11,13,14] implement variants of the MBO scheme applied to segmentation problems in a graph setting. Rigorous proofs of convergence of the original MBO scheme in continuous setting can be found in [1,9] for the binary case, and [7] for the multi-class case. An analogous discussion in a graph setting is given in [23]. Inspired by the work of [7,23], we develop a Lyapunov functional for our proposed variant of the MBO algorithm, which approximates the graph MS energy. Theoretical analysis is given to prove that this Lyapunov energy decreases at each iteration of our algorithm, until it converges within finitely many steps.

In order to solve for each iteration of the MBO scheme, one needs to compute the weight matrix of the graph as well as the eigenvectors of the corresponding graph Laplacian. However, the computational cost can become prohibitive for large datasets. To reduce the numerical expenses, we implement the Nyström extension method [10] to approximately compute the eigenvectors, which only requires computing a small portion of the weigh matrix. Thus the proposed

algorithm is efficient even for large datasets, such as the hyper-spectral video data considered in this paper.

The proposed method can be implemented on general high-dimensional data clustering problems. However, in this work the numerical experiment is focused on the detection of chemical plumes in hyper-spectral video data. Detecting harmful gases and chemical plumes has wide applicability, such as in environmental study, defense and national security. However, the diffusive nature of plumes poses challenges and difficulties for the problem. One popular approach is to take advantage of hyper-spectral data, which provides much richer sensing information than ordinary visual images. The hyper-spectral images used in this paper were taken from video sequences captured by long wave infrared (LWIR) spectrometers at a scene where a collection of plume clouds is released. Over 100 spectral channels at each pixel of the scene are recorded, where each channel corresponds to a particular frequency in the spectrum ranging from 7,820 nm to 11,700 nm. The data is provided by the Applied Physics Laboratory at Johns Hopkins University, (see more details in [3]). Prior analysis of this dataset can be found in the works [12,15,20,22]. The authors of [15] implement a semi-supervised graph model using a similar MBO scheme. In the present paper, each pixel is considered as a node in a graph, upon which the proposed unsupervised segmentation algorithm is implemented. Competitive results are achieved as demonstrated below.

The rest of this paper is organized as follows. Section 2 introduces the graph formula for the multi-class piecewise constant Mumford-Shah model and relevant notations. In Section 3, the Mumford-Shah MBO scheme is presented as well as the theoretical analysis for a Lyapunov functional which is proven to decrease as the algorithm proceeds; techniques such as Nyström method are also introduced for the purpose of numerical efficiency. In Section 4, our algorithm is tested on the hyper-spectral video data for plume detection problem. The results are then presented and discussed.

## 2 Graph Mumford-Shah Model

Consider an  $N$ -node weighted graph  $(G, E)$ , where  $G = \{n_1, n_2, \dots, n_N\}$  is a node set and  $E = \{w_{ij}\}_{i=1}^N$  an edge set. Each node  $n_i$  corresponds to an agent in a given dataset, (such as a pixel in an image). The quantity  $w_{ij}$  represents the similarity between a pair of nodes  $n_i$  and  $n_j$ . Let  $\mathbf{W} = [w_{ij}]$  denote the graph's  $N \times N$  *weight matrix*, and in this work we assume  $\mathbf{W}$  is symmetric, i.e.  $w_{ij} = w_{ji}$ . In the case of hyper-spectral data, each node (pixel)  $n_i$  is associated with a  $d$ -dimensional feature vector (spectral channels). Let  $u_0 : G \rightarrow \mathbb{R}^d$  denote the raw hyper-spectral data, where  $u_0(n_i)$  represents the  $d$ -dimensional spectral channels of  $n_i$ . We use the following notation:

- The matrix  $\mathbf{L} := \mathbf{D} - \mathbf{W}$  is called the (un-normalized) *graph Laplacian* [6], where  $\mathbf{D}$  is a diagonal matrix with the  $i$ -th entry being  $\sum_{j=1}^N w_{ij}$ . For  $v : G \rightarrow \mathbb{R}$ , observe that

$$\langle v, \mathbf{L}v \rangle = \frac{1}{2} \sum_{i,j=1}^N w_{ij} (v(n_i) - v(n_j))^2. \quad (2)$$

– Graph function spaces for  $f = (f_1, f_2, \dots, f_{\hat{n}}) : G \rightarrow \mathbb{R}^{\hat{n}}$ :

$$\mathbb{K} := \left\{ f \mid f : G \rightarrow [0, 1]^{\hat{n}}, \sum_{r=1}^{\hat{n}} f_r(n_i) = 1 \right\},$$

which is a compact and convex set.

$$\mathbb{B} := \left\{ f \mid f : G \rightarrow \{0, 1\}^{\hat{n}}, \sum_{r=1}^{\hat{n}} f_r(n_i) = 1 \right\} \in \mathbb{K}.$$

This simplex constrained vector value taken by  $f \in \mathbb{B}$  indicates class assignment, i.e. if  $f_r(n_i) = 1$  for some  $r$ , then  $n_i$  belongs to the  $r$ -th class. Thus for each  $f \in \mathbb{B}$ , it corresponds to a partition of the graph  $G$  with at most  $\hat{n}$  classes. Let  $\langle f, \mathbf{L}f \rangle = \sum_{r=1}^{\hat{n}} \langle f_r, \mathbf{L}f_r \rangle$ .

– *Total Variation* (TV) for graph  $G$  is given as:

$$|f|_{TV} := \frac{1}{2} \sum_{i,j=1}^N w_{ij} |f(n_i) - f(n_j)|. \quad (3)$$

In this setting, we present a graph version of the multi-class *piecewise constant Mumford-Shah* energy functional:

$$MS(f, \{c_r\}_{r=1}^{\hat{n}}) := \frac{1}{2} |f|_{TV} + \lambda \sum_{r=1}^{\hat{n}} \langle \|u_0 - c_r\|^2, f_r \rangle, \quad (4)$$

where  $\{c_r\}_{r=1}^{\hat{n}} \subset \mathbb{R}^d$ ,  $\|u_0 - c_r\|^2$  denotes an  $N \times 1$  vector

$$\left( \|u_0(n_1) - c_r\|^2, \dots, \|u_0(n_N) - c_r\|^2 \right)^T,$$

and  $\langle \|u_0 - c_r\|^2, f_r \rangle = \sum_{i=1}^N f_r(n_i) \|u_0(n_i) - c_r\|^2$ . Note that when  $n_i$  and  $n_j$  belong to different classes, we have  $|f(n_i) - f(n_j)| = 2$ , which leads to the coefficient in front of the term  $\frac{1}{2} |f|_{TV}$ .

To see the connection between (4) and (1), one first observes that  $f_r$  is the characteristic function of the  $r$ -th class, and thus  $\langle \|u_0 - c_r\|^2, f_r \rangle$  is analogous to the term  $\int_{\Omega_r} (u_0 - c_r)^2$  in (1). Furthermore, the total variation of the characteristic function of a region gives the length of its boundary contour, and therefore  $|f|_{TV}$  is the graph analogy of  $|\Phi|$ .

In order to find a segmentation for  $G$ , we propose to solve the following minimization problem:

$$\min_{f \in \mathbb{B}, \{c_r\}_{r=1}^{\hat{n}} \subset \mathbb{R}^d} MS(f, \{c_r\}_{r=1}^{\hat{n}}). \quad (5)$$

The resulting minimizer  $f$  yields a partition of  $G$ .

One can observe that the optimal solution of (5) must satisfy:

$$c_r = \frac{\langle u_0, f_r \rangle}{\sum_{i=1}^N f_r(n_i)}, \tag{6}$$

if the  $r$ -th class is non-empty.

Note that for the minimization problem given in (5), it is essentially equivalent to the K-means method when  $\lambda$  goes to  $+\infty$ . When  $\lambda \rightarrow 0$ , the minimizer approaches a constant.

### 3 Mumford-Shah MBO and Lyapunov Functional

The authors of [16,17] introduced an efficient algorithm (known as the MBO scheme) to approximate the motion by mean curvature of an interface in a continuous space. The general procedure of the MBO scheme alternates between solving a linear heat equation and thresholding. One interpretation of the scheme is that it replaces the non-linear term of the Allen-Cahn equation with thresholding [8]. In this section we propose a variant of the original MBO scheme to approximately find the minimizer of the energy  $MS(f, \{c_r\}_{r=1}^{\hat{n}})$  presented in (4). Inspired by the work of [7,23], we write out a Lyapunov functional  $Y_\tau(f)$  for our algorithm and prove that it decreases at each iteration of the MBO scheme.

#### 3.1 Mumford-Shah MBO Scheme

We first introduce a “diffuse operator”  $\Gamma_\tau = e^{-\tau\mathbf{L}}$ , where  $\mathbf{L}$  is the graph Laplacian defined above and  $\tau$  is a time step size. The operator  $\Gamma_\tau$  is analogous to the diffuse operator  $e^{-\tau\Delta}$  of the heat equation in PDE (continuous space). It satisfies the following properties.

**Proposition 1.** *Firstly,  $\Gamma_\tau$  is strictly positive definite, i.e.  $\langle f, \Gamma_\tau f \rangle > 0$  for any  $f \in \mathbb{K}$ ,  $f \neq 0$ . Secondly,  $\Gamma_\tau$  conserves the mass, i.e.  $\langle \mathbf{1}, \Gamma_\tau f \rangle = \langle \mathbf{1}, f \rangle$ . At last, the quantity  $\frac{1}{2\tau} \langle \mathbf{1} - f, \Gamma_\tau f \rangle$  approximates  $\frac{1}{2} |f|_{TV}$ , for any  $f \in \mathbb{B}$ .*

*Proof.* Taylor expansion gives

$$e^{-\tau\mathbf{L}} = I - \tau\mathbf{L} + \frac{\tau^2}{2!}\mathbf{L}^2 - \frac{\tau^3}{3!}\mathbf{L}^3 + \dots$$

Suppose  $v$  is an eigenvector of  $\mathbf{L}$  associated with the eigenvalue  $\xi$ . One then has  $\Gamma_\tau v = e^{-\tau\xi}v \Rightarrow \langle v, \Gamma_\tau v \rangle = e^{-\tau\xi} \langle v, v \rangle > 0$ . Let the eigen-decomposition (with respect to  $\mathbf{L}$ ) for a non-zero  $f : G \rightarrow \mathbb{R}$  to be  $f = \sum_{i=1}^N a_i \phi_i$ , where  $\{\phi_i\}_{i=1}^N$  is a set of orthogonal eigenvectors of  $\mathbf{L}$  (note that  $\mathbf{L}$  is positive definite). Because  $\Gamma_\tau$  is a linear operator, one therefore has  $\langle f, \Gamma_\tau f \rangle = \sum_{i=1}^N a_i^2 \langle \phi_i, \Gamma_\tau \phi_i \rangle > 0$ .

For the second property,  $\mathbf{L}\mathbf{1} = 0 \Rightarrow \langle \mathbf{1}, \mathbf{L}^k f \rangle = 0$ , where  $\mathbf{1}$  is an  $N$ -dimensional vector with one at each entry. Therefore, the Taylor expansion of  $\Gamma_\tau$  gives  $\langle \mathbf{1}, \Gamma_\tau f \rangle = \langle \mathbf{1}, f \rangle$ .

At last,  $\Gamma_\tau \simeq I - \tau \mathbf{L} \Rightarrow \frac{1}{2\tau} \langle 1 - f, \Gamma_\tau f \rangle \simeq \frac{1}{2\tau} \langle 1 - f, f \rangle - \frac{1}{2} \langle 1, \mathbf{L} f \rangle + \frac{1}{2} \langle f, \mathbf{L} f \rangle$ . Particularly when  $f \in \mathbb{B}$ , we have  $\frac{1}{2\tau} \langle 1 - f, f \rangle = \frac{1}{2} \langle 1, \mathbf{L} f \rangle = 0$  and  $\frac{1}{2} \langle f, \mathbf{L} f \rangle = \frac{1}{2} |f|_{TV}$ . Hence  $\frac{1}{2\tau} \langle 1 - f, \Gamma_\tau f \rangle$  approximates  $\frac{1}{2} |f|_{TV}$  in  $\mathbb{B}$ . □

Note that the operator  $(I + \tau \mathbf{L})^{-1}$  also satisfies the above three properties, and can serve the same purpose as  $e^{-\tau \mathbf{L}}$ , as far as this paper concerns.

The proposed *Mumford-Shah MBO* scheme for the minimization problem (5) consists of alternating between the following three steps:

For a given  $f^k \in \mathbb{B}$  at the  $k$ -th iteration and  $c_r^k = \frac{\langle u_0, f_r^k \rangle}{\langle \mathbf{1}, f_r^k \rangle}$ ,

1. Compute

$$\hat{f} = \Gamma_\tau f^k - \tau \lambda (\|u_0 - c_1^k\|^2, \|u_0 - c_2^k\|^2, \dots, \|u_0 - c_{\hat{n}}^k\|^2), \tag{7}$$

2. (Thresholding)

$$f^{k+1}(n_i) = e_r, \quad r = \operatorname{argmax}_c \hat{f}_c(n_i)$$

for all  $i \in \{1, 2, \dots, N\}$ , where  $e_r$  is the  $r$ -th standard basis in  $\mathbb{R}^{\hat{n}}$ , i.e.  $f_r^{k+1}(n_i) = 1$  and  $f^{k+1} \in \mathbb{B}$ .

3. (Update  $c$ )

$$c_r^{k+1} = \frac{\langle u_0, f_r^{k+1} \rangle}{\langle \mathbf{1}, f_r^{k+1} \rangle}.$$

### 3.2 A Lyapunov Functional

We introduce a Lyapunov functional  $Y_\tau$  for the Mumford-Shah MBO scheme:

$$Y_\tau(f) := \frac{1}{2\tau} \langle 1 - f, \Gamma_\tau f \rangle + \lambda \sum_{r=1}^{\hat{n}} \langle \|u_0 - c_r\|^2, f_r \rangle, \quad \text{subject to } c_r = \frac{\langle u_0, f_r \rangle}{\langle \mathbf{1}, f_r \rangle}. \tag{8}$$

According to the third property of  $\Gamma_\tau$  in Proposition 1, energy  $Y_\tau(f)$  approximates  $MS(f, \{c_r\}_{r=1}^{\hat{n}})$  for  $f \in \mathbb{B}$  and  $c_r = \frac{\langle u_0, f_r \rangle}{\langle \mathbf{1}, f_r \rangle}$ . A similar functional for the graph total variation is shown and discussed in [23].

Pursuing similar ideas as in [7,23], we present the following analysis which consequently shows that the Mumford-Shah MBO scheme (with time step  $\tau$ ) decreases  $\Gamma_\tau$  and converges to a stationary state within a finite number of iterations.

First define

$$G_\tau(f, c) := \frac{1}{2\tau} \langle 1 - f, \Gamma_\tau f \rangle + \lambda \sum_{r=1}^{\hat{n}} \langle \|u_0 - c_r\|^2, f_r \rangle. \tag{9}$$

**Proposition 2.** *The functional  $G_\tau(\cdot, c)$  is strictly concave on  $\mathbb{K}$ , for any fixed  $\{c_r\}_{r=1}^{\hat{n}} \in \mathbb{R}^d$ .*



*Proof.* Take  $f, g \in \mathbb{K}$ ,  $\alpha \in (0, 1)$ . We have  $(1 - \alpha)f + \alpha g \in \mathbb{K}$ , because  $\mathbb{K}$  is a convex set.

$$\begin{aligned} & G_\tau((1 - \alpha)f + \alpha g, c) - (1 - \alpha)G_\tau(f, c) - \alpha G_\tau(g, c) \\ &= \frac{1}{2\tau}\alpha(1 - \alpha)\langle f - g, \Gamma_\tau(f - g) \rangle \geq 0. \end{aligned} \tag{10}$$

Equality only holds when  $f = g$ . Therefore,  $G_\tau(\cdot, c)$  is strictly concave on  $\mathbb{K}$ .  $\square$

Aside from the concavity of  $G_\tau$ , we observe that the first order variation of  $G_\tau(\cdot, c)$  is given as

$$\frac{\delta}{\delta f}G_\tau(f, c) = \frac{1}{2\tau}(1 - 2\Gamma_\tau f) + \lambda(\|u_0 - c_1\|^2, \|u_0 - c_2\|^2, \dots).$$

Note that since  $\langle \frac{\delta}{\delta f}G_\tau(f^k, c^k), f \rangle$  is linear, the Step 2 (thresholding) in the Mumford-Shah MBO scheme is equivalent to

$$f^{k+1} := \operatorname{argmin}_{f \in K} \langle \frac{\delta}{\delta f}G_\tau(f^k, c^k), f \rangle.$$

**Theorem 1.** *In the Mumford-Shah MBO scheme, the Lyapunov functional  $Y_\tau(f^{k+1})$  at the  $(k + 1)$ -th iteration is no greater than  $Y_\tau(f^k)$ . Equality only holds when  $f^k = f^{k+1}$ . Therefore, the scheme achieves a stationary point in  $\mathbb{B}$  within a finite number of iterations.*

*Proof.*

$$f^{k+1} := \operatorname{argmin}_{f \in K} \langle \frac{\delta}{\delta f}G_\tau(f^k, c^k), f \rangle \tag{11}$$

$\Rightarrow f^{k+1} \in \mathbb{B}$  (due to linearity) and

$$\begin{aligned} 0 &\geq \langle \frac{\delta}{\delta f}G_\tau(f^k, c^k), f^{k+1} - f^k \rangle \\ &\geq G_\tau(f^{k+1}, c^k) - G_\tau(f^k, c^k) \quad (\text{concavity}) \end{aligned} \tag{12}$$

$\Rightarrow G_\tau(f^{k+1}, c^k) \leq G_\tau(f^k, c^k) = Y_\tau(f^k)$ . Observe that  $c_r^{k+1} = \frac{\langle f_r^{k+1}, u_0 \rangle}{\langle f_r^{k+1}, 1 \rangle}$  is the minimizer of

$$\operatorname{argmin}_{\{c_r\}_{r=1}^{\hat{n}} \in \mathbb{R}^d} G_\tau(f^{k+1}, c)$$

$\Rightarrow G_\tau(f^{k+1}, c^{k+1}) \leq G_\tau(f^{k+1}, c^k) \leq Y_\tau(f^k)$ .

$\Rightarrow Y_\tau(f^{k+1}) \leq Y_\tau(f^k)$ . Therefore the Lyapunov functional  $Y_\tau$  is decreasing on the iterations of the Mumford-Shah MBO scheme, unless  $f^{k+1} = f^k$ . Since  $\mathbb{B}$  is a finite set, a stationary point can be achieved in a finite number of iterations.  $\square$

Minimizing the Lyapunov energy  $\Gamma_\tau$  is an approximation of the minimization problem in (5), and the proposed MBO scheme is proven to decrease  $\Gamma_\tau$ . Therefore, we expect the Mumford-Shah MBO scheme to approximately solve (5). In Section 3.3 and Section 3.4, we introduce techniques for computing the MBO iterations efficiently.

### 3.3 Eigen-Space Approximation

To solve for (7) in Step 1 of the Mumford-Shah MBO scheme, one needs to compute the operator  $\Gamma_\tau$ , which can be difficult especially for large datasets. For the purpose of efficiency, we numerically solve for (7) by using a small number of the leading eigenvectors of  $\mathbf{L}$  (which correspond to the smallest eigenvalues), and project  $f^k$  onto the eigen-space spanned from the eigenvectors. By approximating the operator  $\mathbf{L}$  with the leading eigenvectors, one can compute (7) efficiently. We use this approximation because in graph clustering methods, researchers have been using a small portion of the leading eigenvectors of a graph Laplacian to extract structural information of the graph.

Let  $\{\phi_m\}_{m=1}^M$  denote the first  $M$  (orthogonal) leading eigenvectors of  $\mathbf{L}$ , and  $\{\xi_m\}_{m=1}^M$  the corresponding eigenvalues. Assume  $f^k = \sum_{m=1}^M \phi_m a^m$ , where  $a^m$  is a  $1 \times \hat{n}$  vector, with the  $r$ -th entry  $a_r^m = \langle f_r^k, \phi_m \rangle$ . Thus  $\hat{f}$  can be approximately computed as:

$$\hat{f} = \sum_{m=1}^M e^{-\tau \xi_m} \phi_m a^m - \tau \lambda (\|u_0 - c_1^k\|^2, \|u_0 - c_2^k\|^2, \dots, \|u_0 - c_{\hat{n}}^k\|^2). \quad (13)$$

The Mumford-Shah MBO algorithm with the above eigen-space approximation is summarized in Algorithm 1. After the eigenvectors are obtained, each iteration of the MBO scheme is of time complexity  $O(N)$ . Empirically, the algorithm converges after a small number of iterations. Note that the iterations stop when a *purity* score between the partitions from two consecutive iterations is greater than 99.9%. The purity score, as used in [13], measures how “similar” two partitions are. Intuitively, it can be viewed as the fraction of nodes of one partition that have been assigned to the correct class with respect to the other partition.

---

#### Algorithm 1. Mumford-Shah MBO algorithm

---

Input:  $f^0$ ,  $u_0$ ,  $\{(\phi_m, \xi_m)\}_{m=1}^M$ ,  $\tau$ ,  $\lambda$ ,  $\hat{n}$ ,  $k = 0$ .

**while** (purity( $f^k, f^{k+1}$ ) < 99.9%) **do**

$$- c_r = \frac{\langle u_0, f_r^k \rangle}{\sum_{i=1}^{\hat{n}} f_r^k(n_i)}.$$

$$- a_r^m = \langle f_r^k, \phi_m \rangle.$$

$$- \hat{f} = \sum_{m=1}^M e^{-\tau \xi_m} \phi_m a^m - \tau \lambda (\|u_0 - c_1\|^2, \|u_0 - c_2\|^2, \dots, \|u_0 - c_{\hat{n}}\|^2).$$

$$- f^{k+1}(n_i) = \mathbf{e}_r, \text{ where } r = \operatorname{argmax}_c \hat{f}_c(n_i).$$

$$- k \leftarrow k + 1.$$

**end while**

---

### 3.4 Nyström Method

The Nyström extension [10] is a matrix completion method which has been used to efficiently compute a small portion of the eigenvectors of the graph Laplacian for segmentation problems [2,14,11]. In our proposed scheme, leading

eigenvectors of  $\mathbf{L}$  are required, which can require massive computational time and memory. For large graphs such as the ones induced from images, the explicit form of the weight matrix  $\mathbf{W}$  and therefore  $\mathbf{L}$  is difficult to obtain ( $O(N^2)$  time complexity). Hence, we expect to use the Nyström method to approximately compute the eigenvectors for our algorithm.

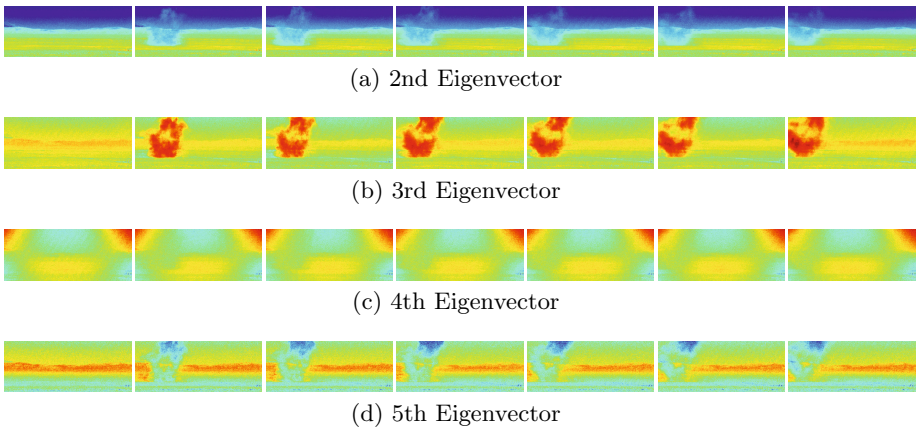
Basically, the Nyström method randomly samples a very small number ( $M$ ) of rows of  $\mathbf{W}$ . Based on matrix completion and properties of eigenvectors, it approximately obtains  $M$  eigenvectors of the symmetric normalized graph Laplacian  $\mathbf{L}_s = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$  without computing the whole weight matrix. Detailed descriptions of the Nyström method can be found in [2,14].

Note that our previous analysis only applies to  $\mathbf{L}$  rather than  $\mathbf{L}_s$ , and the Nyström method can not be trivially formularized for  $\mathbf{L}$ . Therefore this question remains to be studied. However, the normalized Laplacian  $\mathbf{L}_s$  has many similar features compared to  $\mathbf{L}$ , and it has been used in place of  $\mathbf{L}$  in many segmentation problems. In the numerical results shown below, the eigenvectors of  $\mathbf{L}_s$  computed via Nyström perform well empirically.

One can also implement other efficient methods to compute the eigenvectors for the Mumford-Shah MBO algorithm.

## 4 Numerical Results

The hyper-spectral images tested in this work are taken from the video recording of the release of chemical plumes at the Dugway Proving Ground, captured by long wave infrared (LWIR) spectrometers. The data is provided by the Applied Physics Laboratory at Johns Hopkins University. A detailed description of this dataset can be found in [3]. We take seven frames from a plume video sequence in which each frame is composed of  $128 \times 320$  pixels. We use a background frame and the frames numbered 72 through 77 containing the plume. Each pixel



**Fig. 1.** The leading eigenvectors of the normalized graph Laplacian computed via the Nyström method

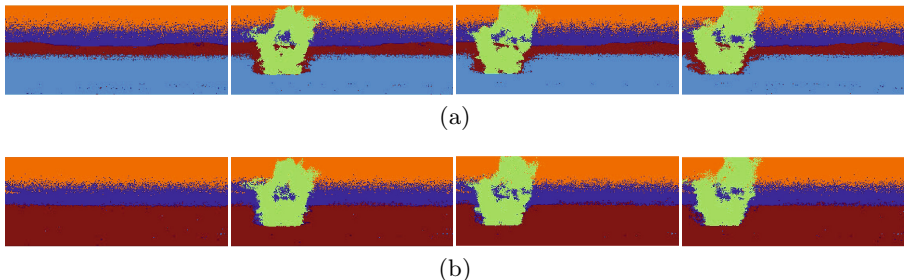
has 129 spectral channels corresponding to a particular frequency in the EM spectrum ranging from 7,820 nm to 11,700 nm. Thus, the graph we construct from these seven frames is of size  $7 \times 128 \times 320$  with each node  $n_i$  corresponding to a pixel with a 129-dimensional spectral signature  $v_i$ . The metric for computing the weight matrix is given as:

$$w_{ij} = \exp\left\{-\frac{\left(1 - \frac{\langle v_i, v_j \rangle}{\|v_i\| \|v_j\|}\right)^2}{2\sigma^2}\right\},$$

where  $\sigma = 0.01$  is chosen empirically for the best performance. Note that in this experiment  $\sigma$  is a robust parameter which gives decent results for a wide range of values ( $0.001 < \sigma < 10$ ).

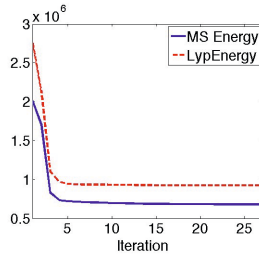
The goal is to segment the image and identify the “plume cloud” from the background components (sky, mountain, grass), without any ground truth. As described in the previous section,  $M = 100$  eigenvectors of the normalized graph Laplacian ( $\mathbf{L}_s$ ) are computed via the Nyström method. The computational time using Nyström is less than a minute on a 2.8GHz machine with Intel Core 2Duo. The visualization of the first five eigenvectors (associated with the smallest eigenvalues) are given in Figure 1 for the first four frames, (the first eigenvector is not shown because it is close to a constant vector).

We implement the Mumford-Shah MBO scheme using the eigenvectors on this seven frames of plume images, with  $\tau = 0.15$ ,  $\lambda = 150$  and  $\hat{n} = 5$ . The test is run for 20 times with different uniformly random initialization, and the segmentation results are shown in Figure 2. Note that depending on the initialization, the algorithm can converge to different local minimum, which is common for most non-convex variational methods. The result in (a) occurred five times among the 20 runs, and (b) for twice. The outcomes of other runs merge either the upper or the lower part of the plume with the background. The segmentation outcome shown in (a) gives higher energy than that in (b). Among the 20 runs, the lowest energy is achieved by a segmentation similar to (a), but with the lower part of the plume merged with the background. It may suggest that the global minimum of the proposed energy does not necessarily give a desired segmentation.



**Fig. 2.** The segmentation results obtained by the Mumford-Shah MBO scheme, on a background frame plus the frames 72-77. Shown in (a) and (b) are segmentation outcomes obtained with different initializations. The visualization of the segmentations only includes the first four frames.

Notice that in Figure 2 (b), even though there actually exist five classes, only four major classes can be perceived, while the other one contains only a very small amount of pixels. By allowing  $\hat{n} = 5$  instead of 4, it helps to reduce the influence of a few abnormal pixels. The computational time for each iteration is about 2-3 seconds on a 1.7GHz machine with Intel Core i5. The number of iterations is around 20-40.

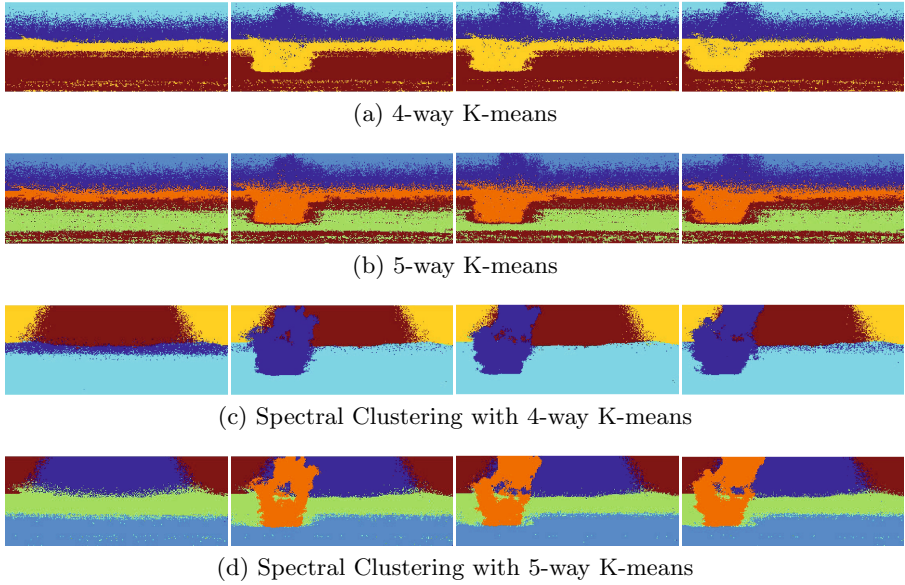


**Fig. 3.** Energy  $MS(f)$  (blue, solid line) and  $Y_\tau(f)$  (red, dash line) at each iteration from the same test as shown in Figure 2 (a)

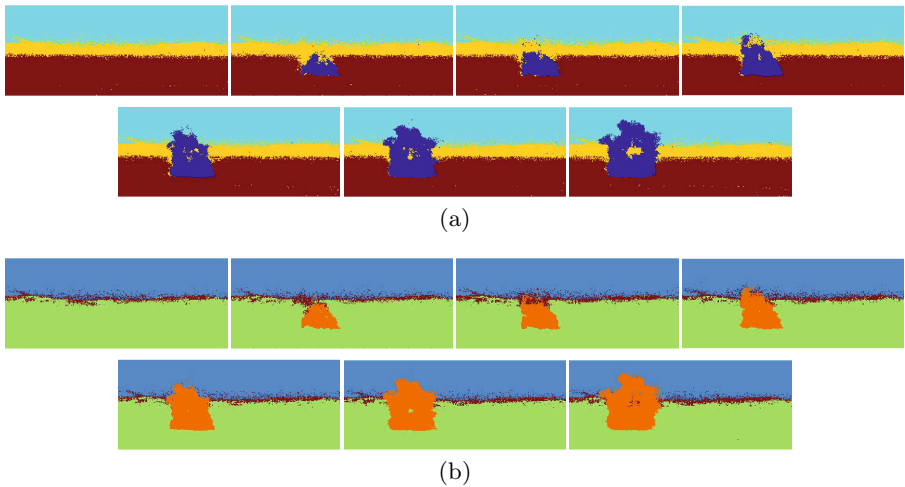
Figure 3 demonstrates a plot of the  $MS(f)$  and  $Y_\tau(f)$  energies at each iteration from the same test as the one shown in Figure 2 (a). The Lyapunov energy  $Y_\tau(f)$  (red, dash line) is non-increasing, as proven in Theorem 1. Note that all the energies are computed approximately using eigenvectors.

As a comparison, the segmentation results using K-means and spectral clustering are shown in Figure 4. The K-means method is performed directly on the raw image data ( $7 \times 128 \times 320$  by 129). As shown in (a) and (b), the results obtained by K-means fail to capture the plume; the segmentations on the background are also very fuzzy. For the spectral clustering method, a 4-way (or 5-way) K-means is implemented on the four (or five) leading eigenvectors of the normalized graph Laplacian (computed via Nyström). As shown in (c) and (d), the resulting segmentations divide the sky region into two undesirable components. Unlike the segmentation in Figure 2 (a) where the mountain component (red, the third in the background) has a well defined outline, the spectral clustering results do not provide clear boundaries. Our approach performs better than other unsupervised clustering results on this dataset [12,20].

Another example of the plume data is shown in Figure 5, where the 67th to 72nd frames (instead of the 72nd to 77th) are taken along with the background frame as the test data. The test is run 20 times using different uniformly random initialization, where  $\tau = 0.15$ ,  $\lambda = 150$  and  $\hat{n} = 5$ . The result in Figure 5 (a) occurred 11 times among the 20 runs, and (b) for 5 times. The outcomes from the other 4 runs segment the background into three components as in (a), but merge the plume with the center component. The segmentation result shown in (a) gives the lowest energy among all the outcomes. The visualization includes all seven frames since the plume is small in the first several frames.



**Fig. 4.** K-means and spectral clustering segmentation results. The visualization of the segmentations only includes the first four frames.



**Fig. 5.** The segmentation results obtained by the Mumford-Shah MBO scheme, on a background frame plus the frames 67-72. Shown in (a) and (b) are segmentation outcomes obtained with different initializations.

## 5 Conclusion

In this paper we present a graph framework for the multi-class piecewise constant Mumford-Shah model using a simplex constrained representation. Based on the graph model, we propose an efficient threshold dynamics algorithm, the Mumford-Shah MBO scheme for solving the minimization problem. Theoretical analysis is developed to show that the MBO iteration decreases a Lyapunov energy that approximates the MS functional. Furthermore, in order to reduce the computational cost for large datasets, we incorporate the Nyström extension method to approximately compute a small portion of the eigenvectors of the normalized graph Laplacian, which does not require computing the whole weight matrix of the graph. After obtaining the eigenvectors, each iteration of the Mumford-Shah MBO scheme is of time complexity  $O(n)$ . The number of iterations for convergence is small empirically.

The proposed method can be applied to general high-dimensional data segmentation problems. In this work we focus on the segmentation of hyper-spectral video data. Numerical experiments are performed on a collection of hyper-spectral images taken from a video for plume detection; using our proposed method, competitive results are achieved. However, there are still open questions to be answered. For example, the Nyström method can only compute eigenvectors for the normalized Laplacian, while the theoretical analysis for the Lyapunov functional only applies to the un-normalized graph Laplacian. This issue remains to be studied. Note that the graph constructed in this paper does not include the spacial information of the pixels, but only the spectral information. One can certainly build a graph incorporating the location of each pixel as well, to generate a non-local means graph as discussed in [2].

**Acknowledgments.** We thank Dr. Luminita A. Vese for useful comments.

## References

1. Barles, G., Georgelin, C.: A Simple Proof of Convergence for an Approximation Scheme for Computing Motions by Mean Curvature. *SIAM J. Numer. Anal.* 32(2), 484–500 (1995)
2. Bertozzi, A.L., Flenner, A.: Diffuse Interface Models on Graphs for Classification of High Dimensional Data. *Multiscale Modeling Sim.* 10(3), 1090–1118 (2012)
3. Broadwater, J.B., Limsui, D., Carr, A.K.: A Primer for Chemical Plume Detection using LWIR Sensors. Tech. Rep., National Security Technology Department (2011)
4. 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)
5. Chan, T., Vese, L.A.: Active Contours without Edges. *IEEE Trans. Image Process.* 10, 266–277 (2001)
6. Chung, F.R.K.: Spectral Graph Theory. *CBMS Reg. Conf. Ser. Math.*, vol. 92. AMS (1997)
7. Esedoglu, S., Otto, F.: Threshold Dynamics for Networks with Arbitrary Surface Tensions (2013) (submitted)

8. Esedoglu, S., Tsai, Y.R.: Threshold Dynamics for the Piecewise Constant Mumford-Shah Functional. *J. Comput. Phys.* 211, 367–384 (2006)
9. Evans, L.C.: Convergence of an Algorithm for Mean Curvature Motion. *Indiana Univ. Math. J.* 42, 553–557 (1993)
10. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral Grouping using the Nystrom Method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(2), 214–225 (2004)
11. Garcia-Cardona, C., Merkurjev, E., Bertozzi, A. L., Flenner, A., Percus, A.: Fast Multiclass Segmentation using Diffuse Interface Methods on Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* (2014)
12. Gerhart, T., Sunu, J., Lieu, L., Merkurjev, E., Chang, J.-M., Gilles, J., Bertozzi, A.L.: Detection and Tracking of Gas Plumes in LWIR Hyperspectral Video Sequence Data. In: *SPIE Conference on Defense, Security, and Sensing* (2013)
13. 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)
14. Merkurjev, E., Kostic, T., Bertozzi, A.L.: An MBO Scheme on Graphs for Segmentation and Image Processing. *SIAM J. Imaging Sci.* 6, 1903–1930 (2013)
15. Merkurjev, E., Sunu, J., Bertozzi, A.L.: Graph MBO Method for Multiclass Segmentation of Hyper Spectral Stand-off Detection Video. In: *Proceedings of the International Conference on Image Processing* (accepted, 2014)
16. Merriman, B., Bence, J.K., Osher, S.J.: Diffusion Generated Motion by Mean Curvature. In: *Proceedings of the Computational Cristal Growers Workshop*, pp. 73–83. AMS, Providence (1992)
17. Merriman, B., Bence, J.K., Osher, S.J.: Motion of Multiple Junctions: A Level Set Approach. *J. Comput. Phys.* 112, 334–363 (1994)
18. Mumford, D., Shah, J.: Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems. *Comm. Pure Appl. Math.* 42, 577–685 (1989)
19. 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)
20. Sunu, J., Chang, J.-M., Bertozzi, A.L.: Simultaneous Spectral Analysis of Multiple Video Sequence Data for LWIR Gas Plumes. In: *SPIE Conference on Defense, Security, and Sensing* (2014)
21. Tai, X.C., Christiansen, O., Lin, P., Skjælaaen, I.: Image Segmentation using Some Piecewise Constant Level Set Methods with MBO Type of Projection. *International Journal of Computer Vision* 73(1), 61–76 (2007)
22. Tochon, G., Chanussot, J., Gilles, J., Dalla Mura, M., Chang, J.-M., Bertozzi, A.L.: Gas Plume Detection and Tracking in Hyperspectral Video Sequences using Binary Partition Trees (preprint, 2014)
23. van Gennip, Y., Guillen, N., Osting, B., Bertozzi, A.L.: Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs. *Milan J. Math.* 82(1), 3–65 (2014)
24. Vese, L.A., Chan, T.F.: A New Multiphase Level Set Framework for Image Segmentation via the Mumford and Shahs model. *International Journal of Computer Vision* 50, 271–293 (2002)



# A Novel Active Contour Model for Texture Segmentation

Aditya Tatu and Sumukh Bansal

DA-IICT - Gandhinagar, India

{aditya\_tatu,sumukh\_bansal}@daiict.ac.in

**Abstract.** Texture is intuitively defined as a repeated arrangement of a basic pattern or object in an image. There is no mathematical definition of a texture though. The human visual system is able to identify and segment different textures in a given image. Automating this task for a computer is far from trivial.

There are three major components of any texture segmentation algorithm: (a) The features used to represent a texture, (b) the metric induced on this representation space and (c) the clustering algorithm that runs over these features in order to segment a given image into different textures.

In this paper, we propose an active contour based novel unsupervised algorithm for texture segmentation. We use intensity covariance matrices of regions as the defining feature of textures and find regions that have the most inter-region dissimilar covariance matrices using active contours. Since covariance matrices are symmetric positive definite, we use geodesic distance defined on the manifold of symmetric positive definite matrices  $PD(n)$  as a measure of dissimilarity between such matrices. Using recent convexification methods, we are able to compute a global maxima of the cost function. We demonstrate performance of our algorithm on both artificial and real texture images.

## 1 Introduction

Texture is intuitively defined as a repeated arrangement of a basic pattern or object in an image. There is no universal mathematical definition of a texture though. The human visual system is able to identify and segment different textures in a given image without much effort. Automating this task for a computer, though, is far from trivial.

Apart from being a tough academic problem, texture segmentation has several applications. Texture segmentation has been applied to detect landscape changes from aerial photographs in remote sensing and GIS [40], content based image retrieval [16] and diagnosing ultrasound images [27] and others.

There are three major components in any texture segmentation algorithm: (a) The model or features that define or characterize a texture, (b) the metric defined on this representation space, and (c) the clustering algorithm that runs over these features in order to segment a given image into different textures.

There are two approaches of modeling a texture: Structural and Statistical. The structural approach describes a texture as a specific spatial arrangement of a primitive element. Voronoi polynomials are used to specify the spatial arrangement of these primitive elements [34,33]. The statistical approach describes a texture using features that encode the regularity in arrangement of gray-levels in an image. Examples of features used are

responses to Gabor filters [15], graylevel co-occurrence matrices [42,20], Wavelet coefficients [12], human visual perception based Tamura features [32], Laws energy measures [25], Local binary patterns[28] and Covariance matrices of features [35,14]. In [23], the authors compare performance of some of the above mentioned features for the specific goal of image retrieval. In fact, Zhu, Wu & Mumford [41] propose a mechanism of choosing an optimal set of features for texture modeling from a given general filter bank. Markov random fields[10], Fractal dimensions[8] and the space of oscillating functions [36] have also been used to model textures.

Various metrics have been used to quantify dissimilarity of features: Euclidean, Chi-squared, Kullback-Leibler & its symmetrized version [38], manifold distance on the Gabor feature space [9] and others.  $k$ -NN, Bayesian inference,  $c$ -means, alongwith active contours algorithms are some of the methods used for clustering/segmentating texture areas in the image with similar features.

In this paper, we use intensity covariance matrices over a region as the texture feature. Since these are symmetric positive definite matrices which form a manifold, denoted by  $PD(n)$ , it is natural to use the intrinsic manifold distance as a measure of feature dissimilarity. Using a novel active contours method, we propose to find the background/foreground texture regions in a given image by maximizing the geodesic distance between the interior and exterior covariance matrices. This is the main contribution of our paper.

In the next subsection we list out some existing texture segmentation approaches using active contours model.

## 1.1 Related Work

Sagiv, Sochen & Zeevi [9] generalize both, geodesic active contours and Chan & Vese active contours, to work on a Gabor feature space. The Gabor feature space is a parametric 2-D manifold embedded in  $\mathbb{R}^7$  whose natural metric is used to define an edge detector function for geodesic active contours, and to define the intra-region variance in case of the Chan & Vese active contours. In [31], the authors use Chan & Vese active contours on Local Binary Pattern features for texture segmentation.

In [30], the authors propose a Chan & Vese active contour model on probability distribution of the structure tensor of the image as a feature. The closest approach to our algorithm is by Houhou et. al.[21], where the authors find a contour that maximizes the KL-divergence based metric on probability distribution of a feature for points lying inside the contour and outside the contour. The feature used is based on principal curvatures of the intensity image considered as a 2-D manifold embedded in  $\mathbb{R}^3$ . In particular, the cost function for a curve  $\Omega$  is defined as

$$KL(p_{in}(\Omega), p_{out}(\Omega)) = \int_{\mathbb{R}^+} (p_{in}(\kappa_t, \Omega) - p_{out}(\kappa_t, \Omega)) \cdot (\log p_{in}(\kappa_t, \Omega) - \log p_{out}(\kappa_t, \Omega)) d\kappa_t$$

where  $p_{in}(\Omega), p_{out}(\Omega)$  is the probability distribution of the feature  $\kappa$  inside and outside the closed contour  $\Omega$  respectively. Gaussian distribution is assumed as the model for the probability distribution of the feature both inside as well as outside the contour.

Recent modifications of this algorithm, as in [22] uses a feature based on determinant of the metric of the manifold (which is a semi-local representation based on the Beltrami framework), while the authors in [11] use the Cauchy-Schwartz distance (instead of the KL-divergence) on the probability density of a feature similar to  $\kappa$  used above.

In our approach, instead of using some scalar feature to represent texture, we compute a contour that maximizes the geodesic distance between the interior and exterior intensity covariance matrix of the contour. It can be seen that the maximization process has to be carried out over the manifold of symmetric positive definite matrices, making it fundamentally different from the approaches in [21,22]. Moreover, we can easily extend this approach to covariance matrices of any other texture feature one may want to use as we show for Gabor features in section3.

The paper is organized as follows: In next section we provide a brief review of active contour models for image segmentation. In Section 2, we describe our active contour model based on geodesic distance between the interior and exterior covariance matrices of a contour. We give our experimental results in Section 3 followed by conclusions and future scope.

### 1.2 Active Contours and Level Sets

Active contour methods can be categorized into two: edge-based and region-based. The classical active contours by Kass, Witkin & Terzopoulos [24] and the newer Geodesic active contours[4] fall into the first category in which region boundaries are detected by iteratively minimizing a cost function that encourages an initial curve to latch on to intensity edges via a curve evolution equation. An efficient numerical scheme to implement these curve evolution equations is via level sets [1]. Since texture boundaries do not typically correspond to intensity edges, we base our method on the second category - region based active contours, which we review next.

In the region-based approach, an energy functional based on regional similarity properties of an object, rather than its edges (image gradient) is minimized. A successful example of such a method is the Active Contours Without Edges (ACWE) model proposed by Chan & Vese [7]. It tries to approximate the given image  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  by a function that can take only two values. Such a function can be represented by the values  $\mu_1, \mu_2$  that it takes and the boundary  $C$  between regions taking the two values. The energy function after including regularization terms is given by

$$\begin{aligned}
 F(\mu_1, \mu_2, C) = & \mu.Length(C) + \nu.Area(int(C)) + \lambda_1 \int_{int(C)} (I(x,y) - \mu_1)^2 dx dy \\
 & + \lambda_2 \int_{ext(C)} (I(x,y) - \mu_2)^2 dx dy, \quad (1)
 \end{aligned}$$

where  $\mu \geq 0, \nu \geq 0, \lambda_1, \lambda_2 > 0$  are fixed scalar parameters, and  $int(C)$  and  $ext(C)$  denotes the interior and exterior region of the curve  $C$  in  $\Omega$  respectively. Re-writing the above functional using a level set function[29]  $\phi$  with the following convention:

$$\begin{aligned}
 int(C) &= \{(x,y) \in \Omega \mid \phi(x,y) < 0\} \\
 ext(C) &= \{(x,y) \in \Omega \mid \phi(x,y) > 0\}, \quad (2)
 \end{aligned}$$

the Heaviside function  $H$ , and the Dirac Delta function  $\delta$ , we obtain the following energy functional:

$$\begin{aligned}
 F(\mu_1, \mu_2, \phi) &= \int_{\Omega} \mu \delta(\phi(x,y)) |\nabla \phi(x,y)| \\
 &+ \nu(1 - H(\phi(x,y))) + \lambda_1(I(x,y) - \mu_1)^2(1 - H(\phi(x,y))) \\
 &+ \lambda_2(I(x,y) - \mu_2)^2 H(\phi(x,y)) \quad dx dy.
 \end{aligned} \tag{3}$$

For a fixed  $C$  (and therefore a fixed  $\phi$ ), the minimizers for  $\mu_1, \mu_2$  can easily be seen to be the mean gray value in  $int(C)$  and  $ext(C)$  respectively. The minimization in terms of  $\phi$ , keeping  $\mu_1, \mu_2$  fixed (and using smooth approximations  $H_{\epsilon}, \delta_{\epsilon}$ , of  $H, \delta$  as given in [7]) is given by the following gradient descent based level set evolution equation:

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left[ \mu \operatorname{div} \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) + \nu + \lambda_1(I - \mu_1)^2 - \lambda_2(I - \mu_2)^2 \right]. \tag{4}$$

A drawback of such gradient descent schemes is that they can get stuck in a local minima and hence these methods heavily rely on a good user initialization. This is essentially the result of non-convex energy functionals. To alleviate this problem, convexification of active contour energies was proposed in [6]. With  $\mu = 1, \nu = 0, \lambda_1 = \lambda_2 = \lambda$ , following the presentation in [3], we let  $r(x, \mu_1, \mu_2) := \lambda \left[ (I - \mu_2)^2 - (I - \mu_1)^2 \right]$ . Then, the level set evolution below also provides the same steady state solution as by the gradient descent equation(4) above.

$$\frac{\partial \phi}{\partial t} = \operatorname{div} \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) - \lambda r(x, \mu_1, \mu_2) \tag{5}$$

This is a gradient descent for the convex energy:

$$E(u, \mu_1, \mu_2, \lambda) = \int_{\Omega} \|\nabla u\| + \lambda r(x, \mu_1, \mu_2) u \, dx, \tag{6}$$

where we replace  $\phi$  with  $u$  since it need not be the usual level set function. This functional is one-homogeneous in  $u$ , therefore the minimization is carried out over  $0 \leq u \leq 1$  in order to get a stationary solution. Thus the task is to solve:

$$\min_{0 \leq u \leq 1} E(u, \mu_1, \mu_2, \lambda) \tag{7}$$

Several numerical schemes have been proposed for fast minimization of such convex energies, of which we mention two: a gradient projection method [5] and the Split-Bregman method [19], with the latter solving a regularized version of the original problem[18]. We choose to use the Split-Bregman<sup>1</sup> method. It introduces an auxiliary variable  $d$  for  $\nabla u$ , which converts the constrained minimization (constraint:  $d = \nabla u$ ) to the following unconstrained problem:

$$(u^{k+1}, d^{k+1}) = \arg \min_{0 \leq u \leq 1, d} \int_{\Omega} \|d\| + \lambda r(x, \mu_1, \mu_2) u + \frac{\mu}{2} \|d - \nabla u - b^k\|^2 \, dx, \tag{8}$$

$$b^{k+1} = b^k + \nabla u^{k+1} - d^k. \tag{9}$$

---

<sup>1</sup> The Split Bregman method has been shown to be equivalent to the Augmented Lagrangian method[39].

The Euler-Lagrange equation for the first minimization problem above is

$$\Delta u = \frac{\lambda}{\mu} r + \nabla \cdot (d^k - b^k) \tag{10}$$

Gauss-Seidel procedure yields the solution:

$$\alpha_{i,j} = d_{i-1,j}^x - d_{i,j}^x - b_{i-1,j}^x + b_{i,j}^x + d_{i,j-1}^y - d_{i,j}^y - b_{i,j-1}^y + b_{i,j}^y, \tag{11}$$

$$\beta_{i,j} = \frac{1}{4} \left( u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - \frac{\lambda}{\mu} r + \alpha_{i,j} \right), \tag{12}$$

$$u_{i,j}^{k+1} = \max\{\min\{\beta_{i,j}, 1\}, 0\}, \tag{13}$$

while minimization with respect to  $d$  in Equation(8), is obtained by soft-wavelet thresholding[37]:

$$d^{k+1} = shrink(b^k + \nabla u^{k+1}, \mu), \tag{14}$$

where  $shrink(p, \mu) = \max\{\|p\| - \mu, 0\} \frac{p}{\|p\|}$ . This process is iterated till  $\|u^{k+1} - u^k\| > \epsilon$ .

Our active contour model for (two-class) texture segmentation is based on finding disjoint regions of the image which have as different texture features as possible. In the next section, we describe our energy functional, derive the gradient descent equation, and convexify the energy as shown above in order to make our model independant of the initialization.

## 2 Proposed Active Contour Model for Texture Segmentation

In what follows, we assume familiarity with concepts from differential geometry like geodesic distance, Riemannian Exponential and Riemannian Logarithm maps. A thorough introduction to these concepts can be found in the books [2,13]. We are given an intensity image  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . Our algorithm assumes that the image contains a background and a foreground texture. At every point  $x \in \Omega$ , let  $N(x)$  be a  $R^2 \times 1$  vector of intensities over a small neighborhood<sup>2</sup>, say of size  $R \times R$ . Given a closed contour  $C$  on  $\Omega$ , we define the following two covariance matrices:

$$M^i(C) = \frac{\int_{int(C)} N(x)N(x)^T dx}{\int_{int(C)} dx}$$

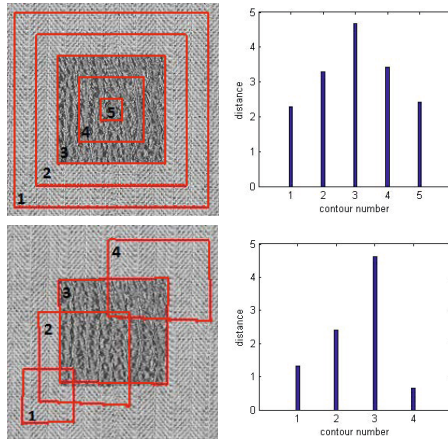
$$M^e(C) = \frac{\int_{ext(C)} N(x)N(x)^T dx}{\int_{ext(C)} dx}. \tag{15}$$

Note that  $M^i(C)$  and  $M^e(C)$  both belong to the set of  $R^2 \times R^2$  symmetric positive definite matrices, which is a Riemannian manifold henceforth denoted by  $PD(R^2)$ .

---

<sup>2</sup> Although we use a continuous region  $\Omega$  to model the image domain, we are implicitly assuming a discrete image domain while defining the concept of a neighborhood  $N(x)$ . We choose to ignore this discrepancy.

Let  $d : PD(R^2) \times PD(R^2) \rightarrow \mathbb{R}$  denote the geodesic distance between two points of this manifold. Since the image contains two different texture regions, it is evident that the two covariance matrices (points on this manifold) defined in (15) will be furthest away (in terms of geodesic distance) from each other when the contour  $C$  lies on the boundary between the two textures. We support this claim with an empirical evidence in Figure 1. For a given texture image  $I : \Omega \rightarrow \mathbb{R}$ , we propose the following cost function on the set of all closed contours defined on  $\Omega$ :



**Fig. 1.** (left) Different contours on an image with a foreground/background textures, (right) the corresponding (referred by appropriate contour number) geodesic distance between the covariance matrices defined in (15)

$$J(C) = d(M^i(C), M^e(C)) \tag{16}$$

where  $M^i(C), M^e(C)$  are defined in Equation (15)<sup>3</sup>. We find the contour  $C$  that maximizes this cost, using the gradient ascent approach giving us a novel active contour scheme. Instead of working on parametric representations of  $C$ , we work with its level set representation which has several benefits as discussed in [1]. Using level set function  $\phi$  with the convention given in Equation(2) and the Heaviside function, we redefine the covariance matrices from Equation (15), as

$$M^i(\phi) = \frac{\int_{\Omega} (1 - H(\phi)) N(x) N(x)^T dx}{\int_{\Omega} (1 - H(\phi)) dx}$$

$$M^e(\phi) = \frac{\int_{\Omega} H(\phi) N(x) N(x)^T dx}{\int_{\Omega} H(\phi) dx}. \tag{17}$$

Re-writing our cost function from Equation (16) in terms of the level set function  $\phi$  gives us

<sup>3</sup> Note the subtle difference between our cost function and that of ACWE:  $M^i, M^e$  are not variables to be optimized in contrast to  $\mu_1, \mu_2$  in Equation(1).

$$J(\phi) = d(M^i(\phi), M^e(\phi)). \quad (18)$$

The gradient of this functional is given as,

$$\frac{\partial J}{\partial \phi}(\phi) = \left\langle \frac{\partial d}{\partial M^i}, \frac{\partial M^i(\phi)}{\partial \phi} \right\rangle_{M^i} + \left\langle \frac{\partial d}{\partial M^e}, \frac{\partial M^e(\phi)}{\partial \phi} \right\rangle_{M^e} \quad (19)$$

where  $\langle \cdot, \cdot \rangle_{M^i}$  and  $\langle \cdot, \cdot \rangle_{M^e}$  are the Riemannian inner products defined on the Tangent space of  $PD(R^2)$  at points  $M^i(\phi)$  and  $M^e(\phi)$ , respectively. Specific details on this inner product can be found in [17]. The derivatives of the geodesic distance  $d$  is given by<sup>4</sup>

$$\frac{\partial}{\partial M^i} d(M^i, M^e) = -\text{Log}_{M^i}(M^e) \in T_{M^i} PD(R^2) \quad (20)$$

$$\frac{\partial}{\partial M^e} d(M^i, M^e) = -\text{Log}_{M^e}(M^i) \in T_{M^e} PD(R^2) \quad (21)$$

where  $\text{Log}$  denotes the Riemannian log map defined on  $PD(R^2)$ . Derivatives of the covariance matrices defined in Equation (17) are given by

$$\frac{\partial M^i}{\partial \phi}(\phi) = \frac{1}{|\Omega_{int}|} \int_{\Omega} (M^i(\phi) - N(x)N(x)^T) \delta(\phi) dx \quad (22)$$

$$\frac{\partial M^e}{\partial \phi}(\phi) = \frac{1}{|\Omega_{ext}|} \int_{\Omega} (N(x)N(x)^T - M^e(\phi)) \delta(\phi) dx \quad (23)$$

where  $|\Omega_{int}|$  and  $|\Omega_{ext}|$  are given by

$$|\Omega_{int}| = \int_{\Omega} (1 - H(\phi)) dx$$

$$|\Omega_{ext}| = \int_{\Omega} H(\phi) dx.$$

Substituting Equations (20),(21),(22),(23) into Equation (19), we get

$$\begin{aligned} \frac{\partial J}{\partial \phi} = \int_{\Omega} \left[ \left\langle -\text{Log}_{M^i}(M^e), \frac{1}{|\Omega_{int}|} (M^i(\phi) - N(x)N(x)^T) \delta(\phi) \right\rangle_{M^i} \right. \\ \left. + \left\langle -\text{Log}_{M^e}(M^i), \frac{1}{|\Omega_{ext}|} (N(x)N(x)^T - M^e(\phi)) \delta(\phi) \right\rangle_{M^e} \right] dx \quad (24) \end{aligned}$$

The gradient ascent as a level set evolution equation, after including the curvature flow as a regularizer (with weight  $1/\lambda$ ) and replacing  $\delta(\phi)$  by its smooth approximation  $\delta_{\varepsilon}(\phi)$ , is then given by

$$\begin{aligned} \frac{\partial \phi}{\partial t}(x) = \text{div} \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) \delta_{\varepsilon}(\phi) + \lambda \left( \left\langle -\text{Log}_{M^i}(M^e), \frac{1}{|\Omega_{int}|} (M^i(\phi) - N(x)N(x)^T) \delta_{\varepsilon}(\phi) \right\rangle_{M^i} \right. \\ \left. + \left\langle -\text{Log}_{M^e}(M^i), \frac{1}{|\Omega_{ext}|} (N(x)N(x)^T - M^e(\phi)) \delta_{\varepsilon}(\phi) \right\rangle_{M^e} \right) \quad (25) \end{aligned}$$

<sup>4</sup> A simpler explanation for this can be given in case we are working with  $\mathbb{R}^2$  instead of  $PD(R^2)$ . In this case  $\frac{\partial d}{\partial x}(x, y) = -(y - x)$  and  $\frac{\partial d}{\partial y}(x, y) = -(x - y)$ . This is exactly what is done by the Riemannian Log map on manifolds.

## 2.1 Convexification of the Energy

Following Section 1.2, we let

$$r(x, M^i, M^e) := \left\langle \text{Log}_{M^i}(M^e), \frac{1}{|\Omega_{int}|} (M^i(\phi) - N(x)N(x)^T) \right\rangle_{M^i} \\ + \left\langle \text{Log}_{M^e}(M^i), \frac{1}{|\Omega_{ext}|} (N(x)N(x)^T - M^e(\phi)) \right\rangle_{M^e} \quad (26)$$

Substituting  $r$  in Equation(25), we get

$$\frac{\partial \phi}{\partial t}(x) = \left[ \text{div} \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) - \lambda r(x, M^i, M^e) \right] \delta_\varepsilon(\phi).$$

Thus, the above evolution also minimizes the following convex energy

$$E(u) = \int_{\Omega} (\|\nabla u\| + \lambda r(x, M^i, M^e) u) dx, \quad (27)$$

where we replace  $\phi$  by  $u$  as in Section 1.2. Finally, our segmentation task is reduced to the minimization:

$$\min_{0 \leq u \leq 1} E(u). \quad (28)$$

The Split Bregman minimization procedure as discussed in Section1.2 is used for this minimization and is summarized below in Algorithm 1. In the next section we give

<b>Algorithm 1.</b> Split Bregman minimization
<p><b>while</b> <math>\ u^{k+1} - u^k\  &lt; \varepsilon</math>, <b>do</b></p> <ul style="list-style-type: none"> <li>- Compute <math>M^i(\phi), M^e(\phi)</math> from Equation(17) and <math>r^k</math> from Equation (26), using <math>\phi = u^k - 0.5</math>, where 0.5 is the selected threshold.</li> <li>- Compute <math>u^{k+1}</math> using Equation(13), <math>d^{k+1}</math> using Equation(14) and <math>b^{k+1}</math> using Equation(9).</li> </ul> <p><b>end while</b></p>

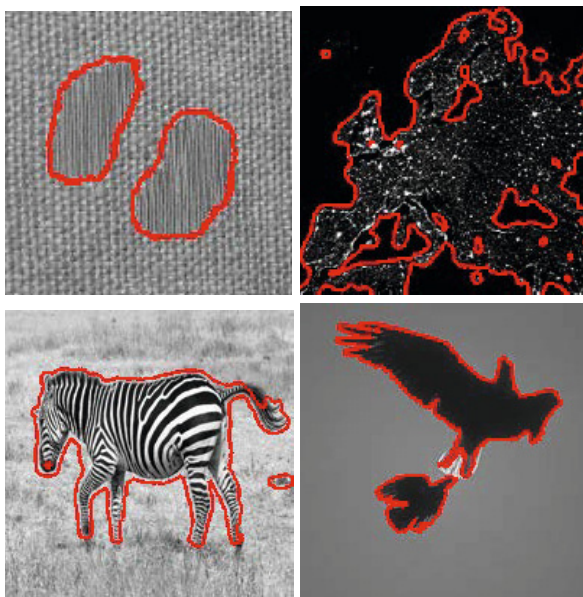
texture segmentation results on synthetic and real textures and compare the results with that of [22].

## 3 Experiments

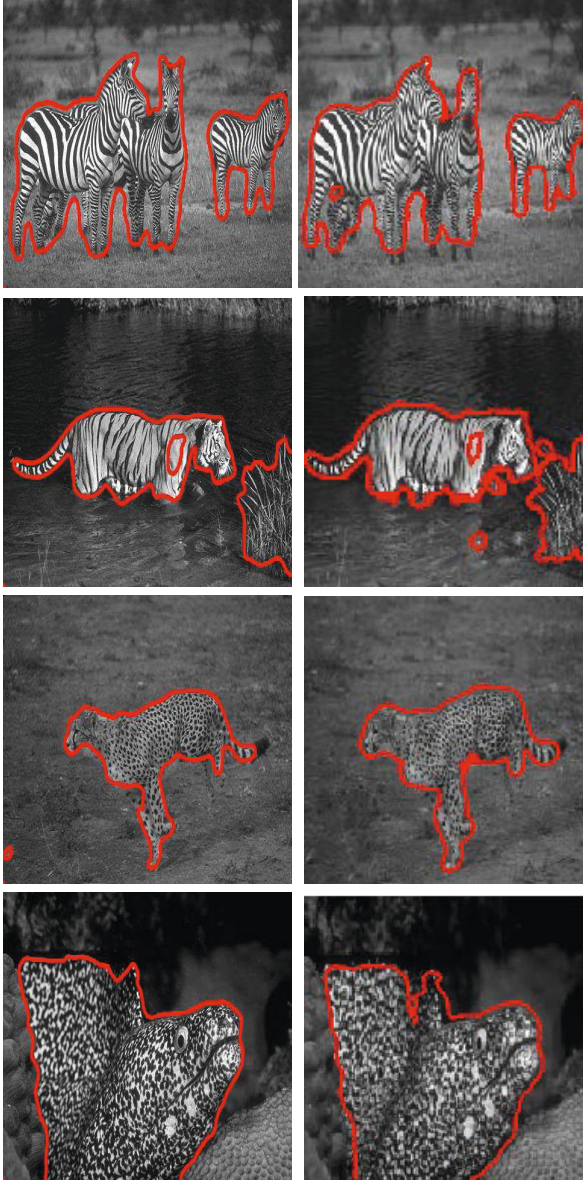
The results shown in this section were obtained on a Intel Core2Duo, 2GB RAM machine using MATLAB. We have used a random initialization for all images and all results were obtained within 2 – 5 minutes. Image sizes vary form  $150 \times 150$  pixels to  $250 \times 250$  pixels.



In Figure 2, we show successful segmentation for a synthetic texture image, on the Europe night sky image, image of a zebra, and an example of gray-valued image segmentation using our model. For gray-valued image segmentation, we use  $N(x) = I(x)$  in our model. The covariance matrices  $M^i(C), M^e(C)$  defined in Equation (15), will simply be the mean of squared intensities in the interior and exterior of the closed contour  $C$ , respectively. Also note that the covariance matrices now belong to  $PD(1)$ , i.e., the set of positive real numbers  $\mathbb{R}^+$ , of course with a metric different from the usual Euclidean one on  $\mathbb{R}$ . Our algorithm will then find the contour that maximizes the difference (geodesic distance on  $PD(1)$ ) between the two numbers  $M^i(C)$  and  $M^e(C)$ . We next compare our results with the results generated by the algorithm in [22], on some images from the Berkeley Segmentation dataset [26], in Figure 3. We have used images from [22] to display their results. One can clearly see that our algorithm gives a better texture segmentation. Small noise-like artifacts are in fact regions where texture similar to the object texture is present, for instance, in the tiger image, there are reflection of the tiger strips in the water that our algorithm is able to successfully segment. Note that the feature used in [22] is the determinant of the metric, which loses all directional properties of the texture. As shown in the center image in Figure 4, the distribution of the feature inside and outside the actual object boundary is not significantly different, while our algorithm is successful in segmenting the texture.



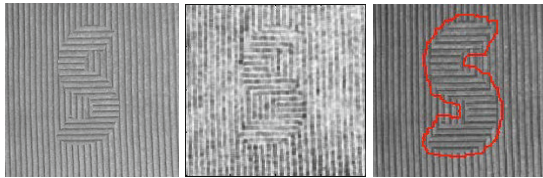
**Fig. 2.** Segmentation results on (top-left) artificial texture image, (top-right) Europe night sky image, (bottom-left) image of a zebra, and (bottom-right) gray-valued image. For the artificial texture image, we use a neighborhood of size  $7 \times 7$ ,  $5 \times 5$  for Europe night sky image,  $7 \times 7$  for the zebra image, and  $1 \times 1$  for image segmentation. Texture being defined using neighborhoods, the computed boundary can be observed to lie few pixels away from the actual object boundary.



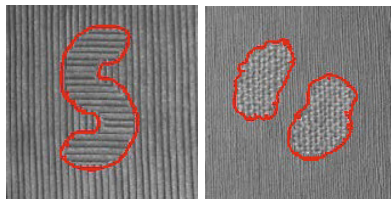
**Fig. 3.** Comparing results from [22](left column) with results from our algorithm (right column). The size of neighborhood for these results is  $9 \times 9$  pixels. It can be seen that our results are comparable if not better in most cases. The small noise-like artifacts are points around which object-like texture is present. For example, in the tiger image, our algorithm also captures the tiger-strips that appear due to reflection in water.

Instead of taking intensity values in the neighborhood as a feature, we next explore multiscale Gabor features at every pixel as a feature. Let  $f(x)$  denote the output of Gabor filters at 5 different scales in a single orientation. Using  $f(x)$  as  $N(x)$  also yields good texture segmentation results, as shown in Figure 5. In fact an advantage of these features is that, since higher scales will contain information from a neighborhood, there is no need to append features from a neighborhood to every pixel. This seems to yield a better boundary localization.

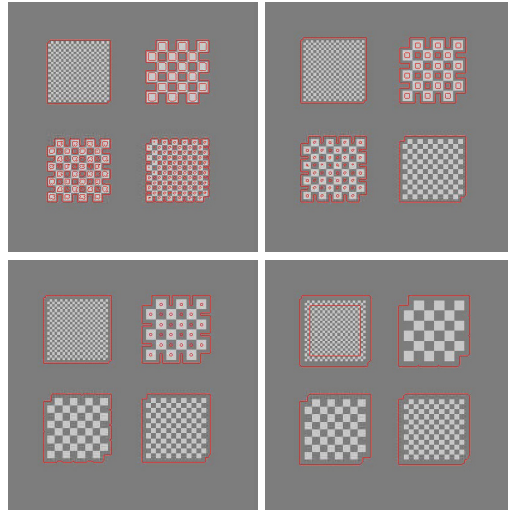
Finally, we show the effect of changing the size of neighborhood on the segmentation results in Figure 6. As expected, if the scale of texture is either too large or too small to be captured by the neighborhood size, the algorithm will over-segment or under-segment respectively. The synthetic image in Figure 6 contains checkerboard patterns of unit block size  $2 \times 2$ ,  $4 \times 4$ ,  $6 \times 6$  and  $8 \times 8$  each. With neighborhood size  $R = 3$ , the algorithm is able to capture only the smallest texture with blocks of size  $2 \times 2$ , with  $R = 5$ , it is able to capture texture with blocks upto  $4 \times 4$  and with  $R = 7$ , it is able to capture texture with blocks upto  $6 \times 6$ . Not surprisingly, with  $R = 9$ , it captures all textures involved, except the smallest with block size  $2 \times 2$ , since with  $R = 9$ , the small scale texture is more like a homogeneous region rather than a textured region.



**Fig. 4.** (left) Original image, texture inside is simply rotated version of the texture outside, (center) Image of the feature used in [22], notice that the feature is unable to distinguish the interior from the exterior, specifically, the distribution of this feature inside and outside the actual object boundary is not that different, (right) Result of our algorithm.



**Fig. 5.** Texture segmentation using Gabor features in our algorithm. Observe better localization of the boundary, especially in the image on the left (compare with Figure 4).



**Fig. 6.** Effect of changing neighborhood size  $R$  on segmentation: (top-left) Segmentation using  $R = 3$ , (top-right) Segmentation using  $R = 5$ , (bottom-left) Segmentation using  $R = 7$ , and (bottom-right) Segmentation using  $R = 9$ . Refer to the text for details.

## 4 Conclusion

In this paper, we propose a novel active contour based unsupervised texture segmentation algorithm. The algorithm finds a contour with maximum geodesic distance between its interior and exterior intensity covariance matrices. With the least possible neighborhood size  $R = 1$ , the process successfully segments gray-level images. Using convexification methods, the algorithm does not depend on user initialization and is able to compute globally optimal solutions. The computational complexity is also brought down by using the Split Bregman approach.

In its current state, the method depends on the size of the neighborhood  $R$ . Efforts are on to make it independent of  $R$ , either using a semi-supervised approach or using other multi-scale methods.

## References

1. Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (Applied Mathematical Sciences)*. Springer-Verlag New York, Inc., Secaucus (2006)
2. Boothby, W.M.: *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, London (1975)
3. Bresson, X., Esedoglu, S., Vanderghyest, P., Thiran, J.-P., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.* 28(2), 151–167 (2007)
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *International Journal of Computer Vision* 22(1), 61–79 (1997)

5. Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* 20(1-2), 89–97 (2004)
6. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics* 66(5), 1632–1648 (2006)
7. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* 10(2), 266–277 (2001)
8. Chaudhuri, B.B., Sarkar, N.: Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. Mach. Intell.* 17(1), 72–77 (1995)
9. Sagiv, C., Sochen, N.A., Zeevi, Y.Y.: Integrated active contours for texture segmentation. *IEEE Trans. Image Process* 15(6), 1633–1646 (2006)
10. Cross, G.R., Jain, A.K.: Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 5(1), 25–39 (1983)
11. Derraz, F., Peyrodie, L., Taleb-Ahmed, A., Forzy, G.: Texture segmentation using globally active contours model and cauchy-schwarz distance. In: 2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 391–395 (October 2012)
12. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE Transactions on Image Processing* 11(2), 146–158 (2002)
13. do Carmo, M.P.: *Riemannian Geometry*. In: *Riemannian Geometry*, Birkhäuser, Boston (1992)
14. Donoser, M., Bischof, H.: Using covariance matrices for unsupervised texture segmentation. In: *ICPR*, pp. 1–4 (2008)
15. Dunn, D.F., Higgins, W.E.: Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing* 5(7), 947–964 (1995)
16. Fauzi, M.F.A., Lewis, P.H.: Automatic texture segmentation for content-based image retrieval application. *Pattern Anal. Appl.* 9(4), 307–323 (2006)
17. Fletcher, P.T., Joshi, S.C.: Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing* 87(2), 250–262 (2007)
18. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split bregman method: Segmentation and surface reconstruction. *J. Sci. Comput.* 45(1-3), 272–293 (2010)
19. Goldstein, T., Osher, S.: The split bregman method for  $l_1$ -regularized problems. *SIAM J. Img. Sci.* 2(2), 323–343 (2009)
20. Haralick, R.M.: Statistical and structural approaches to texture. *Proceedings of the IEEE* 67(5), 786–804 (1979)
21. Houhou, N., Thiran, J.-P., Bresson, X.: Fast texture segmentation model based on the shape operator and active contour. In: *CVPR* (2008)
22. Houhou, N., Thiran, J.-P., Bresson, X.: Fast Texture Segmentation Based on Semi-local Region Descriptor and Active Contour. *Numerical Mathematics: Theory, Methods and Applications* 2(4), 445–468 (2009)
23. Howarth, P., Rüger, S.M.: Evaluation of texture features for content-based image retrieval. In: Enser, P.G.B., Kompatsiaris, Y., O’Connor, N.E., Smeaton, A.F., Smeulders, A.W.M. (eds.) *CIVR 2004. LNCS*, vol. 3115, pp. 326–334. Springer, Heidelberg (2004)
24. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
25. Laws, K.I.: *Textured image segmentation*. PhD thesis, Univ. Southern California, Los Angeles, CA, USA (1980)
26. 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: *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, pp. 416–423 (July 2001)

27. Muzzolini, R., Yang, Y.-H., Pierson, R.: Multiresolution texture segmentation with application to diagnostic ultrasound images. *IEEE Transactions on Medical Imaging* 12(1), 108–123 (1993)
28. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* 29(1), 51–59 (1996)
29. 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)
30. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. In: *CVPR* (2), pp. 699–706 (2003)
31. Savelonas, M.A., Iakovidis, D.K., Maroulis, D.E.: An LBP-Based Active Contour Algorithm for Unsupervised Texture Segmentation. In: *Proc. 18th International Conference on Pattern Recognition, ICPR 2006*, vol. 2 (2006)
32. Tamura, H., Mori, S., Yamawaki, T.: Textural Features Corresponding to Visual Perception. *IEEE Transaction on Systems, Man, and Cybernetics* 8(6), 460–472 (1978)
33. Todorovic, S., Ahuja, N.: Texel-based texture segmentation. In: *ICCV*, pp. 841–848 (2009)
34. Tuceryan, M., Jain, A.K.: Texture segmentation using voronoi polygons. *IEEE Trans. Pattern Anal. Mach. Intell.* 12(2), 211–216 (1990)
35. Tuzel, O., Porikli, F., Meer, P.: Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
36. Vese, L.A., Osher, S.: Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.* 19(1-3), 553–572 (2003)
37. Wang, Y., Yin, W., Zhang, Y.: A fast algorithm for image deblurring with total variation regularization. CAAM technical reports
38. Wang, Z., Vemuri, B.C.: DTI segmentation using an information theoretic tensor dissimilarity measure. *IEEE Trans. Med. Imaging* 24(10), 1267–1277 (2005)
39. Wu, C., Tai, X.: Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. *SIAM Journal on Imaging Sciences* 3(3), 300–339 (2010)
40. Yu, L., Gimpel, G.L.: Separating a colour texture from an arbitrary background. In: *DICTA*, pp. 489–498 (2003)
41. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (frame) – towards a unified theory for texture modeling. *International Journal of Computer Vision* 27(2), 1–20 (1998)
42. Zucker, S.W., Terzopoulos, D.: Finding Structure in Co-Occurrence Matrices for Texture Analysis. *Computer Graphics and Image Processing* 12, 286–308 (1980)

# Segmentation Using SubMarkov Random Walk<sup>\*</sup>

Xingping Dong<sup>1</sup>, Jianbing Shen<sup>1,\*\*</sup>, and Luc Van Gool<sup>2</sup>

<sup>1</sup> Beijing Laboratory of Intelligent Information Technology,  
School of Computer Science, Beijing Institute of Technology, Beijing 100081, P.R. China

<sup>2</sup> Computer Vision Laboratory, ETH Zurich, Zurich 8092, Switzerland  
shenjianbing@bit.edu.cn

**Abstract.** In this paper, we propose a subMarkov random walk (subRW) with the label prior with added auxiliary nodes for seeded image segmentation. We unify the proposed subRW and the other popular random walk algorithms. This unifying view can transfer the intrinsic findings between different random walk algorithms, and offer the new ideas for designing the novel random walk algorithms by changing the auxiliary nodes. According to the second benefit, we design a subRW algorithm with label prior to solve the segmentation problem of objects with thin and elongated parts. The experimental results on natural images with twigs demonstrate that our algorithm achieves better performance than the previous random walk algorithms.

**Keywords:** Segmentation, subMarkov random walk, auxiliary nodes.

## 1 Introduction

In many computer vision tasks, the random walk (RW) has been widely used such as segmentation [4], clustering [12], and classification [17]. Grady *et. al.* [5] first proposed the RW for medical image segmentation and extend it in [4] for general image segmentation. In their work, the user should give labels to a small number of pixels. Then the probability of each unlabeled pixel belonging to a label is determined by the probability that a random walker starting at each unlabeled pixel will first reach one pixel with this label. By assigning each pixel to a label with the greatest probability, the interactive image segmentation result can be obtained. After [5], there are many related and important works based on RW [3,14,2]. In [3], the RW has been extended to segment out disconnected objects by using prior models without labeling each object. In other words, the user only indicates labels on some objects and the other similar objects will be segmented. Sinop and Grady [14] proposed a common framework to unify the RW, the graph cuts, and the shortest path algorithms for interactive segmentation. Further, the

---

<sup>\*</sup> This work was supported in part by the National Basic Research Program of China (973 Program) (No. 2013CB328805), the Key Program of NSFC-Guangdong Union Foundation (No. U1035004), the National Natural Science Foundation of China (No. 61272359), and the Program for New Century Excellent Talents in University (NCET-11-0789). Beijing Higher Education Young Elite Teacher Project. Specialized Fund for Joint Building Program of Beijing Municipal Education Commission.

<sup>\*\*</sup> Corresponding author.

authors in [2] added the popular watershed segmentation algorithm to this framework and make the theoretical analysis for the connection of these algorithms. This unifying framework brings some benefits, like opening new possibilities for using unary terms in traditional watershed algorithm, and using watershed to optimize more general models.

In general, these algorithms [7,13] are graph-based so we can use a graph to describe the image for introducing them. In [7], Kim *et. al.* propose a random walk with a restarting probability (RWR) for segmentation. It means that this random walker will return to the starting node with a probability  $c$  at each step, and walk to other adjacent nodes with the probability  $1 - c$ . Shen *et. al.* [13] proposed a novel lazy random walk (LRW) algorithm with self-loops to effectively solve the superpixel segmentation problem in weak boundary and complex texture regions. In [16], Wu *et. al.* proposed another similar RW algorithm called partially absorbing random walk (PARW) for some applications based on cluster, such as ranking, classification and so on. Comparing the above three RW-based algorithms, we have found that they all satisfy the subMarkov property [9] i.e. the sum of transition probabilities  $\sum q(i, j)$ , that a random starts from a node to the other adjacent nodes, is less than or equal to 1.

In this paper, we propose a subMarkov random walk (subRW) to unify the three RW-based algorithms: RWR, LRW and PARW, and extend it by adding label prior to solve the twig problem. First, according to the subMarkov property, we build a subRW framework for segmentation. In subRW, a random walker will leave a graph  $G$  from a node  $i$  with probability  $c_i$  and walk to the other adjacent nodes in  $G$  with probability  $1 - c_i$ . This random walker can be transformed to a random walker with Markov transition probability ( $\sum q(i, j) = 1$ ) walks in the expanded graph  $G_e$ . This graph is constructed by adding auxiliary staying nodes connected with seeds and an auxiliary killing nodes connected with unseeded nodes into the graph  $G$ . Then we unify the subRW and the aforementioned three RW-based algorithms in the expanded graph. After analyzing the connections between them, we find an idea to design a new RW-based algorithm by changing edges or adding auxiliary nodes. According to this idea, we design a novel subRW with label prior to solve the twig problem. This label prior can be viewed as global ‘seeds’ connected with all nodes. Each global ‘seed’ corresponds to a label. So we can add some prior nodes connected with all nodes into the graph  $G_e$  to build a new expanded graph  $G_p$ . Then we compute the probability that a random walker starting from each node reaches the staying nodes or the prior nodes in the graph  $G_p$ , as the likelihoods probability of corresponding labels. In the other word, we want to compute the probability of reaching the user specified seeds plus the probability of reaching the global ‘seeds’. These global ‘seeds’ will help to segment out the twigs parts. Our subRW source code will be publicly available online<sup>1</sup>.

## 2 An Unifying View of subRW

In this section, we propose a novel random walk algorithm with a subMarkov transition probability (subRW) for interactive mutil-labeled image segmentation and analyze the relations between this proposed algorithm and other popular RW algorithms, such as RWR [7], LRW [13], and PARW [16].

<sup>1</sup> <https://github.com/shenjianbing/subrw14>



We first give some important notations and their corresponding descriptions. An image is formulated as a weighted undirected graph  $G = (V, E)$  with nodes  $v \in V$ , and edges  $e \in E \subseteq V \times V$ . Each node  $v_i$  represents an image pixel  $x_i$ . An edge  $e_{ij}$  connects two nodes  $v_i$  and  $v_j$  in neighborhood system. The weight  $w_{ij} \in \mathbf{W}$  of edge  $e_{ij}$  measures the likelihood that a random walker will cross this edge. As many previous graph-based segmentation algorithms [4,7,15,6,8,13], a weight  $w_{ij}$  is defined as the Gaussian weighting function:

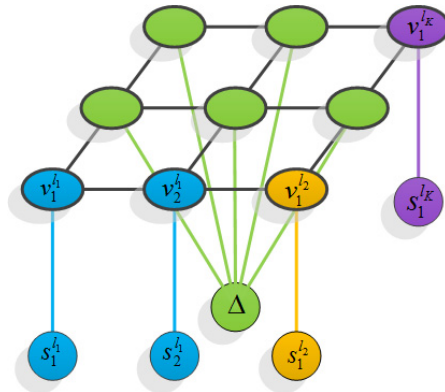
$$w_{ij} = \exp\left(-\frac{\|I_i - I_j\|^2}{\sigma}\right) + \epsilon, \tag{1}$$

where  $I_i$  and  $I_j$  are the pixel colors at two nodes  $v_i$  and  $v_j$  in Lab color space,  $\sigma$  is a controlling parameter which is set as 1/60 in this paper, and  $\epsilon$  is a small constant as  $10^{-6}$ . The degree matrix  $\mathbf{D}$  is a diagonal matrix where  $\mathbf{D}_{ii} = d_i$ ,  $d_i = \sum_{j \sim i} w_{ij}$  is the degree of a node  $v_i$ ,  $j \sim i$  represents a node  $v_j$  is in the neighborhood (not include itself) of  $v_i$ .  $N$  is the number of nodes (pixels).

In our approach, the user needs to indicate some scribbles on foreground objects as multi-labeled seeds. Here, we define these seeds as a set of labeled nodes  $V_M = \{V^{l_1}, V^{l_2}, \dots, V^{l_K}\}$ . Then a set of labels  $LS = \{l_1, l_2, \dots, l_K\}$  is also defined, where  $K$  is the number of labels  $V^{l_k} = \{v_1^{l_k}, v_2^{l_k}, \dots, v_{M_k}^{l_k}\}$  and  $M_k$  is the number of seeds with label  $l_k$ .

### 2.1 The subMarkov Random Walk

Given a weighted graph  $G$ , a set of labeled nodes  $V_M$ , and a set of unlabeled nodes  $V_U$ , where  $V_U \cap V_M = V$ , the multi-labeled image segmentation can be formulated as a labeling problem, which means assigning each node  $v_i \in V$  with a label from the set  $LS$ . This problem can be solved by comparing the likelihoods probability  $r_i^{l_k}$  of



**Fig. 1.** The nodes graph of a subRW. The ellipse nodes denote the original nodes in  $V$  and the circle nodes are the newly added auxiliary nodes. The green ellipses are the unseeded nodes and the others are the seeded nodes.

each node belonging to a label  $l_k$  in our algorithm. Before computing this likelihoods probability, we define the subMarkov transition probability  $q$  on  $V$  as follows:

**Definition 1:**  $q$  denotes a subMarkov transition probability if for each node  $v_i$

$$\sum_{j \sim i} q(i, j) \leq 1. \quad (2)$$

According to [9], a subMarkov transition probability has the following property:

**Property 1:** through adding a auxiliary node  $\Delta$ , a subMarkov transition probability  $q$  on  $G$  can be made into a (Markov) transition probability on  $V \cup \{\Delta\}$  by setting  $q(\Delta, \Delta) = 1$  and  $q(i, \Delta) = 1 - \sum_{j \sim i} q(i, j)$ . The probability  $q(i, \Delta)$  can be viewed as a probability that a random walker leaves the graph  $G$ .

According to the above property, we can design different subMarkov random walk algorithms by adding different auxiliary nodes. In fact, the popular random walk algorithms, such as RWR [7], LRW [13], and PARW [16], can be interpreted in this view (more details will be given in next subsection). We first consider a general subRW algorithm for interactive seeded image segmentation. Two kinds of auxiliary nodes are added into the graph  $G$  to get an expanded graph  $G_e$ . As shown in Fig. 1, one kind of auxiliary node is a killing node  $\Delta$  connected with all unseeded nodes (e.g. the green circle node in Fig. 1). When a random walker from a node  $v_i$  reaches this node, it will be killed at this node and the corresponding probability will be omitted. In other words, an effective random walker will not reach this node. The other one is the staying node  $s_m^{l_k}$  connected with the  $m$ -seeded node with label  $l_k$  (e.g. the blue, orange or purple circle nodes in Fig. 1). When a random walker reaches this node which can be viewed as a target node, it will stay at this node. We denote  $S_M$  as a set of staying nodes corresponding to the seeds set  $V_M$ , and  $c_i$  as the leaving probability for each node in  $V$ . Then, the transition probability on  $V \cup \{\Delta\} \cup S_M$  is formulated as follows:

$$q(i, j) = \begin{cases} c_i, & \text{if } i \in V, j \in \{\Delta\} \cup S_M \\ (1 - c_i) \frac{w_{ij}}{d_i}, & \text{if } j \sim i \in V \\ 1, & \text{if } i = j \in \{\Delta\} \cup S_M \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Suppose a random walker starts from a node  $v_i \in V$  and walks on  $V \cup \{\Delta\} \cup S_M$  with the transition probability  $q(i, j)$  in (3). By setting  $r_{im}^{l_k}$  as the reaching probability that this random walker reaches the auxiliary staying node  $s_m^{l_k}$ , then we have

$$r_{im}^{l_k} = \begin{cases} (1 - c_i) \sum_{j \sim i} \frac{w_{ij}}{d_i} r_{jm}^{l_k} + c_i \cdot 1 & \text{if } v_i = v_m^{l_k}, \\ (1 - c_i) \sum_{j \sim i} \frac{w_{ij}}{d_i} r_{jm}^{l_k} + c_i \cdot 0 & \text{otherwise.} \end{cases} \quad (4)$$

The vector notation  $\mathbf{r}_m^{l_k} = [r_{im}^{l_k}]_{N \times 1}$  ( $N = |V|$  is the number of nodes, which is formulated as follows:

$$\mathbf{r}_m^{l_k} = (\mathbf{I} - \mathbf{D}_c) \mathbf{P} \mathbf{r}_m^{l_k} + \mathbf{D}_c \mathbf{b}_m^{l_k}, \quad (5)$$

where  $\mathbf{b}_m^{l_k} = [b_{im}^{l_k}]_{N \times 1}$  is the  $N$ -dimensional indicating vector with  $b_{im}^{l_k} = 1$  if  $v_i = v_m^{l_k}$  and  $b_{im}^{l_k} = 0$  otherwise.  $\mathbf{D}_c$  is a diagonal matrix which diagonal element is  $c_i$  i.e.  $\mathbf{D}_c = \text{diag}(c_1, c_2, \dots, c_N)$ , and  $\mathbf{I}$  is a  $N \times N$  identity matrix. The transition matrix  $\mathbf{P} = [p_{ij}]_{N \times N}$  is a row-normalized matrix of the adjacency matrix  $\mathbf{W}$  (defined in (1)):

$$p_{ij} = w_{ij}/d_i. \quad (6)$$

The number of a set of seeded nodes with same label is often larger than one. A good RW approach should consider all seeded nodes. We use an average reaching probability  $r_i^{l_k}$ , that a random walker from a node  $v_i$  reaches a set of staying nodes with label  $l_k$ , as the likelihood of assigning this node to the label  $l_k$ . The formulation (5) can be rewritten as follows:

$$\mathbf{r}_m^{l_k} = (\mathbf{I} - (\mathbf{I} - \mathbf{D}_c)\mathbf{P})^{-1} \mathbf{D}_c \mathbf{b}_m^{l_k} = \mathbf{E}^{-1} \mathbf{D}_c \mathbf{b}_m^{l_k}, \quad (7)$$

where  $\mathbf{E} = \mathbf{I} - (\mathbf{I} - \mathbf{D}_c)\mathbf{P}$ .

Thus, a vector formulation of this average steady-state probability  $\mathbf{r}^{l_k}$  can be given as follows:

$$\mathbf{r}^{l_k} = \frac{1}{Z M_k} \sum_{m=1}^{M_k} \mathbf{r}_m^{l_k} = \frac{1}{Z_k M_k} \mathbf{E}^{-1} \mathbf{D}_c \mathbf{b}^{l_k}, \quad (8)$$

where  $\mathbf{b}^{l_k} = [b_i^{l_k}]_{N \times 1}$  is a vector with  $b_i^{l_k} = 1$  if  $v_i \in \mathbf{V}^{l_k}$  and  $b_i^{l_k} = 0$  otherwise,  $Z_k$  is a normalized constant. The final labeling result (i.e. the segmentation result) for each node  $v_i \in V$  is obtained as follows:

$$R_i = \arg \max_{l_k} r_i^{l_k}. \quad (9)$$

## 2.2 Relations with other Well-Known RW Algorithms

We will analyze the relations between the proposed subMarkov random walk and the other popular algorithms: RWR [7], LRW [13], and PARW [16]. And we will find these algorithms can be unified and related with the subRW.

**Relations with RWR.** In [7], Kim *et al.* suppose a random walker starts from a  $m$ -th seed node  $v_m^{l_k}$  of label  $l_k$  in a graph  $G$ . Different from the traditional random walker, it has a restarting probability  $c$  to return to the seed  $v_m^{l_k}$  at each step. Then each node is assigned a steady-state probability  $f r_{im}^{l_k}$  that this random walker will finally stay at this node, which is formulated as:

$$f r_{im}^{l_k} = (1 - c) \sum_{j \sim i} \frac{w_{ij}}{d_i} f r_{jm}^{l_k} + c \cdot b_{im}^{l_k}. \quad (10)$$

Equation (4) of subRW is rewritten as follows:

$$r_{im}^{l_k} = (1 - c_i) \sum_{j \sim i} \frac{w_{ij}}{d_i} r_{jm}^{l_k} + c_i \cdot b_{im}^{l_k}. \quad (11)$$

Combining Equations (10) and (11), we find that the RWR is a special case of subRW with leaving probability  $c_i = c, i = 1, 2, \dots, N$ . In other words, the subRW can be viewed as a set of RWR with different restarting probability at each node.

**Relations with LRW.** In [13], Shen *et al.* propose a lazy random walks for superpixel segmentation, which is viewed as a multi-labeled segmentation method. Under their framework, a random walker will stay at the current position with the probability  $(1 - \alpha)$  and walk out along arbitrary edge with the probability  $\alpha$ . They use the commute time  $CT_{ij}$ , which is the expected number of steps for a lazy random walker starting at  $v_i$  to  $v_j$  and then returning, to measure the likelihoods probability that this two nodes belong to the same label. After normalizing the commute time, the likelihoods probability  $f_m^{l_k}$ , that a node  $v_i$  has the same label with the seeded node  $v_m^{l_k}$ , can be formulated as:

$$\mathbf{f}_m^{l_k} = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{b}_m^{l_k}, \tag{12}$$

where  $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ ,  $\mathbf{D}$  is a diagonal matrix and  $\mathbf{D}_{ii} = d_i$ . We rewrite Equation (12) as follows:

$$\mathbf{f}_m^{l_k} = \alpha\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\mathbf{f}_m^{l_k} + \mathbf{b}_m^{l_k}, \tag{13}$$

In fact, the labeling result of LRW (or other algorithms based on RW) only depends on the likelihoods probabilities of each node with different labels i.e.  $Rl_i = \arg \max_{l_k} f_{im}^{l_k}$ . So we can scale these likelihoods probabilities to interpret the LRW in a subRW view, which will not change this labeling result. By setting  $\beta = (1 - \alpha)d_{\min}^{1/2}$ ,  $d_{\min} = \min_{i=1:N} d_i$ , and  $\mathbf{r}_m^{l_k} = \beta\mathbf{D}^{-1/2}\mathbf{f}_m^{l_k}$ , we then multiply Equation (13) by  $\beta\mathbf{D}^{-1/2}$  to obtain:

$$\mathbf{r}_m^{l_k} = \alpha\mathbf{D}^{-1}\mathbf{W}\mathbf{r}_m^{l_k} + \beta\mathbf{D}^{-1/2}\mathbf{b}_m^{l_k}, \tag{14}$$

This above equation is rewritten as follows:

$$r_{im}^{l_k} = \alpha \sum_{j \sim i} \frac{w_{ij}}{d_i} r_{jm}^{l_k} + (1 - \alpha)\gamma_i b_{im}^{l_k} + (1 - \alpha)(1 - \gamma_i) \cdot 0, \tag{15}$$

where  $\gamma_i = (\frac{d_{\min}}{d_i})^{1/2}$ . According to this equation, we find that the LRW can also be viewed as a subMarkov random walk by adding an edge between each seeded node and the killing node  $\Delta$ . Then the corresponding transition probability on  $V \cup \{\Delta\} \cup S_M$  can be formulated as:

$$q(i, j) = \begin{cases} 1 - \alpha, & \text{if } i \in V_U, j = \Delta \\ (1 - \alpha)\gamma_i & \text{if } i \in V_M, j \in S_M \\ (1 - \alpha)(1 - \gamma_i) & \text{if } i \in V_S, j \in S_M \\ \alpha \frac{w_{ij}}{d_i}, & \text{if } j \sim i \in V \\ 1, & \text{if } i = j \in \{\Delta\} \cup S_M \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

**Relations with PARW.** In [16], partially absorbing random walks (PARWs) is proposed for ranking, clustering, and classification. Wu *et al.* suppose that a PARW is absorbed at current node  $i$  with probability  $\alpha_i$ , and walk out of it following a random edge with probability  $1 - \alpha_i$ . They set the probability  $a_{ij}$  that a PARW starting from node  $i$ , is absorbed at node  $j$  in any finite number of steps. Then the matrix  $\mathbf{A} = [a_{ij}]_{N \times N}$  of absorption probabilities is formulated as:

$$\mathbf{A} = (\mathbf{\Lambda} + \mathbf{D} - \mathbf{W})^{-1} \mathbf{\Lambda}, \quad (17)$$

where  $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ ,  $\lambda_i \geq 0, i = 1 : N$ . To solve the segmentation or labeling problem, we can use the absorption probability  $rp_{im}^{l_k}$  that a PARW starting from a seeded node  $v_m^{l_k}$  is absorbed at node  $v_i$ , as the likelihoods belonging to the label  $l_k$ . The vector notation  $\mathbf{rp}_m^{l_k}$  is:

$$\mathbf{rp}_m^{l_k} = \mathbf{A} \mathbf{b}_m^{l_k}. \quad (18)$$

By combining Equation (17), Equation (18) can be rewritten as:

$$\mathbf{rp}_m^{l_k} = (\mathbf{\Lambda} + \mathbf{D})^{-1} (\mathbf{W} \mathbf{rp}_m^{l_k} + \mathbf{\Lambda} \mathbf{b}_m^{l_k}). \quad (19)$$

The above equation is then equivalent to:

$$rp_{im}^{l_k} = \sum_{j \sim i} \frac{w_{ij}}{d_i + \lambda_i} rp_{jm}^{l_k} + \frac{\lambda_i}{d_i + \lambda_i} b_{im}^{l_k}. \quad (20)$$

By comparing the above equation and Equation (11), we find that (20) is equivalent to (11) when  $c_i = \frac{\lambda_i}{d_i + \lambda_i}$ , i.e. the PARW is equivalent to the subRW. In fact, after submit  $c_i = \frac{\lambda_i}{d_i + \lambda_i}$  to  $\frac{1-c_i}{d_i}$ , then we have  $\frac{1-c_i}{d_i} = \frac{1}{d_i + \lambda_i}$ .

**Merits of an Unifying View.** We have shown that subRW can unify or relate the popular models based on RW. There are at least two merits of the unifying view. First, it builds the connections between different RW-based algorithms, so that it is easier to transfer findings between them. For example, in RWR [7], the authors have discussed the influence of parameter  $c$ . We can choose the parameter  $c_i$  according their discussion since the subRW is the generalized version of RWR. Second, a unifying view offers a new way to design the novel RW-based algorithm by adding some new auxiliary nodes or changing the edges between the auxiliary nodes and the original nodes in  $V$ . For example, the LRW can be viewed as an expansion of the subRW. Inspired by the second merit, we design a new subRW with label prior to segment out the twigs object.

### 3 Segmentation via subRW with Label Prior

An object with twigs can be separated into two parts: main branch object and the twigs part. Usually, the twigs part is similar to the main object, so the appropriate user specified scribbles on main object have included enough information for segmenting out the twigs part. But most RW-based algorithms does not make full use of this information and often omit the twigs part. In this section, we want to add a label prior constructed by these scribbles into the subRW to help segment out the twigs part.

In general, the user specified scribbles are considered as an exact label prior. Unfortunately, this prior only work at the seeded nodes and all unseeded nodes do not have this prior. Therefore, we want to give all nodes in  $V$  a new label prior, which maybe less exact than user scribbles, but can be used for unseeded nodes. This label prior is constructed by the user scribbles i.e. the seeded nodes. We can use probability distributions to build the prior model. Assume a label  $l_k$  has an intensity distribution  $H_k$

for each node, where  $u_i^k$  denotes the probability density belonging to  $H_k$  at node  $v_i$ . This distribution can be learnt via a wide array of techniques, such as the kernel estimation, Gaussian Mixture Model (GMM), and so on. Here, the GMM is used as the prior model. Each prior distribution  $H_k$  is viewed as a GMM learnt by the seeded nodes with label  $l_k$  (more details about GMM learning can be seen in [1]). Given these prior distributions, we can add a set of prior auxiliary nodes  $H_M = \{h_1, h_2, \dots, h_K\}$  into the expanded graph  $G_e$  and get a graph with prior  $\bar{G}$ . Each prior node is connected with all nodes in  $V$  and the weight  $w_{ih_k}$  of an edge between a prior node  $h_k$  and a node  $v_i \in V$  is proportional to the probability density  $u_i^k$  i.e.  $w_{ih_k} \propto u_i^k$ . In this paper, we set the weight  $w_{ih_k} = (1 - c_i)\lambda u_i^k$ , where  $\lambda$  is a controlling parameter, which measures the importance of the prior distribution.

Then the transition probability on  $V \cup \{\Delta\} \cup S_M \cup H_M$  is formulated as follows:

$$\bar{q}(i, j) = \begin{cases} c_i, & \text{if } i \in V, j \in \{\Delta\} \cup S_M \\ (1 - c_i) \frac{\lambda u_i^k}{d_i + \lambda g_i}, & \text{if } i \in V, j = h_k \\ (1 - c_i) \frac{w_{ij}}{d_i + \lambda g_i}, & \text{if } j \sim i \in V \\ 1, & \text{if } i = j \in \{\Delta\} \cup S_M \cup H_M \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

where  $g_i = \sum_{k=1}^K u_i^k$ .

Given a transition probability  $\bar{q}$  on a graph with prior  $\bar{G}$ , the probability  $\bar{r}_{im}^{l_k}$ , that a random walker from a node  $v_i \in V$  reaches the  $m$ -th staying node  $s_m^{l_k}$  with label  $l_k$  or the prior node  $h_k$ , is formulated as follows:

$$\bar{r}_{im}^{l_k} = (1 - c_i) \sum_{j \sim i \in V} \frac{w_{ij} \bar{r}_{jm}^{l_k}}{d_i + \lambda g_i} + (1 - c_i) \frac{\lambda u_i^k}{d_i + \lambda g_i} + cb_{im}^{l_k}, \quad (22)$$

where  $b_{im}^{l_k} = 1$  if  $v_i = v_m^{l_k}$  and  $b_{im}^{l_k} = 0$  otherwise.

In fact, this prior node  $h_k$  can be viewed as a new staying node with label  $l_k$ , so this reaching probability of  $h_k$  should be considered. By setting a vector  $\bar{\mathbf{r}}_m^{l_k} = [\bar{r}_{im}^{l_k}]_{N \times 1}$ , we can get the vector formulation of Equation (22):

$$\begin{aligned} \bar{\mathbf{r}}_m^{l_k} &= (\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{P}} \bar{\mathbf{r}}_m^{l_k} + (\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{u}}^k + \mathbf{D}_c \mathbf{b}_m^{l_k} \\ &= (\mathbf{I} - (\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{P}})^{-1} ((\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{u}}^k + \mathbf{D}_c \mathbf{b}_m^{l_k}) \\ &= \bar{\mathbf{E}}^{-1} ((\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{u}}^k + \mathbf{D}_c \mathbf{b}_m^{l_k}), \end{aligned} \quad (23)$$

where  $\bar{\mathbf{E}} = \mathbf{I} - (\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{P}}$ , the transition probability matrix  $\bar{\mathbf{P}} = [\bar{p}_{ij}]_{N \times N}$  in  $V$  is defined as:

$$\bar{p}_{ij} = \frac{w_{ij}}{d_i + \lambda g_i}, \quad (24)$$

$\bar{\mathbf{u}}^k = [\bar{u}_i^k]_{N \times 1}$  is a vector with

$$\bar{u}_i^k = \frac{\lambda u_i^k}{d_i + \lambda g_i}. \quad (25)$$

As mentioned before, we use the average reaching probability  $\bar{r}_i^{l_k}$  for each node  $v_i \in V$  as the likelihoods belonging to the label  $l_k$ . The vector notation  $\bar{\mathbf{r}}^{l_k}$  is formulated as:

$$\bar{\mathbf{r}}^{l_k} = \frac{1}{Z_k} \bar{\mathbf{E}}^{-1} ((\mathbf{I} - \mathbf{D}_c) \bar{\mathbf{u}}^k + \frac{1}{M_k} \mathbf{D}_c \mathbf{b}^{l_k}), \quad (26)$$

The final labeling (segmentation) result with a label prior is obtained as follows:

$$\bar{R}_i = \arg \max_{l_k} \bar{r}_i^{l_k}, \quad (27)$$

where  $\bar{R}_i$  represents the final label for each node i.e. the pixel in an image.

Adding the label prior may produce some noises. One solution is decreasing the parameter  $\lambda$ . However, when  $\lambda$  is too small, the twig parts may be lost. Then we need to use the other strategy to decrease these noises. Combining the label prior value for each node, we can get a coarse segmentation result:

$$CR_i = \arg \max_k u_i^k. \quad (28)$$

## 4 Experimental Results

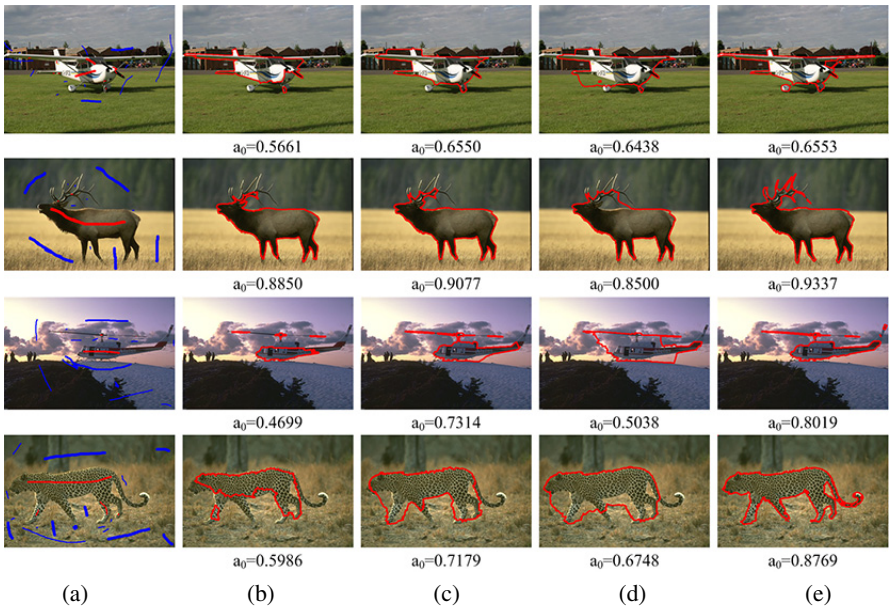
In this section, we evaluate the performance of the proposed subRW with label prior on both synthetic and natural images. We compare our algorithm with the state-of-the-art methods including RW [4], RWR [7], and LRW [13] algorithms in qualitative and quantitative aspects. The implementation codes of these three algorithms are offered by the respectful authors, and the optimal parameters in their papers are used to run the experimental results. Our algorithm includes two main parameters: the leaving probabilities  $c_1, c_2, \dots, c_N$  and the label prior parameter  $\lambda$ . The leaving probabilities control the probability that a random walker reaches the staying nodes (seeds), which principally influence the regions without twigs. As mentioned before, when all of  $c_i$  are set as the same constant, the subRW is equivalent to the RWR. And a RWR with a proper restarting probability performs well in most nature images without twigs. Then we empirically set the leaving probabilities to a constant  $c_i = 4e - 4$ .



(a) scribbled image (b)  $\lambda = 1e - 11$  (c)  $\lambda = 1e - 10$  (d)  $\lambda = 5e - 10$  (e)  $\lambda = 1e - 9$

**Fig. 2.** The segmentation results with varying parameter  $\lambda$  with thin and elongated objects

Then we should pay more attention to the parameter  $\lambda$ . In order to better explain the effect of  $\lambda$ , we should consider the influence of noise. The selecting parameter  $\gamma$  for the noise reducing process is set as 0. Fig. 2 shows an example of the segmentation results by our algorithm with varying parameter  $\lambda$ . It appears that more and more twigs of bee are successfully segmented out (Fig. 2(b)-(e)) in a proper range of  $\lambda$  when the value of  $\lambda$  increases gradually. If  $\lambda$  is too large, some noise may occur such as the right-up corner of Fig. 2(e). This may be caused by the inaccurate distribution estimation of GMMs. If  $\lambda$  is too small, the twigs of object will be lost (Fig. 2(b)). Therefore, we can make  $\lambda$  to be small or large according to the results with too much noise or losing many twigs. Through the extensive experiments, we find the proper  $\lambda$  may fall into the range of  $[1e - 11, 1e - 9]$  for most nature images. In this paper, we set  $\lambda = 2e - 10$  for most test images in our experiments.



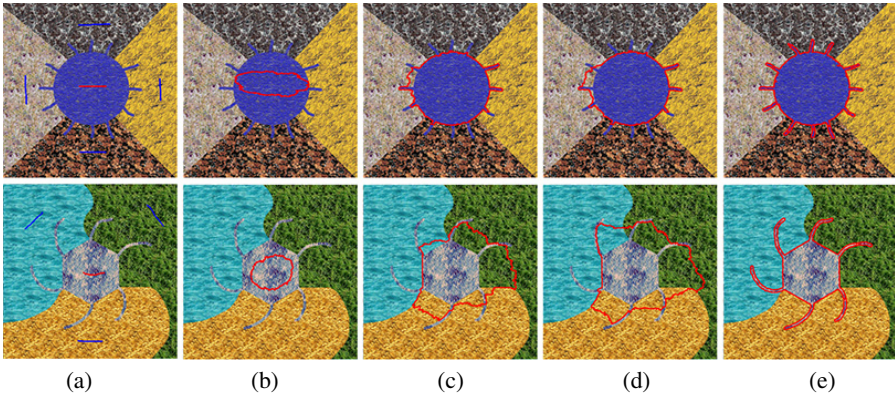
**Fig. 3.** Comparisons between our algorithm and the state-of-the-art algorithms. (a) The input scribbled images. (b), (c), (d) and (e) are the segmentation results of RW [4], RWR [7] with  $c = 4e - 4$ , LRW [13] with  $1 - \alpha = 1e - 4$ , and our method with  $c_i = 4e - 4$ ,  $\lambda = 2e - 10$ .

We have also compared our algorithm with the other three well-known RW algorithms for natural images shown in Fig. 3. These images are taken from two datasets: the Berkeley segmentation dataset (BSD) [10] and the Microsoft research cambridge object recognition image database (MSRC). The manual labeled ground-truth masks are also provided in these two datasets. We adopt a normalized overlap  $a_0$  [14] to measure the accuracy of the segmentation result for quantitative comparison. We choose the natural images with twigs for our experiments. Some of them own complex texture like the cheetah, and some of them own very thin twigs such as the helicopter. Fig. 3 shows



that our algorithm outperforms the other algorithms no matter in quality or in quantity. In the qualitative comparisons, it is obvious that our method not only successfully segment out the most twigs of objects, but also adhere the edges to the boundaries better than the others. This is due to the adding label prior also has impact on the main part of the object. In the quantitative comparisons, our improvements is also significant such as the airplane and the leopard in Fig. 3, where the twigs parts are very small and our improvements are still very evident for these complicated images.

We further compared our algorithm with the other algorithms on synthetic textured images. The goal of texture segmentation is to extract the texture with twigs parts from these images. The segmentation results in Fig. 4 (e) show that all the twigs are completely segmented out and the boundaries of main part are also correctly detected by our algorithm. As shown in Fig. 4(b)-(d), the other RW algorithms [4,7,13] do not perform well for these texture images. The RW method almost does not find the right boundaries since there are too many short noise edges in textured images. Both RWR and LRW algorithms reduce the probability that a random walker walks on the original image graph, which also reduce the influence of these barriers. Then the random walker in RWR or LRW will be more likely to find the correct boundaries. Fig. 4 (c) and (d) show the similar results by RWR and LRW algorithms. Our algorithms still perform very well for this complicated situation.



**Fig. 4.** Comparisons between our algorithm and other RW algorithms on synthetic texture images. (a) The input scribbled images. (b), (c), (d) and (e) are the segmentation results by RW [4], RWR [7] ( $c = 1e-6$ ), LRW [13] ( $1 - \alpha = 1e-6$ ), and our subRW ( $\gamma = 1$ ,  $c_i = 4e-4$ ,  $\lambda = 4e-11$ ).

## 5 Conclusions

A novel subMarkov random walk approach has been proposed for seeded image segmentation in this work. Our framework can be explained as a traditional random walk walks on the graph by adding some new auxiliary nodes, which makes it to be easily understand and to be more flexible. Under this framework, we unify the well-known RW-based algorithms, which satisfy the subMarkov property and build bridges to make it easy to transform the findings between them. Furthermore, we design a novel subRW

with label prior to solve the twigs segmentation problems by adding prior nodes into our framework. This also demonstrates that it is feasible to design a new subRW algorithm by adding new auxiliary nodes into our framework. The experimental results have shown that our algorithm outperforms the previous well-known RW-based methods. In the future, we will extend our framework to more applications by adding different auxiliary nodes in computer vision, such as saliency cut for stereo images [11].

## References

1. Calinon, S., Guenter, F., Billard, A.: On learning, representing and generalizing a task in a humanoid robot. *IEEE Trans. on Systems, Man and Cybernetics, Part B* 37(2), 286–298 (2007)
2. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watershed: A unifying graph-based optimization framework. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 33(7), 1384–1399 (2011)
3. Grady, L.: Multilabel random walker image segmentation using prior models. In: *IEEE CVPR*, pp. 763–770 (2005)
4. Grady, L.: Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28(11), 1768–1783 (2006)
5. Grady, L., Funka-Lea, G.: Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In: Sonka, M., Kakadiaris, I.A., Kybic, J. (eds.) *CVAMIA/MMBIA 2004*. LNCS, vol. 3117, pp. 230–245. Springer, Heidelberg (2004)
6. Jegelka, S., Bilmes, J.: Submodularity beyond submodular energies: coupling edges in graph cuts. In: *IEEE CVPR*, pp. 1897–1904 (2011)
7. Kim, T.H., Lee, K.M., Lee, S.U.: Generative image segmentation using random walks with restart. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 264–275. Springer, Heidelberg (2008)
8. Kohli, P., Osokin, A., Jegelka, S.: A principled deep random field model for image segmentation. In: *IEEE CVPR*, pp. 1971–1978 (2013)
9. Lawler, G.F., Limic, V.: *Random walk: a modern introduction*. Cambridge University Press (2010)
10. 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: *IEEE ICCV*, vol. 2, pp. 416–423 (2001)
11. Peng, J., Shen, J., Jia, Y., Li, X.: Saliency cut in stereo images. In: *IEEE ICCVW*, pp. 22–28 (2013)
12. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(11), 1873–1890 (2007)
13. Shen, J., Du, Y., Wang, W., Li, X.: Lazy random walks for superpixel segmentation. *IEEE Trans. on Image Processing* 23(4), 1451–1462 (2014)
14. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: *IEEE ICCV*, pp. 1–8 (2007)
15. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: *IEEE CVPR*, pp. 1–8 (2008)
16. Wu, X.M., Li, Z., So, A.M., Wright, J., Chang, S.F.: Learning with partially absorbing random walks. In: *NIPS*, pp. 3077–3085 (2012)
17. Zhu, X., Nejdil, W., Georgescu, M.: An adaptive teleportation random walk model for learning social tag relevance. In: *ACM SIGIR*, pp. 223–232 (2014)

# Automatic Shape Constraint Selection Based Object Segmentation

Kunqian Li<sup>1</sup>, Wenbing Tao<sup>1</sup>, Xiangli Liao<sup>1</sup>, and Liman Liu<sup>2</sup>

<sup>1</sup> School of Automation and National Key Laboratory of Science and Technology on Multi-spectral Information Processing, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>2</sup> School of Biomedical Engineering, South-Central University for Nationalities, Wuhan 430074, China

**Abstract.** In this paper, an object segmentation algorithm based on automatic shape constraint selection is proposed. Different from the traditional shape prior based object segmentation methods which only provide loose shape constraints, our proposed object segmentation gives more accurate shape constraint by selecting the most appropriate shape among the standard shape set. Furthermore, to overcome the inevitable differences between the true borders and the standard shapes, the Coherent Point Drift (CPD) is adopted to project the standard shapes to the local ones. A quantitative evaluating mechanism is introduced to pick out the most suitable shape prior. The proposed algorithm mainly consists of four steps: 1) the initial GrabCut segmentation; 2) standard shape projection by CPD registration; 3) rank the standard shapes according to the evaluation scores; 4) refine GrabCut segmentation with the chosen shape constraint. The comparison experiments with the related algorithms on Weizmann\_horse dataset have demonstrated the good performance of the proposed algorithm.

## 1 Introduction

Object segmentation and shape matching are fundamental tasks in computer vision and image processing which are closely related with each other. On the one hand, accurate segmentation of foreground object is the premise of shape matching, and on the other the appropriate mapped shapes which can be regraded as prior knowledge provide global auxiliary boundary information for segmentation in a complex scene where local information is ambiguous and unreliable.

We propose here a point sets matching based segmentation algorithm where a standard shape set is considered as candidate shape prior pool. A standard shape with the highest confidence of boundary prediction is automatically selected in our shape evaluation system. The selected shape prior will be incorporated into MRF-based framework in the form of level-set which keeps all the pair-wise terms submodular, and then graph cuts is used to achieve the global optimization.

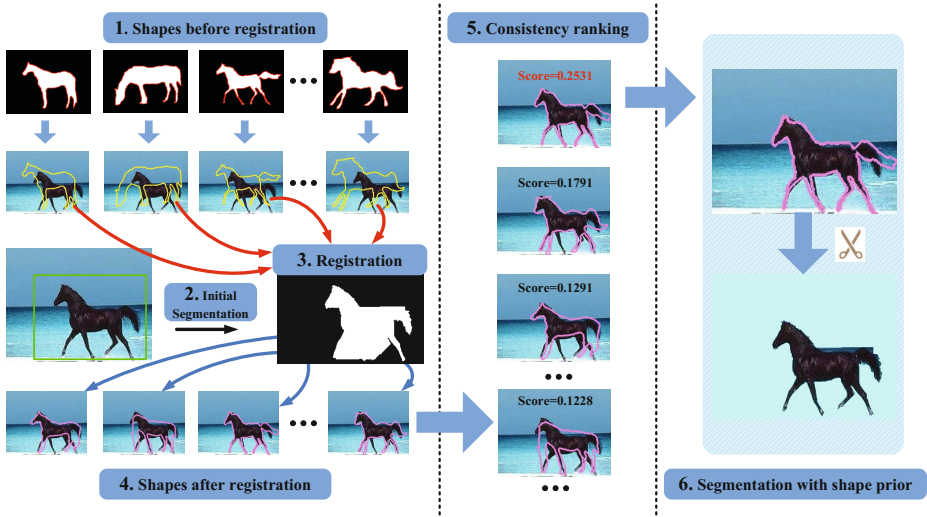
## 1.1 Related Works

**Graph Cut Based Segmentation.** In recent years, graph cuts based segmentation approaches have been shown to be quite accurate and efficient and have attracted more and more attention. Compared with fully automatic graph-based segmentation approaches, such as normalized cut [1], graph cuts based segmentation methods with interactions alleviate their inherent problems, i.e., the high computational complexity and the unavailability for label modification.

The graph cuts based segmentation approach was first proposed and tested by Boykov *et al.* [2]. They raised an interactive segmentation method for monochrome images with N-dimension. Firstly, the users are required to label some pixels as foreground or background. Afterwards, the histograms of gray values are computed to describe the feature distributions of foreground and background respectively. In the end, graph cuts is used to find the globally optimal segmentation. However, for color images it is impractical to get adequate description of the color space. Grab-Cut [3] extends it to color images by replacing the histograms based model with Gaussian mixture model (GMM). And furthermore, the segmentation process iteratively alternates between GMM parameters learning and segmentation estimation to solve the min-cut problem until it converges. Lazy snapping [4] produces high quality segmentations almost in real-time by incorporating the advantages of graph cuts and pre-segmentation methods (such as watershed segmentation [5]), but it works poor for thin structures.

**Segmentation with Shape Priors.** Along with the development of interactive segmentation algorithms, more and more investigators find it hard to get satisfactory segmentations within loose interactions especially in complex scenes. But at the same time, prior knowledge has been demonstrated that it makes segmentation more robust by reducing ambiguous partitions which are inconsistent with the prior. Shape prior, as a kind of auxiliary information, has been incorporated into many segmentation methods [6–12].

Slabaugh *et al.* [6] included an elliptical shape prior into the interactive graph cut framework, which is implemented only within an iterative refinement process. In [7], Funka-Lea *et al.* encouraged the target to be a convex blob surrounding the certain point. Another blob-like prior named compact shape prior is introduced in [8]. Vicente *et al.* [9] imposed connectivity constraints in the segmentation to overcome the shrinking bias of graph cut methods. In [12], an object specific shape prior is introduced in the form of level-set which keeps all the pairwise terms submodular. Another prior knowledge called star shape prior [10] provides more generic constraint for graph cut segmentation and Gulshan *et al.* [11] developed it by replacing the single star and Euclidean rays with multiple stars and Geodesic paths which make the method more generally applicable. Beyond that, Kim *et al.* [13] introduced another category-independent shape prior for object segmentation which utilizes the shared shapes among the objects from different categories. A non-parametric prior that transfers object shapes from an exemplar database to a test image based on local shape matching is presented and is incorporated into graph cut formulation to produce a pool



**Fig. 1.** An overview of the flow of the standard shape set constraint based object segmentation pipeline. We start with the input image and performing initial segmentation on it with GrabCut method [3]. And then, by using CPD registration method, the standard shapes are mapped to the initial shape which is extracted from the initial object segmentation. Through the consistency ranking process, the most appropriate shape will be automatically determined by selecting the one with the highest consistency score. Eventually, by taking the chosen shape as constraint, segmentation result improves greatly.

of segmentation hypotheses. This method assumes no specific classes, however training a huge exemplar database is needed. Besides, the general applicability of these shape priors also brings the lack of constraint ability which results in the poor segmentations in complicated scene.

## 1.2 Contribution

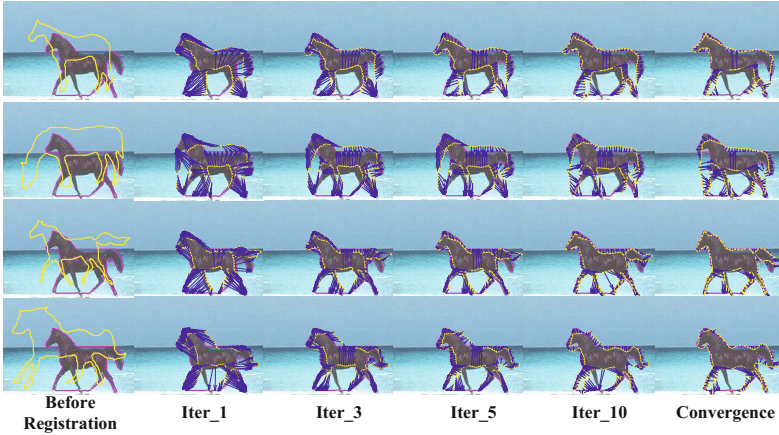
To overcome the limitations of the traditional shape constraint mentioned above, a standard shape set is selected as the constraint in our approach, where a group of shapes depicting a particular object in different postures are included. As Fig. 1 described, we start with projecting these standard shapes to the correct locations of the image according to the initial segmentation. And then, the projected standard shapes are put into the proposed consistency evaluation system, the shape with highest evaluation score will be selected as the final shape constraint which will be integrated into MRF-based framework in the next step. The key contribution of our approach lies in three aspects:

- a) A group of standard shapes which come from natural images are provided as constraint. Rather than general shape prior discussed in the former works, concrete shape constraint is full of detail shape information and global posture constraint. On the other hand, multiple constraints overcome the significant shape changings brought by postures diversity which are common in natural images, the best fitted shape will be selected as the final constraint.
- b) An excellent point sets matching method called Coherent Point Drift (CPD) is used to project the standard shapes to the initial segmentation boundary. This method not only overcomes the rigid transformations such as translation, rotation and scaling, captures the local non-rigid deformations under the similar posture, but also keeps the topological structures of the point sets thanks to the coherently moving constraint. The fixed topological structure guarantees the correctness and effectiveness of the boundary prediction.
- c) A statistical distance distribution based shape consistency evaluation system is proposed, where the most appropriate shape constraint can be automatically determined. The evaluation criterion is based on the fact that the more parts of the standard boundaries are overlapped with the initial segmentation edges, the higher the boundary prediction confidence of the standard shape will be. The deviated parts of the choose standard shape are assumed to have predicted the true boundaries in a complicated background.

**Organization.** The rest of the paper is organized as follows. After a brief introduction of the Coherent Point Drift (CPD) method, how to evaluate the conformity between the standard shapes and the target objects is presented in Section 3. Section 4 presents how to integrate the shapes with graph cut framework. Section 5 describes the implementation details and gives the settings of parameters. The experiment results and comparisons are also displayed in this section. Finally, some concluding remarks are presented in Section 6.

## 2 Point Sets Registration: Coherent Point Drift

The goal of point set registration is to find the meaningful correspondences between the two point sets and to recover the underlying transformation that maps one point set to the other. When it comes to shape matching, it appears as if a shape is moving towards the fixed one in the registration process. The Coherent Point Drift (CPD) method [14] is an excellent point set registration method which is robust to noise, outliers, rotation and slight non-rigid transformations. The alignment of two point sets is viewed as a probability density estimation problem in CPD algorithm. The points of the moving point set, which is called the model set, are treated as the centroids of Gaussian Mixture Models(GMM). The fixed point set is treated as the data set generated by those GMMs. Then, the point sets registration can be formulated as a maximum likelihood estimation problem. The GMM centroids are forced to move toward the data set coherently so as to keep the topological structure of the point set.



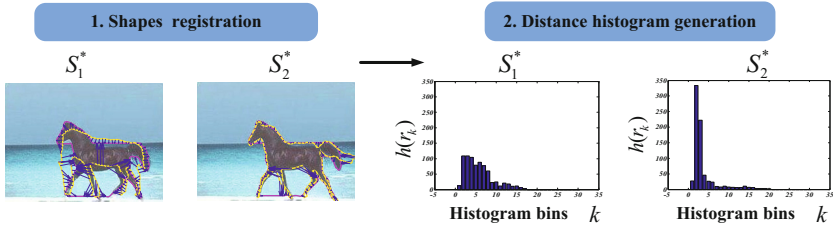
**Fig. 2.** Some examples of iterative process of shape registration, where the purple lines denote the initial shape extracted from initial segmentation and the yellow lines describe the dynamic changing process of the standard shape. The locations of the initial and standard shapes turn out to be overlapped when the shapes are well fitted.

Let  $S = \{x_k | x_k \in \mathbb{R}^2, k = 1, \dots, n\}$  be the initial object shape in the given image and  $S_i = \{y_l | y_l \in \mathbb{R}^2, l = 1, \dots, m\}$  be a standard shape template from the standard shape set  $\mathcal{F} = \{S_i\}, i = 1, \dots, N$ . We consider the points in  $S_i$  as the centroids of a Gaussian Mixture Model and the points in  $S$  as the data points generated by the GMM centered at  $y_l$ . Assume all the GMM components have the equal mixture coefficient and an additional uniform distribution is added to the mixture model to account for the noises and outliers. Specifying the weight of the uniform distribution as  $\omega$ ,  $0 \leq \omega \leq 1$ , then the mixture model takes the form

$$p(x_k | \mathbf{T}, \sigma, \omega) = (1 - \omega) \sum_{l=1}^m p_l p(x_k | y_l) + \omega \cdot p(\text{outlier}), \quad (1)$$

where  $p(x_k | y_l)$  describes the probability of the points  $x_k$  in the Gauss distribution centered at point  $y_l$  and  $\sigma$  is the variance for all Gaussian components.  $\mathbf{T}$  is the motion function which can be written as the linear combination of kernels  $\mathbf{T}(z) = \sum_{j=1}^m \varphi_j K(z, y_j) = \mathbf{K}\Phi$ , where  $\mathbf{K}$  is a  $m \times m$  kernel matrix and  $K(y_i, y_j) = e^{-\frac{\|y_i - y_j\|^2}{2\beta}}$ . Thus, the likelihood is a mixture model of distributions for inliers and outliers which is defined as  $\mathcal{L}(\mathbf{T}, \sigma, \omega) = \prod_{k=1}^n p(x_k | \mathbf{T}, \sigma, \omega)$ .

The GMM centroids  $y_l$  in  $S_i$  move coherently as a group to be fit to the data points  $x_k$  in  $S$  under the coherence constraint given by Tikhonov regularization which is defined as the prior  $p(\mathbf{T}) = e^{-\frac{\alpha}{2} \|\mathbf{T}\|_{\mathcal{H}}^2}$ , where  $\|\mathbf{T}\|_{\mathcal{H}}^2$  is the norm of  $\mathbf{T}(y)$  in the Reproduction Kernel Hilbert Space (RKHS). Therefore, the posterior probability  $P(\mathbf{T}, \sigma, \omega) \propto L(\mathbf{T}, \sigma, \omega)p(\mathbf{T})$ . Thus, we can solve the maximum a



**Fig. 3.** Two unnormalized distance histograms of the mapped standard shapes. Take the standard shapes from Fig. 1 for example, their distance histograms are calculated respectively. A distance histogram with the majority elements falling into the first several bins will be favored by the evaluation criteria, such as the histogram of  $S_2^*$ . The scattered distribution of the elements usually implies that the mapped shape deviates from the initial one to some extent,  $S_1^*$  for example.

posteriori (MAP) problem to estimate the transformation  $\mathbf{T}$ . This is equivalent to minimizing the negative log-posterior  $\varepsilon(\mathbf{T}, \sigma, \omega)$  as

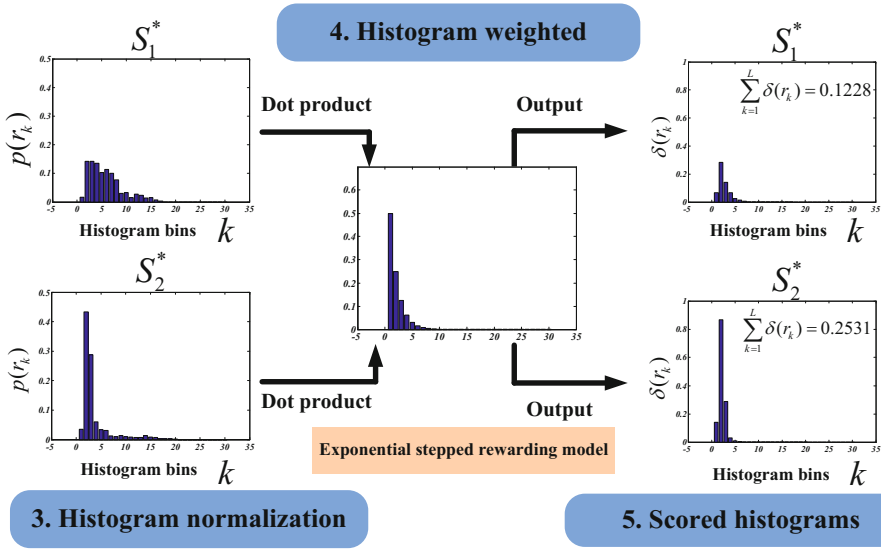
$$\varepsilon(\mathbf{T}, \sigma, \omega) = -\sum_{i=1}^n \log(p(x_k|\mathbf{T}, \sigma, \omega)) + \frac{\lambda}{2} \|\mathbf{T}\|_{\mathcal{H}}^2. \quad (2)$$

The Expectation-Maximization (EM) algorithm [15] is chosen to optimize the energy function (2). After the algorithm converges, a transformation  $\mathbf{T} = \mathbf{K}\Phi$  is estimated, which is used to map the standard shape  $S_i$  to the initial shape  $S$  to get  $S_i^*$ . In Fig. 2, several examples are used to show the shape registration process where the correspondences between two shape point sets and the mapped shapes are presented during the iterations.

### 3 Consistency Evaluations of Initial Object Shape and the Standard Shapes

After all the standard shapes in the template set  $\mathcal{S}$  are mapped to the initial shape  $S$  to get  $\mathcal{F}^* = \{S_i^*\}$ ,  $i = 1, \dots, N$ , we need to find the most appropriate shape prior which keeps the topological structure as well as the local shape features by evaluating the consistency between the initial shape and the mapped standard shapes. Considering that the error shape information implied in the priors will make the local boundaries more ambiguous and seriously degrade the segmentation results, the shape priors should be able to provide as much accurate shape information as possible and contain relatively few borders that may generate equivocalty. We propose here a quantitative way to rank the standard shape which is based on the statistical distance distribution. As already stated in our previous letter, the initial segmentation is supposed to have captured partial meaningful boundaries. Through the matching between the given standard shapes and the meaningful boundaries, the unmatched parts of the standard





**Fig. 4.** The scoring mechanism of the mapped shapes. The normalized distance histograms of  $S_1^*$  and  $S_2^*$  are displayed in left. The exponential stepped rewarding function ( $\rho = 0.5$ ) is shown in the middle as a histogram where the height of each bar denotes the corresponding scoring factor. Perform point multiplication operation on the distance histogram and the rewarding model by viewing them as vectors, then the scoring result can be obtained by adding all the bars of point product histogram. The rewarding factor  $M = 8$ .

shapes provide reliable guiding information for the following segmentation. Note that, another assumption proposed here is that the better the consistency of initial segmentation and the standard shape is, the more likely the standard shape shares the same outlines with the foreground object. Here, consistency is used to describe the fitness between two shapes. We first define the distance between a point  $p$  and a point set  $S$  in two-dimensional space as the minimum Euclidean distance of  $p$  and  $p_i \in S$ , i.e.,

$$\varepsilon(p, S) = \min\{\varepsilon(p, p_1), \varepsilon(p, p_2), \dots, \varepsilon(p, p_n)\}, p_i \in S. \quad (3)$$

To characterize the aberration of the mapped standard shapes with the initial segmentation borders, distance histogram, which is analogous to density histogram, is introduced into our evaluation approach. The distance histogram of shape  $S_i$  corresponding to  $S$  is a discrete function  $h(r_k) = n_k$ , where  $r_k$  is the  $k$ th distance level and  $n_k$  is the number of pixels in the image having distance level  $d_k$ . Equivalently,

$$h(r_k) = \#\{p | \varepsilon(p, S) \in \text{bin}(k)\}, k = 1, 2, \dots, K. \quad (4)$$

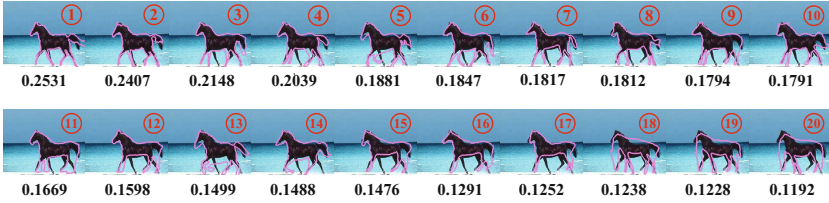


Fig. 5. The ranking results of 20 mapped standard shapes

Considering that the outliers of  $S_i$  may be far from the initial shape  $S$ , we only select pixels with  $\varepsilon(p, S)$  less than a threshold (one-tenth diagonal distance of the image). Note that, we normalize the distance histograms to overcome the influence caused by different numbers of shape points. By dividing  $n_k$  by the total number of adopted points in  $S^*$ , denoted by  $n^*$ , the normalized histogram can be given as

$$p(r_k) = n_k/n^*, k = 1, 2, \dots, K. \tag{5}$$

In Fig. 3, the unnormalized distance histograms are displayed on the right. A distance histogram with most elements falling into the first several bins means that, for most shape points, their projected location are almost overlapped with the initial shape points. On the contrary, the scattered distribution of the distance elements usually implies that the projected shape has totally deviated from the initial one. Obviously, the former one is more favorable.

To provide each mapped shape with a quantitative evaluation result which is based on the distance histogram, we propose a scoring mechanism and rank the shapes with their scores. Especially, an exponential stepped rewarding model is proposed, and the score of mapped shape  $S^*$  corresponding to  $S$  can be calculated by

$$\delta(S^*, S) = \sum_{k=1}^L \delta(r_k) = \sum_{k=1}^L (\rho^{k-1} \times p(r_k)), k = 1, 2, \dots, K, \tag{6}$$

where  $\rho(0 < \rho \leq 1)$  is the falling strength of the exponential stepped rewarding function which reflects the tolerance of shape deviation. In our experiments,  $\rho = 0.5$  is a good choice. The diagram of the scoring mechanism is displayed in Fig. 4. The normalized distance histograms of  $S_1^*$  and  $S_2^*$  are shown in the left. The exponential stepped rewarding function is shown in the middle as a histogram where the height of each bar denotes the corresponding scoring factor. Weight each histogram bin with the exponential stepped rewarding function, then the scoring results of the mapped shape can be obtained by adding all the weighted terms. The ranking results of 20 mapped standard shapes are shown in Fig. 5, which are obtained by following the steps in Fig. 3 and Fig. 4 according to the serial numbers.

## 4 Graph Cut Based Segmentation with Shape Prior

In the traditional graph-based segmentation formulation, image  $I$  which is defined on pixel set  $\bar{V}$  is mapped to a weighted graph  $G = (\bar{V}, \bar{E})$ . Extracting objective foreground of image can be posed as an pixel-wise binary labeling problem. That is to say, each pixel will be assigned a unique label  $L_p$  from label set  $\mathcal{L} = \{0, 1\}$  by performing max-flow/min-cut algorithm on the graph  $G$  to minimize the following energy function,

$$E = \mu \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}: L_p \neq L_q} V_{p,q}(L_p, L_q), \quad (7)$$

where  $\mathcal{P}$  is the set of all pixels in image  $I$ ,  $\mathcal{N}$  is the set of all pairs of neighboring pixels defined on 4 or 8-connected neighborhood system,  $L_p = 1$  ( $L_p = 0$ ) stands for assigning pixel  $p$  as foreground (background) pixel,  $\mu$  is weight for different terms. The first term of the energy function called regional or data term, which encodes individual label-preferences of pixels based on observed data and the specified likelihood function. For color images, it indicates how the color feature of pixel  $p \in \mathcal{P}$  fits into the known appearance models (e.g. Gaussian mixture model), i.e.  $M_{fg}$  and  $M_{bg}$ . Let  $f(p)$  be the color feature distribution of pixel  $p$ , and then the probability of pixel  $p$  belonging to the foreground (background) is formulated as:

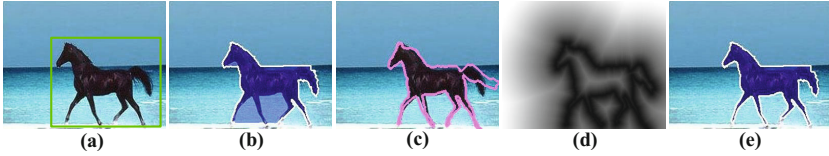
$$D_p(L_p) = \begin{cases} -\log \Pr(f(p)|M_{fg}) & \text{if } L_p = 1, \\ -\log \Pr(f(p)|M_{bg}) & \text{if } L_p = 0. \end{cases} \quad (8)$$

Obviously, a smaller value indicates a better matching. The second term of (7) called boundary or smooth term encourages spatial coherence by penalizing neighboring pixels with different labels, where  $V_{p,q}(L_p, L_q) \propto e^{-\alpha(f(p)-f(q))^2/2\gamma^2}$ .  $V_{p,q}(L_p, L_q)$  is large when  $f(p)$  and  $f(q)$  are similar and  $V_{p,q}(L_p, L_q)$  is close to zero when they are extremely different. In another word, a lower punishment indicates a higher possibility of  $p, q$  to be boundary pixels.

Supposed that a specific shape has been provided as the global constraint for segmentation, and then we prefer a good fitness between segmentation boundary and shape prior. Freedman *et al.* [12] proposed an ingenious way to incorporate shape prior into the graph cut framework. The shape prior is parametrically specified as a curve named  $s(c)$  and the boundary of the segmentation is defined as  $\tilde{s}(c) = \mathcal{B}\{p \in \mathcal{P}, L_p = 1\}$ , where  $\mathcal{B}\{\Omega\}$  is the boundary of point set  $\Omega$ . If we limit the shape parameter  $c$  on  $[0, 1]$ , and then the fitness between segmentation boundary and shape constraint can be evaluated by a natural energy function:

$$E_s[\tilde{s}(c)] = \int_0^1 \|s(c) - \tilde{s}(c)\|^2 dc. \quad (9)$$

Obviously,  $E_s[\tilde{s}(c)]$  will achieve global minimum  $E_s[\tilde{s}(c)] = 0$  when  $s(c) = \tilde{s}(c)$  for all values of  $c$ . A segmentation whose border is deviated from the given shape prior will receive a large punishment. And now the new object function can be defined as



**Fig. 6.** Segmenting horse. (a) The user interactions: only a rectangle around the horse is provided. (b) The segmentation results without shape constraints, the foreground targets are covered by blue masks. (c) The corresponding shape constraints. (d) The level-set form of the shape constraints. (e) The segmentation results are greatly improved with the constraint of shape priors.

$$E_s = E + E_s[\tilde{s}(c)]. \quad (10)$$

In order to make (10) an energy function able to be minimized by graph cuts, the shape prior constraint is specified as a regular and unsigned distance function whose zero level set is correspondent to the shape itself. Then, the terms of shape constraint in the energy function can be rewritten as

$$E_s[\tilde{s}(c)] = \sum_{(p,q) \in \mathcal{N}: L_p \neq L_q} \psi(p, q, s), \quad (11)$$

where  $\psi(p, q, s)$  measures the distance between the differently labeled neighboring pixels  $p, q$  and the shape prior  $s$ , it satisfies  $\psi(p, q, s) \approx 0$  when the pixel pair lies near the shape prior. An example of such a function for a horse shape curve is given in Fig. 6.

Eventually, the newly formulated energy function for segmentation with shape prior can be written as:

$$E_s = \sum_{p \in \mathcal{P}} (1 - \eta) \mu D_p L_p + \sum_{(p,q) \in \mathcal{N}: L_p \neq L_q} (1 - \eta) w_{p,q} + \eta \psi(p, q, s) \quad (12)$$

where  $\eta (0 \leq \eta \leq 1)$  is weight for different parts of pairwise terms. Minimizing this energy function with graph cuts leads to desired segmentation constrained by shape prior.

## 5 Experiments

### 5.1 Implementation Details

The standard shape set constraint based segmentation algorithm first segments the image by GrabCut method to obtain the initial object shape. And then, each shape in the standard shape set is mapped to the initial object shape by CPD registration. The key of our algorithm lies in the scoring mechanism, which

determines the final ranking of the mapped shapes. Our proposed scoring mechanism will give each mapped shape a score to describe the goodness of boundary fitting as well as the confidence of being a constraint. The standard shape with highest score will be selected as the constraint to refine the unsatisfactory object segmentation.

During the iterations, the algorithm terminates itself when the segmentation does not change anymore or reaches maximum number of iteration (the number is set to 3 in our implementation). Note that, we update the shape  $S$  by extracting the segmentation border of last loop to make the mapped shapes more accurate than the previous ones. We employ the publicly available implementation of GrabCut [3] and CPD registration [14] in our experiments and build the whole segmentation system in MATLAB/C++. The running time is closely related with the time cost on CPD registration. Generally, for two shapes each contains 200-300 points, CPD algorithm needs about 0.5 secs to make them matched on a 3.0 Ghz processor. Given a  $300 \times 200$  image constrained by 10 standard shapes, the overall segmentation process takes about 5-10 secs on the same platform.

## 5.2 Parameter Setting

There are several parameters that must be appropriately determined for the implementation of the proposed method. Part of the default values for these parameters have been given in the place where we discussed the corresponding algorithms. In view of integrity and clarity, here the descriptions of the parameter settings are given again. There are two important free parameters:  $\lambda$  and  $\beta$  in the CPD registration algorithm which are set to 25 and 2, respectively. In CPD registration algorithm, parameter  $\lambda$  is the weight of the smoothness penalization term which reflects the trade-off between data fitting and the smoothness of transformation.  $\beta$  is the Gaussian bandwidth when computing the kernel matrix  $\mathbf{K}$  which reflects the strength of interaction between points. Generally speaking, the larger  $\lambda$  and  $\beta$  are, the more coherent and smooth the transformation will be. The original intention of the proposed method lies in rectifying the unsatisfactory segmentation with the assistance of the standard shapes, the premise of which is that the standard shapes should keep their overall structures and local shape features along the registration. Therefore, relatively larger values for  $\lambda$  and  $\beta$  are preferred here. The distance histogram parameter  $L$  is chosen as 30 to build an accurate statistical model for the deviation distributions of the mapped shapes. Another important parameter  $\rho$  which greatly influences the accuracy of the chosen shape constraint is fixed to 0.5 to guarantee the distinctions of the evaluation scores. As to the parameters in formula (12), we set  $\mu = 10$  and the particular choice is not very important since changing  $\gamma$  will change the relative weight between the data and smooth terms. Generally,  $\gamma$  can be set as an average absolute intensity difference between neighboring pixels. The remaining parameter in formula (12) is  $\eta$  which is the weight of shape constraint. We fix it as 0.3 in all the experiments.

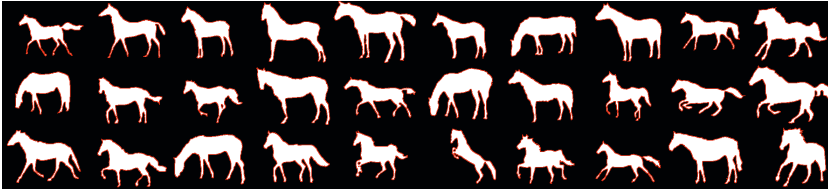


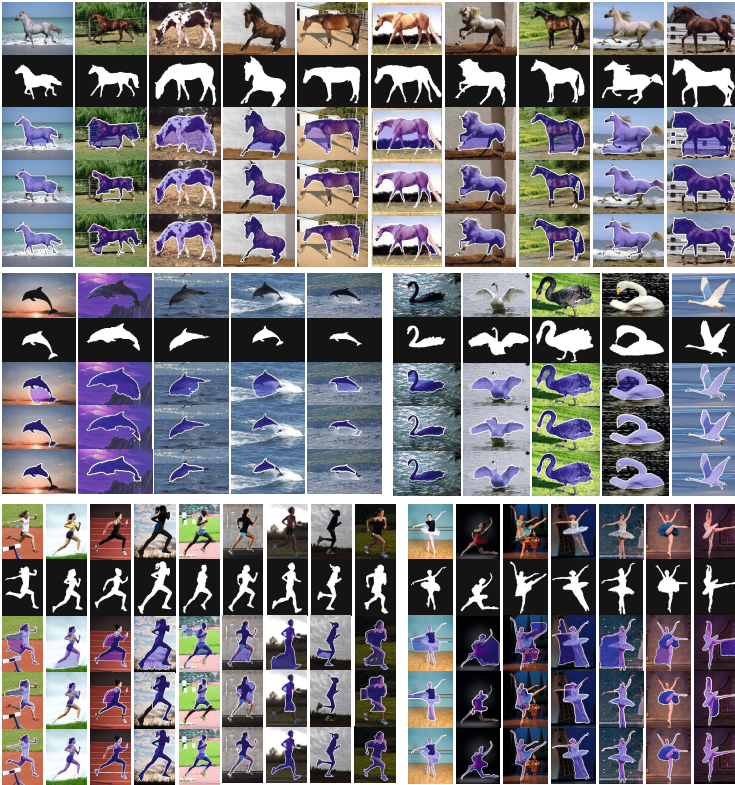
Fig. 7. The standard shape set for Weizmann\_horse dataset

### 5.3 Segmentation Results and Comparison

We firstly testify the performance of our proposed segmentation algorithm on the Weimann\_horse dataset and compare with the other segmentation algorithms in [3] and [13].

We firstly perform GrabCut method on the whole Weizmann\_horse dataset which consists of 328 side-view color images of horses, and here only a rectangle around the foreground object is provided for each image. And then, we pick out a subset of Weizmann\_horse dataset which is composed of 110 images that are not well segmented with GrabCut. Taking the shapes displayed in Fig. 7 as standard shape set, we segment the 110 horse images with our method. The result shows that the segmentations of 28 images keep unchanged compared with the GrabCut segmentations while the segmentations of the other 82 images are significantly improved. The total segmentation accuracy of the subset of Weizmann\_horse dataset increases from 0.6524 to 0.7608, which is defined as intersection-over-union score. We also compared our method with the shape sharing method in [13], whose segmentation accuracy on Weizmann\_horse dataset is 0.7477. Note that this approach provides a segmentation pool for each image, therefore we select the segmentation with best segmentation accuracy by comparing each proposal with the ground truth result. Some segmentation results are displayed in Fig. 8, where the first row shows the original images and the second row displays the corresponding ground truth segmentations. The experimental results of the methods in [3], [13] and our method are presented in the following three rows respectively. Additionally, comparison results on some other groups are also presented in this figure.

It can be noticed that the proposed algorithm achieves the best performance among the three approaches. It rectifies the error segmentation of GrabCut which is brought by the ambiguous object borders, such as the shadows on the ground. Compared with the shape sharing method, the proposed method performs better on the details, such as the thin and branch structures, horse legs for example. That is because the details are not well captured in the shared shapes which are provided by the BPLRs, and then they cannot provide constraint with high confidence. However, our shape priors have rich detail information thanks to the CPD registration which fixes the deviation between real shapes and the standard shapes from real world objects.



**Fig. 8.** Segmentation results of [3], [13] and the proposed method on Weizmann\_horse dataset. The original images are displayed in first and sixth rows, the corresponding ground truth are displayed in second and seventh rows. The following three rows display the experimental results of the methods in [3], [13] and our method, respectively.

## 6 Conclusion

In this paper, we introduce a novel shape constraint for object segmentation that collects a group of standard shapes for a specific object. To eliminate the individual differences between the standard shapes and the images to be segmented as well as determine the most appropriate shape constraint, we perform CPD registration on each standard shape and the initial shape which is extracted from the current segmentation to get the projections of the standard shapes. The score mechanism gives each mapped standard shape a quantitative evaluation about the appropriateness. And finally, the segmentation will be refined with the guide of the selected shape constraint. Compared with the traditional general shape prior which may not be specified enough to provide detailed guide information, our standard shape set prior which is composed of shapes with non-rigid transformation is able to provide more detailed constraint information

with high confidence, and that significantly reduces the dependency of manual interaction and handles the objects in complicated scenes well. The experimental results have verified the superior performance of the proposed method.

**Acknowledgment.** The research has been supported by the National Natural Science Foundation of China (61371140 and 61305044).

## References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
2. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: *Proc. Int. Conf. Computer Vision*, vol. 1, pp. 105–112 (2001)
3. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph* 23, 309–314 (2004)
4. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. In: *Proc. SIGGRAPH Conf.*, pp. 303–308 (2004)
5. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(6), 583–598 (1991)
6. Slabaugh, G., Unal, G.: Graph cuts segmentation using an elliptical shape prior. In: *IEEE International Conference on Image Processing*, vol. 2, pp. II-1222–II-1225 (2005)
7. Funke-Lea, G., Boykov, Y., Florin, C., Jolly, M.P., Moreau-Gobard, R., Ramaraj, R., Rinck, D.: Automatic heart isolation for ct coronary visualization using graph-cuts. In: *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 614–617 (2006)
8. Das, P., Veksler, O.: Semiautomatic segmentation with compact shape prior. In: *The 3rd Canadian Conference on Computer and Robot Vision*, pp. 28–28 (2006)
9. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
10. Veksler, O.: Star shape prior for graph-cut image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 454–467. Springer, Heidelberg (2008)
11. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 3129–3136 (2010)
12. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 1, pp. 755–762 (2005)
13. Kim, J., Grauman, K.: Shape sharing for object segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part VII*. LNCS, vol. 7578, pp. 444–458. Springer, Heidelberg (2012)
14. Myronenko, A., Song, X.: Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12), 2262–2275 (2010)
15. Dempster, A.P., Laird, N.M., Rubin, D.B., et al.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39(1), 1–38 (1977)



# Justifying Tensor-Driven Diffusion from Structure-Adaptive Statistics of Natural Images

Pascal Peter<sup>1</sup>, Joachim Weickert<sup>1</sup>, Axel Munk<sup>2</sup>,  
Tatyana Krivobokova<sup>3</sup>, and Housen Li<sup>2</sup>

<sup>1</sup> Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science,  
Campus E1.7, Saarland University, 66041 Saarbrücken, Germany

{peter,weickert}@mia.uni-saarland.de

<sup>2</sup> Felix-Bernstein-Chair for Mathematical Statistics, Institute of Mathematical Stochastics,  
Goldschmidtstrasse 7, 37077 Göttingen, Germany

munk@math.uni-goettingen.de, housen.li@stud.uni-goettingen.de

<sup>3</sup> Statistical Methods Group, Courant Research Centre “Poverty, Equity and Growth”,  
Wilhelm-Weber-Str. 2, 37073 Göttingen, Germany

tkrivob@uni-goettingen.de

**Abstract.** Tensor-driven anisotropic diffusion and regularisation have been successfully applied to a wide range of image processing and computer vision tasks such as denoising, inpainting, and optical flow. Empirically it has been shown that anisotropic models with a diffusion tensor perform better than their isotropic counterparts with a scalar-valued diffusivity function. However, the reason for this superior performance is not well understood so far. Moreover, the specific modelling of the anisotropy has been carried out in a purely heuristic way. The goal of our paper is to address these problems. To this end, we use the statistics of natural images to derive a unifying framework for eight isotropic and anisotropic diffusion filters that have a corresponding variational formulation. In contrast to previous statistical models, we systematically investigate structure-adaptive statistics by analysing the eigenvalues of the structure tensor. With our findings, we justify existing successful models and assess the relationship between accurate statistical modelling and performance in the context of image denoising.

**Keywords:** diffusion, regularisation, anisotropy, diffusion tensor, statistics of natural images, image priors.

## 1 Introduction

Anisotropic diffusion and regularisation models involve a positive definite  $2 \times 2$  matrix called diffusion tensor. Its eigenvalues steer the amount of data propagation in the direction of the corresponding eigenvector. Throughout more than two decades of research, such anisotropic methods have been successfully used for a large number of image processing and computer vision problems. These tasks include denoising [3], inpainting [24], image compression [6], optical flow computation [16], stereo reconstruction [31], and shape from shading [1]. Application domains cover e.g. computer aided quality control [25], medical image processing [14], and seismic image analysis [8].

To this date, modelling nonlinear diffusion filters is a heuristic, task-driven procedure, where images are processed towards a certain goal. It is a well-known fact that

anisotropic models can be much more powerful in certain applications than isotropic diffusion approaches with a scalar-valued diffusivity function. Clearly, one reason for the success of anisotropic concepts are their additional degrees of freedom which can be adapted to a task at hand. However, another potential explanation for this success is still unexplored: Could it be that smoothness assumptions of anisotropic models reflect statistical properties of natural images more accurately than isotropic ones?

For specific *isotropic* diffusion models, there exists a well-known connection to probabilistic filter models based on the statistics of natural images [29]: There is a negative logarithmic correspondence between natural image priors and regularisation terms in variational models. However, in particular for *anisotropic* diffusion, previous investigations have focused on isolated, specific models in practical contexts such as parameter learning. In particular, there is a lack of a cohesive theory that systematically analyses the correspondence between probabilistic filters and diffusion filters that can be expressed by energy minimisation.

**Our Contributions.** The goal of our paper is to provide a justification of tensor-driven diffusion models via the statistics of natural images. We aim at systematically assessing the differences between isotropic and anisotropic approaches from a probabilistic perspective. To this end, we use natural image priors to derive a unifying framework that incorporates eight existing diffusion filters that have a corresponding variational formulation. In order to cover the full range of nonlinear models, these statistics have to reflect the local image structure and allow to involve directional information. The eigenvalue statistics of the structure tensor in databases of natural images provide not only such information, but also offer a lot of flexibility to generate a wide range of derivative-based priors. This allows us to construct probabilistic filters that represent existing isotropic and anisotropic filter classes and analyse the differences in the underlying priors. We discuss the implications of these differences on filter performance in the context of image denoising.

**Related Work.** At its core, our work relies on the non-Gaussian nature of the histograms that result from applying filters to natural images. For wavelet coefficients, these specific attributes were first reported by Huang and Mumford [5]. These observations were systematically investigated for both derivative filters and wavelet coefficients in [10]. Invariances of these statistics are vital for their practical relevance. Zhu and Mumford proposed that these statistical priors are invariant to scale and verified this empirically in [29]. Evaluations on databases containing different motives were conducted by Huang and Mumford in [9]. For more details on the statistics of natural images, we refer to the recent monograph of Pouli et al. [19].

General connections between diffusion processes and statistical image processing models have been pioneered by Zhu and Mumford [29] within a Gibbs diffusion–reaction framework. Later on, Roth and Black [20] have found additional relations in the context of fields of experts. Works considering anisotropic diffusion models are, however, very rare. In the context of parameter learning, Scharr et al. [22] introduced an anisotropic model with Gaussian derivatives. A more recent parameter-free model goes back to Krajssek and Scharr [12]. They consider a two step procedure. In the first step, an isotropic diffusion process is derived. Afterwards, this is used to construct a *linear*

anisotropic regularisation model. More recently, Kunisch and Pock [13] have analysed parameter learning for regularisation methods with a bilevel optimisation scheme.

**Organisation of the Paper.** We start with a brief overview of existing tensor-driven diffusion in Section 2. In Section 3, we investigate the properties of the structure tensor as an image feature and use it to derive a probabilistic denoising filter in Section 4. We show that this model is related to a unifying framework for diffusion filtering in Section 5. In Section 6 we investigate diffusion models that are learned from a database, evaluate their performance for denoising and interpret the results. Finally, we present our conclusions and outlook on future work in Section 7.

## 2 Tensor-Driven Diffusion Processes

Let us start by reviewing a number of isotropic and anisotropic diffusion filters which can be derived from a general energy functional that we present in Section 5.

**General Structure.** Let  $\mathbf{f} = (f_1, \dots, f_m)^\top$  represent a vector-valued image with  $m$  channels. Each of these channels is a function  $f_k : \Omega \rightarrow \mathbb{R}$  that maps the rectangular image domain  $\Omega \subset \mathbb{R}^2$  to the colour value range  $\mathbb{R}$ . A tensor-driven, vector-valued diffusion process computes filtered versions  $\{\mathbf{u}(x, y, t) \mid (x, y) \in \Omega, t \geq 0\}$  of  $\mathbf{f}(x, y)$  as solutions of the diffusion equation

$$\partial_t u_k = \nabla^\top (D \nabla u_k) \quad \text{on } \Omega \times (0, \infty), \quad k = 1, \dots, m \quad (1)$$

with  $\mathbf{u}(x, y, 0) = \mathbf{f}(x, y)$  as initial condition on  $\Omega$ , and reflecting boundary conditions:

$$\langle D \nabla u_k, \mathbf{n} \rangle = 0 \quad \text{on } \partial\Omega \times (0, \infty), \quad k = 1, \dots, m. \quad (2)$$

The diffusion time  $t$  serves as a scale parameter: Larger times yield simpler image representations. The nabla operator  $\nabla$  and the divergence operator  $\nabla^\top$  involve spatial derivatives only, and  $\mathbf{n}$  denotes the outer normal vector to the image boundary  $\partial\Omega$ . The diffusion tensor  $D$  is a positive definite  $2 \times 2$  matrix that steers the diffusion. Its eigenvalues specify the amount of diffusion in the direction of the eigenvectors.

**Isotropic Models.** The simplest diffusion process, *homogeneous* diffusion [11], is obtained for  $D := I$  with a unit matrix  $I$ . In this case, the diffusion does not depend on the image structure. For more sophisticated *nonlinear isotropic* diffusion models the diffusion tensor is of the form  $D := g(|\nabla u|^2)I$ . If one wants to permit strong smoothing within homogeneous regions and inhibit smoothing across edges, one chooses the diffusivity  $g(|\nabla u|^2)$  as a decreasing positive function of its argument. Many diffusivity functions have been proposed, e.g. the Perona/Malik diffusivity  $g_{PM}$  [18] or the Charbonnier diffusivity  $g_C$  [2]:

$$g_{PM}(s^2) := \left(1 + \frac{s^2}{\lambda^2}\right)^{-1}, \quad g_C(s^2) := \left(1 + \frac{s^2}{\lambda^2}\right)^{-1/2}. \quad (3)$$

Note that locations where  $|\nabla u| \gg \lambda$  are regarded as edges where the diffusivity is close to 0, while we have full diffusion in regions with  $|\nabla u| \ll \lambda$ . Therefore,  $\lambda > 0$

acts as a contrast parameter. Isotropic models allow space-variant smoothing, but due to their scalar-valued diffusivity, the diffusion process acts in the same way in all directions. The first isotropic nonlinear model goes back to Perona and Malik [18] and is designed for greyscale images. Gerig et al. [7] have extended it to colour image processing by coupling the evolution of the individual channels through a diffusivity of the form  $g(\sum_{k=1}^m |\nabla u_k|^2)$ . Scherzer and Weickert [23] have investigated an isotropic nonlinear diffusion model where all spatial gradients  $\nabla$  are replaced by Gaussian-smoothed gradients  $\nabla_\sigma := K_\sigma * \nabla$ . Here  $K_\sigma$  denotes a Gaussian with standard deviation  $\sigma$ .

**Anisotropic Models.** In order to model direction-dependent diffusion processes, we need an anisotropic diffusion tensor  $D$  whose eigenvalues can differ significantly. These eigenvalues and their corresponding eigenvectors are adapted to the local image structure. A popular descriptor of the local image geometry is the structure tensor of Di Zenso [4]. In its most sophisticated form, it is given by the symmetric positive semidefinite matrix

$$\mathbf{J}_{m,\rho,\sigma} := K_\rho * \left( \sum_{k=1}^m \nabla_\sigma u_k \nabla_\sigma u_k^\top \right) \quad (4)$$

with eigenvalues  $\mu_{1,\rho,\sigma} \geq \mu_{2,\rho,\sigma} \geq 0$ . On greyscale images ( $m = 1$ ), the tensor  $\mathbf{J}_{m,0,0}$  without Gaussian smoothing has rank 1, while on colour images, it retains full rank in general. The corresponding diffusion tensor  $D := g(\mathbf{J}_{m,\rho,\sigma})$  uses the same set of eigenvectors and obtains its eigenvalues as functions of  $\mu_{1,\rho,\sigma}$  and  $\mu_{2,\rho,\sigma}$ . The anisotropic models of Weickert/Brox [27] and Tschumperlé/Deriche [24] do not incorporate any smoothing in the structure tensor (i.e.  $\sigma = \rho = 0$ ). However, such models degenerate to isotropic diffusion on greyscale images ( $m = 1$ ). The methods of Roussos/Maragos [21] and Scharr et al. [22] involve a smoothing scale  $\rho > 0$  and remain also anisotropic for  $m = 1$ . While Roussos/Maragos use  $\sigma = 0$ , Scharr et al. consider the case  $\sigma > 0$  and replace all gradients  $\nabla$  by their Gaussian-smoothed counterparts  $\nabla_\sigma$ .

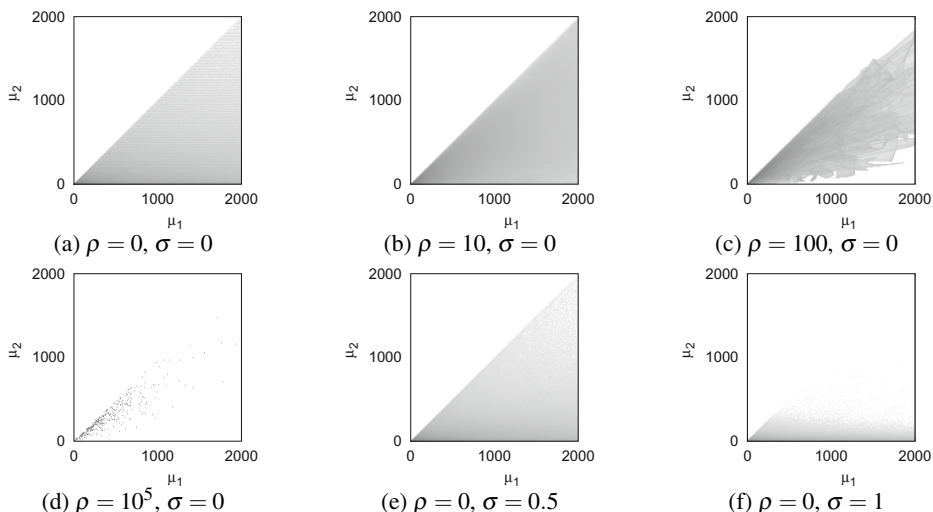
### 3 Structure-Adaptive Analysis of the Berkeley Database

**Interpretation of the Structure Tensor.** The local image structure of a vector-valued image  $u$  with  $m$  channels can be characterised by the joint structure tensor from Eq. (4). Its eigenvalues  $\mu_1 \geq \mu_2$  represent the local contrast in the direction of the corresponding eigenvectors  $v_1$  and  $v_2$ . For  $\mu_1 \gg \mu_2$ , the eigenvector  $v_2$  describes the direction of coherent structures while  $v_1$  points across these structures. Locally isotropic image content is characterised by  $\mu_1 \approx \mu_2$ . Thus, the eigenvalues of the structure tensor are image features that describe local geometry.

The Gaussian smoothing scales  $\sigma$  and  $\rho$  play distinct roles for the analysis of local image structure: Smoothing with  $K_\sigma$  removes noise and small-scale details. Thus, it should be chosen as small as possible. The smoothing scale  $\rho$  is usually chosen to be larger since its task is to accumulate neighbourhood information in the structure tensor.

In our implementation of the structure tensor, we use the finite difference discretisation from [28] with a parameter  $\alpha = 1/6$ . Its leading error term is rotationally invariant.

**Anisotropic Statistics of Colour Images.** Let us now use the aforementioned structure tensor for a statistical analysis of the Berkeley database [15]. The histogram of



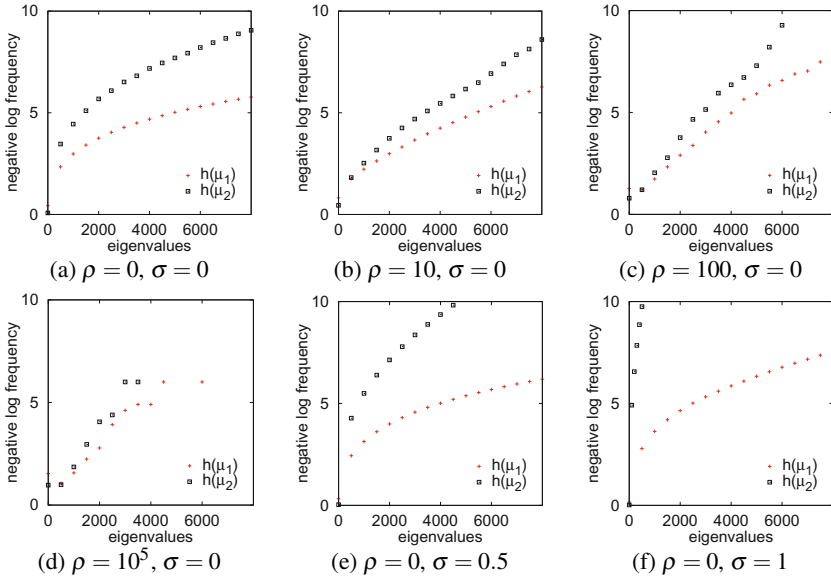
**Fig. 1.** Evolution of the negative logarithmic histogram of the eigenvalue pairs  $(\mu_{1,\rho,\sigma}, \mu_{2,\rho,\sigma})$  of the structure tensor  $\mathbf{J}_{m,\rho,\sigma}$  over different scales  $\rho$  and  $\sigma$ . Dark values indicate high occurrences and bright values low occurrences.

the eigenvalue pairs  $(\mu_1, \mu_2)$  with  $\sigma = \rho = 0$  is displayed in Fig. 1(a). The fact that the eigenvalue  $\mu_1$  clearly dominates and there are many structure tensors where  $\mu_2$  is significantly smaller confirms two things: Firstly, colour images contain many strongly oriented structures which legitimates the use of anisotropic filters. Secondly, these structures have some correlations over the colour channels. Fig. 2(a) reveals that both eigenvalues have the heavy-tailed distributions that are characteristic for filter results on natural images. Such kurtotic distributions are captured well by the function

$$\psi(x^2) = \frac{\lambda^2}{1-\gamma} \left(1 + \frac{x^2}{\lambda^2}\right)^{1-\gamma}. \quad (5)$$

The free parameters  $\lambda$  and  $\gamma$  can be adapted to fit  $\psi$  to the discrete histograms. A related model with one more degree of freedom was also proposed in [12]. Similar statistics have been shown to be nearly identical on many databases of natural images such as the Berkeley [15] or McGill [17] test sets. In particular, they are also invariant for image content on different scales. Therefore, they form a good prior for natural images. This scale invariance implies that the statistics do hardly change under subsampling.

**Behaviour under Smoothing.** If one averages with overlapping neighbourhoods, the statistics depend significantly on the neighbourhood size. This happens for the Gaussian-smoothed structure tensor  $\mathbf{J}_{m,\rho,\sigma}$ , where the tensor entries are embedded in a Gaussian scale-space. Let us first fix  $\sigma$  and consider the scale-space behaviour with respect to  $\rho$ . Fig. 1(a)–(d) shows the evolution of the histogram for the eigenvalue pairs  $(\mu_{1,\rho,\sigma}, \mu_{2,\rho,\sigma})$ . We observe that for increasing  $\rho$ , the joint histogram clusters towards



**Fig. 2.** Evolution of the negative logarithmic histograms  $h(\mu_1)$ ,  $h(\mu_2)$  of the eigenvalues  $\mu_{1,\rho,\sigma}$ ,  $\mu_{2,\rho,\sigma}$  of the structure tensor  $\mathbf{J}_{m,\rho,\sigma}$  over different scales  $\rho$  and  $\sigma$

the diagonal. This shows that  $\mu_{1,\rho,\sigma}$  and  $\mu_{2,\rho,\sigma}$  approach each other, i.e. the structure tensor becomes more isotropic. This is plausible, since one smoothes over structures with different orientations. For  $\rho \rightarrow \infty$ , all tensors  $\mathbf{J}_{m,\rho,\sigma}$  converge to the average structure tensor of the whole image. If all directions were equally prominent over the database, this average tensor would be purely isotropic. However, the steady state of the statistics ( $\rho = 10^5$  in Fig. 1(d) and Fig. 2(d)) reveals some anisotropy. Thus, the average eigenvalue histograms show the inherent directional bias of the image database.

Now we fix  $\rho$  and investigate the evolution under  $\sigma$ . For  $\sigma \rightarrow \infty$ , the local contrast given by  $\mu_{1,\rho,\sigma}$  and  $\mu_{2,\rho,\sigma}$  approaches 0 and the corresponding diffusion tensor  $\mathbf{D}$  converges to the unit matrix  $\mathbf{I}$ . Interestingly, Figs. 1(e)–(f) and 2(e)–(f) show that for small  $\sigma$ , the presmoothing increases the difference between the histograms of  $\mu_{1,\rho,\sigma}$  and  $\mu_{2,\rho,\sigma}$ . This fosters anisotropy of the image prior. We conjecture that Gaussian convolution effectively removes high-frequent isotropic perturbations, such that anisotropic image structures become more dominant. For larger  $\sigma$  their dominance decreases again.

In conclusion, we observe that natural images contain pronounced anisotropies and their statistics strongly depend on the smoothing scales  $\rho$  and  $\sigma$ . This suggest to design filters that take into account such anisotropic phenomena as priors.

### 4 Probabilistic Denoising with a Structure Tensor Prior

We can use the statistics from Section 3 as a prior for a Bayesian denoising approach. Let a discrete, noisy image  $\mathbf{f}$  of size  $M \times N$  with  $m$  channels be given. The goal is to compute an approximation  $\mathbf{u}$  to the original image  $\mathbf{v}$  under two assumptions:  $\mathbf{v}$  belongs

to the class of natural images and is degraded by Gaussian noise. For any image  $\mathbf{u}$ , let  $p(\mathbf{u})$  be the natural image prior. It describes the probability that  $\mathbf{u}$  is a natural image and is derived from the statistics of image features on a suitable database. Furthermore, an assumption on the distribution of the noise yields the noise prior  $p(\mathbf{f}|\mathbf{u})$ . According to Bayes' rule, the posterior probability for a candidate image  $\mathbf{u}$  to be the ground truth to an observed noisy image  $\mathbf{f}$  obeys

$$p(\mathbf{u}|\mathbf{f}) \sim p(\mathbf{f}|\mathbf{u}) \cdot p(\mathbf{u}). \quad (6)$$

Thus, the optimally denoised image  $\hat{\mathbf{u}}$  can be obtained by maximising the posterior probability  $p(\mathbf{u}|\mathbf{f})$  over all candidates  $\mathbf{u}$ :

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\operatorname{argmax}} p(\mathbf{u}|\mathbf{f}). \quad (7)$$

Since we assume independent identically distributed Gaussian noise for each channel  $k$  with  $k \in \{1, \dots, m\}$ , the noise prior is given by

$$p(\mathbf{f}|\mathbf{u}) \sim \prod_{k=1}^m \prod_{i=1}^M \prod_{j=1}^N \exp\left(-\frac{1}{2\sigma^2}(u_{k,i,j} - f_{k,i,j})^2\right). \quad (8)$$

In order to formulate a natural image prior, we follow the minimax entropy model that has been used to model texture [30] and whole images [29]. For a set of given linear or nonlinear filters  $\{\mathbf{F}_1, \dots, \mathbf{F}_L\}$  the distribution of natural images is modelled as

$$p(\mathbf{u}) = \prod_{\ell=1}^L \prod_{i=1}^N \prod_{j=1}^M \phi_{\ell}(\mathbf{F}_{\ell}(\mathbf{u})_{i,j}). \quad (9)$$

Here the potential functions  $\phi_{\ell}$  model the distribution of the corresponding filter  $\mathbf{F}_{\ell}$ . Current state-of-the-art models like the fields of experts approach [20] use specifically learned linear filters as a feature set. Interestingly many of these learned filters resemble derivative filters as was shown in [20].

Let  $\phi(\mu_1, \mu_2)$  define the distribution of an arbitrary image feature that is derived from the eigenvalues  $\mu_1$  and  $\mu_2$  of  $\mathbf{J}_{m,\rho,\sigma}$ . In particular, this formulation also includes separate statistics for both eigenvalues, i.e.  $\phi(\mu_1, \mu_2) := \phi_1(\mu_1) \cdot \phi_2(\mu_2)$ . Such image features can be interpreted as *second-level priors* in the terminology of [29], since they model the local geometry of image structures. In particular, these priors adapt to dominant directions in the image in contrast to linear filters that approximate derivatives in a fixed, global direction. By specifying the natural image prior (9) with a feature based on  $\mu_1$  and  $\mu_2$  and including the noise prior (8) we obtain the following energy:

$$E_P(\mathbf{u}) = \prod_{i=1}^M \prod_{j=1}^N \left( \prod_{k=1}^m \left( \exp\left(-\frac{(u_{k,i,j} - f_{k,i,j})^2}{2\sigma^2}\right) \right) \cdot \phi(\mu_{1,i,j}, \mu_{2,i,j}) \right). \quad (10)$$

Maximising  $E_P$  gives the denoised image  $\hat{\mathbf{u}}$ .

**Table 1. Existing diffusion models and their relation to the unifying framework.** The models primarily differ in the number of image channels  $m$ , the smoothing scale  $\rho$  for the structure tensor and the presmoothing scale  $\sigma$ . Additionally, most models impose certain restrictions to the general prior  $\phi(\mu_1, \mu_2)$ . The priors and penalisers are always given in the most general form that the corresponding model allows. Note that the Roussos/Maragos model has the same prior structure as Tschumperlé/Deriche, but yields a different PDE due to the nonzero smoothing scale  $\rho$ .

Model	$m$	$\sigma$	$\rho$	PDE	Penaliser	Prior
Homogeneous Diffusion [11]	1	0	0	$\partial_t u = \nabla^\top \nabla u$	$\psi(\mu_1 + \mu_2) =  \nabla u ^2$	$\phi(\mu_1 + \mu_2) = e^{- \nabla u ^2}$
Perona/Malik [18]	1	0	0	$\partial_t u = \nabla^\top (\psi'( \nabla u ^2) \nabla u)$	$\psi(\mu_1 + \mu_2) = -\ln \phi( \nabla u ^2)$	$\phi(\mu_1 + \mu_2) = \phi( \nabla u ^2)$
Gerig et al. [7]	$\geq 1$	0	0	$\partial_t u_k = \nabla^\top (\psi'(\sum_{\ell=1}^m  \nabla u_\ell ^2) \nabla u_k)$	$\psi(\mu_1 + \mu_2) = -\ln \phi(\sum_{\ell=1}^m  \nabla u_\ell ^2)$	$\phi(\mu_1 + \mu_2) = \phi(\sum_{\ell=1}^m  \nabla u_\ell ^2)$
Scherzer/Weickert [23]	1	$\geq 0$	0	$\partial_t u = \nabla^\top \sigma (\psi'( \nabla_\sigma u ^2) \nabla_\sigma u)$	$\psi(\mu_1 + \mu_2) = -\ln \phi( \nabla_\sigma u ^2)$	$\phi(\mu_1 + \mu_2) = \phi( \nabla_\sigma u ^2)$
Weickert/Brox [27]	$> 1$	0	0	$\partial_t u_k = \nabla^\top \left( (\psi'(\mu_1) v_1 v_1^\top + \psi'(\mu_2) v_2 v_2^\top) \nabla u_k \right)$	$\psi(\mu_{1,2}) = -\ln \phi(\mu_{1,2})$	$\phi(\mu_1) \cdot \phi(\mu_2)$
Tschumperlé/Deriche [24]	$> 1$	0	0	$\partial_t u_k = \nabla^\top \left( \left( \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_1} v_1 v_1^\top + \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_2} v_2 v_2^\top \right) \nabla u_k \right)$	$\psi(\mu_1, \mu_2) = -\ln \phi(\mu_1, \mu_2)$	$\phi(\mu_1, \mu_2)$
Roussos/Maragos [21]	$\geq 1$	0	$> 0$	$\partial_t u_k = \nabla^\top \left( K_\rho * \left( \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_1} v_1 v_1^\top + \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_2} v_2 v_2^\top \right) \nabla u_k \right)$	$\psi(\mu_1, \mu_2) = -\ln \phi(\mu_1, \mu_2)$	$\phi(\mu_1, \mu_2)$
Scharr et al. [22]	1	$\geq 0$	$> 0$	$\partial_t u = \nabla_\sigma^\top \left( K_\rho * (\psi'(\mu_1) v_1 v_1^\top + \psi'(\mu_2) v_2 v_2^\top) \nabla_\sigma u \right)$	$\psi_{1,2}(\mu_{1,2}) = -\ln \phi_{1,2}(\mu_{1,2})$	$\phi_1(\mu_1) \cdot \phi_2(\mu_2)$



## 5 The Unifying Prior-Based Diffusion Framework

Let us now show that the probabilistic denoising model (10) is the discrete counterpart to a unifying diffusion framework that incorporates a large family of existing diffusion approaches. Instead of maximising the energy  $E_P$ , we consider the minimisation of its negative logarithm

$$E_{\log}(\mathbf{u}) := \frac{1}{2} \sum_{i=1}^M \sum_{i=1}^N \left( \sum_{k=1}^m \frac{1}{\tau} (u_{k,i,j} - f_{k,i,j})^2 + \psi(\mu_{1,i,j}, \mu_{2,i,j}) \right). \quad (11)$$

Here, we define the penaliser  $\psi$  as  $\psi(\mu_1, \mu_2) = -\log \phi(\mu_1, \mu_2)$ , and we choose  $\tau \sim \sigma^2$ . A variational regularisation approach is obtained by the minimisation of the continuous counterpart to  $E_{\log}$ :

$$E(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \left( \frac{1}{\tau} |\mathbf{u} - \mathbf{f}|^2 + \psi(\mu_1, \mu_2) \right) dx dy \quad (12)$$

where  $|\cdot|$  denotes the Euclidean norm. Interestingly, this energy provides a unifying framework for the eight diffusion models from Section 2. The key result for understanding this connection is given by the following proposition.

**Proposition 1 (Euler-Lagrange Equations of the General Energy Functional).**

The energy functional  $E(\mathbf{u})$  from Eq. (12) gives rise to the Euler-Lagrange equations

$$\frac{u_k - f_k}{\tau} = \nabla_{\sigma}^{\top} ((K_{\rho} * D) \nabla_{\sigma} u_k), \quad k = 1, \dots, m, \quad (13)$$

with natural boundary conditions  $\mathbf{n}^{\top} (K_{\sigma} * K_{\rho} * D \nabla_{\sigma} u_k) = 0$  on  $\partial\Omega$ . Here,  $\mathbf{n}$  is the outer image normal and  $D$  is given in terms of the eigenvectors  $\mathbf{v}_1, \mathbf{v}_2$  and eigenvalues  $\mu_1, \mu_2$  of the structure tensor  $\mathbf{J}_{m,\sigma,\rho}$ :

$$D := \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_1} \mathbf{v}_1 \mathbf{v}_1^{\top} + \frac{\partial \psi(\mu_1, \mu_2)}{\partial \mu_2} \mathbf{v}_2 \mathbf{v}_2^{\top}. \quad (14)$$

*Proof.* The Euler-Lagrange equations are obtained from the Gâteaux derivatives of  $E(\mathbf{u})$ . We focus on the derivative of the penaliser  $\psi$ . With  $d_{\epsilon_k}(f) := \frac{\partial}{\partial \epsilon_k} f|_{\epsilon_k=0}$ ,  $k \in \{1, \dots, m\}$ , a test function  $\mathbf{h} : \mathbb{R}^2 \mapsto \mathbb{R}^m$ ,  $\mathbf{H} := \text{diag}(\mathbf{h})$ , and  $\epsilon \in \mathbb{R}^m$  we calculate:

$$d_{\epsilon_k}(\psi(\mu_1(\mathbf{u} + \mathbf{H}\epsilon), \mu_2(\mathbf{u} + \mathbf{H}\epsilon))) = \frac{\partial \psi}{\partial \mu_1} d_{\epsilon_k}(\mu_1) + \frac{\partial \psi}{\partial \mu_2} d_{\epsilon_k}(\mu_2). \quad (15)$$

Therefore, the derivatives of the eigenvalues  $\mu_1$  and  $\mu_2$  of  $\mathbf{J}_{m,\rho,\sigma}$  must be computed. In terms of the matrix elements  $J_{1,1}, J_{1,2}, J_{2,2}$ , the eigenvalue  $\mu_1$  is given by

$$\mu_1 = \frac{1}{2} \left( J_{1,1} + J_{2,2} + \sqrt{(J_{1,1} - J_{2,2})^2 + 4J_{1,2}^2} \right). \quad (16)$$

By writing the derivatives  $d_{\varepsilon_k}(J_{1,2})$ ,  $d_{\varepsilon_k}(J_{1,1} + J_{2,2})$ , and  $d_{\varepsilon_k}(J_{1,1} - J_{2,2})$  as dot products with  $\nabla_{\sigma}h_k$ , we can simplify  $d_{\varepsilon_k}(\mu_1)$  to

$$d_{\varepsilon_k}(\mu_1) = K_{\rho} * \left( (M \nabla_{\sigma} u_k)^{\top} \nabla_{\sigma} h_k \right), \tag{17}$$

$$M := \frac{2}{\mu_1 - \mu_2} \begin{pmatrix} \mu_1 - \mu_2 + J_{1,1} - J_{2,2} & 2J_{1,2} \\ 2J_{1,2} & \mu_1 - \mu_2 - J_{1,1} + J_{2,2} \end{pmatrix}. \tag{18}$$

Algebraic computations similar to [21] lead to  $M = 2v_1 v_1^{\top}$ . With analogous results for  $d_{\varepsilon_k}(\mu_2)$ , we obtain  $d_{\varepsilon_k}(\psi(\mu_1, \mu_2)) = (K_{\rho} * D \nabla_{\sigma} u_k)^{\top} \nabla_{\sigma} h_k$ . Plugging these results into the Gâteaux derivative  $d_{\varepsilon_k}E$  of the energy and applying partial integration yields

$$d_{\varepsilon_k}E = \sum_{\ell=1}^2 \left[ \left( K_{\sigma} * K_{\rho} * D \nabla_{\sigma} u_k \right)_{\ell} h_k \right]_{a_{\ell}}^{b_{\ell}} - \int_{\Omega} \nabla_{\sigma}^{\top} \left( (K_{\rho} * D) \nabla_{\sigma} u_k \right) h_k dx dy \tag{19}$$

with  $\Omega = [a_1, b_1] \times [a_2, b_2]$ . Variational calculus yields Eq. (13) and the natural boundary conditions  $\mathbf{n}^{\top} (K_{\sigma} * K_{\rho} * D \nabla_{\sigma} u_k) = 0$  on  $\partial\Omega$ .  $\square$

According to [23], Eq. (13) can be interpreted as an implicit time discretisation with one time step of size  $\tau$  of the general diffusion equation

$$\partial_t u_k = \nabla_{\sigma}^{\top} \left( (K_{\rho} * D) \nabla_{\sigma} u_k \right), \quad k = 1, \dots, m \tag{20}$$

with initial condition  $\mathbf{u}(t = 0) = \mathbf{f}$ . In Table 1 we demonstrate that a large number of existing diffusion models can be considered as special cases of this unifying partial differential equation. To see this, note that the isotropic models use  $\rho = 0$  and the prior

$$\phi(\mu_1 + \mu_2) = \phi(\text{tr} J_{m,0,\sigma}) = \phi \left( \sum_{\ell=1}^m |\nabla_{\sigma} u_{\ell}|^2 \right). \tag{21}$$

Moreover, for greyscale images ( $m = 1$ ) and smoothing scale  $\rho = 0$ , the structure tensor  $J_{1,0,\sigma} = \nabla_{\sigma} u \nabla_{\sigma} u^{\top}$  has the normalised eigenvectors  $\mathbf{v}_1 = \frac{\nabla_{\sigma} u}{|\nabla_{\sigma} u|}$  and  $\mathbf{v}_2 = \mathbf{v}_1^{\perp}$ . As a consequence, the diffusion process from Eq. (20) degenerates to isotropic diffusion with a scalar diffusivity: Using (14) we get

$$\begin{aligned} D \nabla_{\sigma} u &= \left( \frac{\partial \psi}{\partial \mu_1} \frac{\nabla_{\sigma} u \nabla_{\sigma} u^{\top}}{|\nabla_{\sigma} u|^2} + \frac{\partial \psi}{\partial \mu_2} \frac{\nabla_{\sigma} u^{\perp} \nabla_{\sigma} u^{\perp \top}}{|\nabla_{\sigma} u|^2} \right) \nabla_{\sigma} u \\ &= \frac{\partial \psi}{\partial \mu_1} \nabla_{\sigma} u = \psi'(|\nabla_{\sigma} u|^2) \nabla_{\sigma} u. \end{aligned} \tag{22}$$

Homogeneous diffusion is also captured by the model (20), if one chooses  $\phi(|\nabla u|^2) := \exp(-|\nabla u|^2)$  as prior distribution. The four anisotropic models are covered as follows: Weickert/Brox [27] and Schar et al. [22] use the factorised prior  $\phi_1(\mu_1) \cdot \phi_2(\mu_2)$ , in the case of Weickert/Brox with identical functions  $\phi_1$  and  $\phi_2$  and  $\sigma = \rho = 0$ . The models of Tschumperlé/Deriché [24] and Roussos/Maragos [21] allow general priors  $\phi(\mu_1, \mu_2)$ , but specify  $\sigma := 0$ . Moreover, Tschumperlé/Deriché also set  $\rho := 0$ .

The whole framework was derived from a common natural image prior, the directional statistics of the structure tensor. This shows that the ad hoc choices that were made for diffusion models during decades of research in fact reflect inherent properties of natural images. This observation can be even extended to the choice of diffusivities: If we consider the special case  $\phi(\mu_1, \mu_2) = \phi_1(\mu_1) \cdot \phi_2(\mu_2)$ , we are able to decompose  $\psi(\mu_1, \mu_2) := \psi_1(\mu_1) + \psi_2(\mu_2)$  into two separate penalisers  $\psi_\ell = -\ln \phi_\ell$  with  $\ell \in \{1, 2\}$ . The kurtotic distribution model (5) gives rise to the following family of diffusivities:

$$\psi'(x^2) = \left(1 + \frac{x^2}{\lambda^2}\right)^{-\gamma}. \quad (23)$$

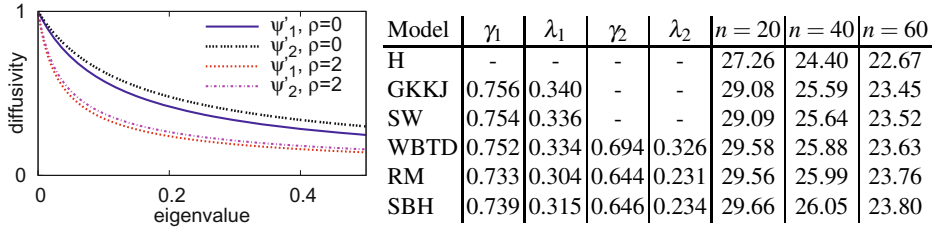
Comparing this to (3) shows that the Perona/Malik diffusivity [18] is covered for  $\gamma = 1$  and the Charbonnier diffusivity [2] results for  $\gamma = 0.5$ . To the best of our knowledge, our framework covers all relevant diffusion models that offer a variational interpretation. Since it is a variational framework, it is natural that it cannot be applied to models for which no variational formulation is known, e.g. edge- and coherence-enhancing diffusion filters [26].

## 6 Denoising Experiments

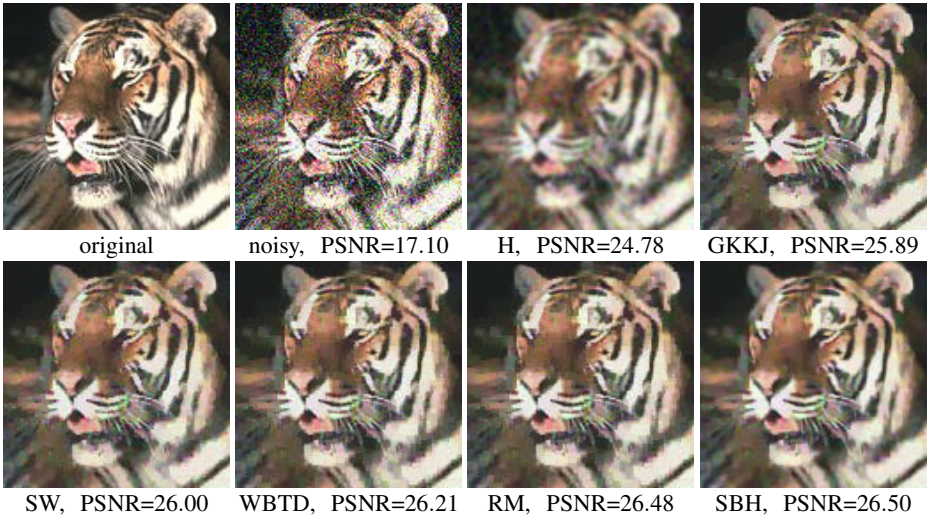
In the following, we compare the performance of different diffusion models in the context of image denoising. We focus on those models from Table 1 that are designed for colour images and apply small modifications where necessary: In analogy to [7], we extend the Scherzer/Weickert model to colour images by coupling the gradient within a joint diffusivity. Furthermore, we use separate penalisers  $\psi_1(\mu_1)$  and  $\psi_2(\mu_2)$  for the anisotropic models. This extends the Weickert/Brox model with individual diffusivities for both eigenvalues, which is a special case of the Tschumperlé/Deriche model. In the accompanying figures we use the abbreviations H for homogeneous diffusion [11], GKKJ for Gerig et al. [7], SW for Scherzer/Weickert [23], WBTD for the hybrid model of Weickert/Brox [27] and Tschumperlé/Deriche [24], RM for Roussos/Maragos [21], and SBH for a vector-valued extension of Scharr et al. [22].

For our experiments, we first determine the parameters  $\lambda$  and  $\gamma$  of the prior distribution (5) and the corresponding diffusivity (23). To this end, we compute the discrete histograms of  $\mu_1$  and  $\mu_2$  on the 200 training images of the Berkeley database [15]. For a nonlinear least squares fit to these histograms, we have chosen the Matlab implementation of the Levenberg–Marquardt algorithm (version 3.2.1 of the Matlab curve fitting toolbox). In Fig. 3(a) we see that the resulting diffusivities decrease more rapidly for  $\mu_1$  than for  $\mu_2$ . Thus, they inhibit diffusion across coherent structures more than along them. For increasing smoothing scales  $\sigma$  and  $\rho$  this anisotropic behaviour is reduced, since the difference between the diffusivities  $\psi'_1$  and  $\psi'_2$  is less pronounced. In the following, we use  $\rho = 0.5$  and  $\sigma = 0.2$ .

For our denoising experiments, we consider the partial differential formulation of the statistically-derived diffusion filters and apply them to the 100 images of the Berkeley test set [15] with added Gaussian noise. The average peak signal to noise (PSNR) values for different standard deviations of the noise are given in Fig. 3(b). We observe that for all noise levels, homogenous diffusion H yields the worst results, and the isotropic



**Fig. 3. (a) Left:** Diffusivities estimated for the eigenvalues  $\mu_1$  and  $\mu_2$  on the Berkeley database for different smoothing scales. **(b) Right:** Diffusivity parameters and denoising results for different diffusion models on the Berkeley test set. See Section 6 for the abbreviations. In the last three columns, the average PSNR for Gaussian noise with standard deviation  $n$  is given.



**Fig. 4.** Denoising results for image 108082 of the Berkeley test set with Gaussian noise of standard deviation  $n = 40$  for the models L, GKKJ, SW, WBTD, RM and SBH. The PSNR is given for the whole image, but only a zoom of size  $128 \times 128$  is shown.

methods GKKJ and SW perform consistently below the anisotropic models WBTD, RM and SBH. With increasing noise levels, the Gaussian smoothing scales  $\sigma$  and  $\rho$  within the models SW, RM and SBH offer a slight PSNR advantage over their counterparts GKKJ and WBTD that have to cope without Gaussian smoothing. Visually, the most distinct difference is the severe blurring of edges in homogenous diffusion that sets it apart from the other models.

Let us now interpret these findings from a probabilistic modelling perspective. The performance ranking according to the PSNR mirrors the accuracy of the underlying natural image priors. In particular, the large gap between homogenous diffusion and the rest of the models is caused by the wrongly assumed Gaussian-like distribution of the underlying image prior  $\mu_1 + \mu_2 = |\nabla u|^2$  in model H (see Tab. 1). Since all of the remaining filters accurately reproduce the kurtotic shape of the prior distributions, they

perform much better. Finally, the inherent directional bias in natural image models is only respected by the anisotropic models WBTD, RM and SBH, which gives them a consistent advantage over the isotropic models GKKJ and SW.

## 7 Conclusion and Outlook

We have presented a unifying framework for eight diffusion filters that have a corresponding variational formulation. It enabled us to derive these models from probabilistic filters with a structure tensor prior. We have verified experimentally that those filters which model the structure adaptive statistics of natural images more accurately also offer a better performance in practice. This justifies their use in digital image processing and computer vision, and it establishes a hitherto unknown reason for the success of anisotropic filters. From a statistical viewpoint, we have emphasised the importance of directional statistics that take into account the local image structure and its scale dependency. Interestingly, our statistical foundation of tensor-driven diffusion gives also additional insights that go beyond a pure statistical foundation of existing models: For instance, it sheds light on how the decay function of each eigenvalue should be adapted to the smoothing scales of the structure tensor.

Our results give rise to a number of ongoing and future activities. We are focussing our current research on anisotropic models that are tailored optimally to the statistics of natural images in a specific application context. Moreover, we expect that our framework can also be extended to novel energy functionals once they will be discovered for other important classes of anisotropic diffusion filters.

## References

1. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 578–591. Springer, Heidelberg (2006)
2. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: Proc. 1994 IEEE International Conference on Image Processing, vol. 2, pp. 168–172. IEEE Computer Society Press, Austin (1994)
3. Cottet, G.H.: Diffusion approximation on neural networks and applications for image processing. In: Hodnett, F. (ed.) Proc. Sixth European Conference on Mathematics in Industry, pp. 3–9. Teubner, Stuttgart (1992)
4. Di Zenzo, S.: A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing* 33, 116–125 (1986)
5. Field, D.J.: Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A* 4(12), 2379–2394 (1987)
6. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision* 31(2-3), 255–269 (2008)
7. Gerig, G., Kübler, O., Kikinis, R., Jolesz, F.A.: Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging* 11, 221–232 (1992)
8. Höcker, C., Fehmers, G.: Fast structural interpretation with structure-oriented filtering. *The Leading Edge* 21(3), 238–243 (2002)

9. Huang, J., Mumford, D.: Image statistics for the British Aerospace segmented database. Tech. rep., Divison of Applied Math, Brown University, Providence (1999)
10. Huang, J., Mumford, D.: Statistics of natural images and models. In: Proc. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1541–1547. IEEE Computer Society Press, Ft. Collins (1999)
11. Iijima, T.: Basic theory on normalization of pattern (in case of typical one-dimensional pattern). Bulletin of the Electrotechnical Laboratory 26, 368–388 (1962) (in Japanese)
12. Krajssek, K., Scharr, H.: Diffusion filtering without parameter tuning: Models and inference tools. In: Proc. 2010 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536–2543. IEEE Computer Society Press, San Francisco (2010)
13. Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences* 6(2), 938–983 (2013)
14. Manniesing, R., Viergever, M.A., Niessen, W.J.: Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical Image Analysis* 10, 815–825 (2006)
15. 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: Proc. Eighth International Conference on Computer Vision, Vancouver, Canada, pp. 416–423 (July 2001)
16. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 565–593 (1986)
17. Olmos, A., Kingdom, F.: A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception* 33(12), 1463–1473 (2004)
18. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 629–639 (1990)
19. Pouli, T., Reinhard, E., Cunningham, D.W.: *Image Statistics in Visual Computing*. CRC Press, Boca Raton (2013)
20. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 860–867. IEEE Computer Society Press, San Diego (2005)
21. Roussos, A., Maragos, P.: Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals. In: Proc. 17th IEEE International Conference on Image Processing, Hong Kong, pp. 4141–4144 (September 2010)
22. Scharr, H., Black, M.J., Haussecker, H.W.: Image statistics and anisotropic diffusion. In: Proc. Ninth International Conference on Computer Vision, vol. 2, pp. 840–847. IEEE Computer Society Press, Nice (2003)
23. Scherzer, O., Weickert, J.: Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision* 12(1), 43–63 (2000)
24. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(4), 506–516 (2005)
25. Weickert, J.: Anisotropic diffusion filters for image processing based quality control. In: Fasano, A., Primicerio, M. (eds.) Proc. Seventh European Conference on Mathematics in Industry, pp. 355–362. Teubner, Stuttgart (1994)
26. Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart (1998)
27. Weickert, J., Brox, T.: Diffusion and regularization of vector- and matrix-valued images. In: Nashed, M.Z., Scherzer, O. (eds.) *Inverse Problems, Image Analysis, and Medical Imaging, Contemporary Mathematics*, vol. 313, pp. 251–268. AMS, Providence (2002)
28. Welk, M., Steidl, G., Weickert, J.: Locally analytic schemes: A link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis* 24, 195–224 (2008)

29. Zhu, S.C., Mumford, D.: Prior learning and Gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(11), 1236–1250 (1997)
30. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision* 27(2), 107–126 (1998)
31. Zimmer, H., Valgaerts, L., Bruhn, A., Breuß, M., Weickert, J., Rosenhahn, B., Seidel, H.P.: PDE-based anisotropic disparity-driven stereo vision. In: Deussen, O., Keim, D., Saupe, D. (eds.) *Vision, Modelling, and Visualization 2008*, pp. 263–272. AKA, Heidelberg (2008)

# Variational Time-Implicit Multiphase Level-Sets

## A Fast Convex Optimization-Based Solution

Martin Rajchl<sup>1,2</sup>, John S.H. Baxter<sup>1</sup>, Egil Bae<sup>3</sup>, Xue-Cheng Tai<sup>4</sup>,  
Aaron Fenster<sup>1</sup>, Terry M. Peters<sup>1</sup>, and Jing Yuan<sup>1</sup>

<sup>1</sup> Robarts Research Institute, Western University, London, Canada

<sup>2</sup> Department of Computing, Imperial College London, London, UK

<sup>3</sup> Department of Mathematics, University of California, Los Angeles, United States

<sup>4</sup> Department of Mathematics, University of Bergen, Norway

**Abstract.** We propose a new principle, the *variational region competition*, to simultaneously propagate multiple disjoint level-sets in a fully time-implicit manner, minimizing the total cost w.r.t. region changes. We demonstrate, that the problem of multiphase level-set evolution can be reformulated in terms of a Potts problem, for which fast optimization algorithms are available using recent developments in convex relaxation. Further, we use an efficient recently proposed duality-based continuous max-flow method [1] implemented using massively parallel computing on GPUs for high computational performance. In contrast to conventional multi-phase level-set evolution approaches, ours allows for large time steps accelerating the evolution procedure. Further, the proposed method propagates all regions simultaneously, as opposed to the one-by-one phase movement of current time-implicit implementations. Promising experiment results demonstrate substantial improvements in a wide spectrum of practical applications.

**Keywords:** Level-Set, Multiphase, Image Segmentation, Convex Optimization, ASeTs.

## 1 Introduction

An important problem in the fields of image processing and computer vision is the identification of objects from 2D images or 3D volumes. The mean-curvature-driven evolution of contours, e.g 2D curves or 3D surfaces, has been established as a fundamental tool to address these problems. The ability to reliably propagate a contour,  $\mathcal{C}$ , with respect to optimization criteria or prior knowledge towards an object of interest within a given image is the subject of a wide spectrum of recent studies [2, 3]. Active contours [4], particularly their implementations via level sets, is capable of incorporating sophisticated energies from image intensities, shape models, and statistical criteria [5] while overcoming the main drawback of the classical formulation by allowing for topological change.

Despite the large number of successful applications of level-sets, there are two major drawbacks of the classical time-explicit implementations of the mean-curvature-driven level-sets evolution process: first, it requires a complex numerical scheme for the highly non-smooth second-order derivatives [6] and second, the



discrete time step-size must be small enough to achieve convergence resulting in low computational performance. In addition, it is cumbersome to extend the time-explicit evolution scheme for a single level-set to the case of multiphase level-sets, a great interest in the fields of image processing and computer vision.

In fact, if the applied embedding function of the conventional level-set is constrained to separate one region into two, it does not lead directly to the partition of multiple regions. To address this, several approaches have been proposed: Zhao et al. [7, 8] introduced a penalty to enforce region disjointness, that is:

$$\sum_{i=1}^n u_i(x) = 1; \quad u_i(x) \in \{0, 1\}, \quad \forall x \in \Omega, \tag{1}$$

where  $n$  is the number of regions represented by the indicator function  $u$  over the image domain  $\Omega$ .

Brox et al. [9] proposed an additional force term to enforce this constraint (1). Vese et al. [10] suggested that  $n = 2^m$  regions be represented by recursively splitting the domain to 2 subdomains  $m$  times, hence only  $m$  binary level sets were used to reduce the associated computational cost. However, these approaches [7, 8, 10, 9] result in more complicated numerical schemes and slow convergence.

### 1.1 Time-Implicit Level-Set Method

Recent studies [11–16] describe a method substantially distinct from the classical level-set approach, proposing fully time-implicit level-set schemes in terms of global optimization with advantages in both implementation and computation.

Luckhaus et al [12] and Boykov et al [13] first proved that, given the outer force  $f$  and mean-curvature  $\kappa$ , the mean-curvature-driven level-set problem:

$$\partial_t \mathcal{C} = -\kappa + f \tag{2}$$

can be solved iteratively, for each discrete time frame from  $t$  to  $t + h$ , by minimizing the variational energy [12]:

$$\mathcal{C}_{t+h} := \min_{\mathcal{C}} \int_{\partial \mathcal{C}} ds + \int_{\mathcal{C} \Delta \mathcal{C}_t} \frac{1}{h} \text{dist}(x, \partial \mathcal{C}_t) dx - \int_{\mathcal{C}} f dx, \tag{3}$$

where the function  $\text{dist}(x, \partial \mathcal{C}_t)$  denotes the distance from  $x$  to the region boundary  $\partial \mathcal{C}_t$ , and  $\mathcal{C}_t$  and  $\mathcal{C}_{t+h}$  is the respective position of the region at time  $t$  and  $t + h$ . This problem (3) can be expressed as:

$$\min_{\mathcal{C}} \int_{\partial \mathcal{C}} ds + \int_{\mathcal{C}} \left( \frac{1}{h} \text{sdist}(x, \partial \mathcal{C}_t) dx - f \right) dx, \tag{4}$$

where  $\text{sdist}(x, \partial \mathcal{C}_t)$  denotes the signed distance of  $x$  to  $\partial \mathcal{C}_t$ . More recent developments in convex optimization proved that the minimization problem (3) or (4)

can be solved exactly by means of the continuous min-cut [17, 18]. This indicates that during each discrete time-frame the level-set can be moved to the globally optimal position without further constraints on the time step-size  $h$ . This approach is entirely different from classical time-explicit ones, which are based on approximations which strictly require the time step-size to be sufficiently small as to converge. Another advantage of time-implicit approaches is the availability of fast global optimizers via convex optimization [18–20], or graph-cuts [21, 22].

In parallel, Boykov et al. [13] proposed the same variational principle (3) to the mean-curvature-driven level-set evolution and studied it under a discrete graph-cut perspective. Yuan et al. [15] investigated the proposed time-implicit evolution scheme of level-set introduced in [12] with help of continuous max-flow theory [18], which demonstrated that the global optimum to (3) or (4) is essentially the backward motion of (2), that is:

$$x = x_t - h(f - \kappa)(x) \mathbf{n}(x_t), \quad (5)$$

where the projection of any pixel  $x$  at the computed new boundary  $\partial\mathcal{C}_{t+h}$ , on the boundary  $\partial\mathcal{C}_t$ , is  $x_t$ , and  $\mathbf{n}(x_t)$  is the unit outward normal to  $\mathcal{C}_t$  at  $x_t$ . Obviously, (5) becomes to the equation of mean-curvature motion (2) as  $h \rightarrow 0$ .

It should be noted that Chambolle [14] also studied the mean-curvature driven motion (2) of contours with the force term  $f(x) = 0$ , which showed that at each discrete time frame, the next contour position  $\mathcal{C}_{t+h}$  can be obtained by the zero level set of the total-variation regularized signed distance function  $\text{sdist}(x, \partial\mathcal{C}_t)$  w.r.t.  $\mathcal{C}_t$ , with the backward motion scheme

$$x = x_t - h\kappa(x)\mathbf{n}(x_t), \quad (6)$$

where is equal to (5) given  $f(x) = 0$ . Bresson & Chan [23] extended Chambolle’s work [14] to the case of geodesic level-set evolution with region forces.

Despite the advantages of the time-implicit level-set methods in both theory and implementation, very few studies have dealt with the propagation of multiple level-sets in a fully time-implicit style. One interesting approach was proposed by Yuan et al. [16], which introduced a global optimization scheme for the evolution of multiple level-sets  $\mathcal{C}_i$ ,  $i = 1 \dots n$ , preserving a linear order over said level-sets:

$$\mathcal{C}_n \subset \dots \subset \mathcal{C}_1.$$

Yuan et al. [16] demonstrated that such level-sets can be simultaneously moved to their globally best positions in a fully time-implicit manner.

## 1.2 Contributions

In this work, we study the evolution of multiphase level-sets by means of global optimization. We propose a novel principle, the *variational region competition*, to jointly propagate multiple disjoint level-sets by minimizing the total cost w.r.t. region changes. We show that the variational time-implicit scheme [11, 12, 15] for a single level-set is just a special case of the introduced *variational*

*region competition* principle. In addition, we prove that the reduced optimization problem can be expressed as Potts problem, where fast optimization solvers are available via convex relaxation [19, 24, 1, 20] and graph-cuts [21, 22]. To address the resulting optimization problem, we make use of the fast duality-based continuous max-flow method developed in [1], which is implemented using a massively parallel computing architecture (GPU) for high performance.

In contrast to classical approaches to multiphase level-sets, the new global optimization-based multiphase approach is fully time-implicit, allowing large step-sizes for contour propagation which ultimately improve efficiency. In addition, it propagates the level-sets simultaneously, instead of one-by-one phase movement of the conventional level-set implementation [6]. Moreover, the convex relaxation solver of the introduced Potts problem approximates the global optimum well in practice, at least within some bound [25]. Promising experimental results show the proposed time-implicit multiphase level-set method substantially improves results in both efficiency and reliability for different applications.

## 2 Variational Time-Implicit Multiphase Level-Sets

In this section, we study the evolution of multiple mean-curvature-driven contours with respect to a disjointness constraint, for which we propose a novel variational principle, i.e. the *variational region competition*. The proposed *variational region competition* generalizes recent developments in level-set methods and establishes a variational basis for simultaneously propagating multiple disjoint level-sets by means of minimizing costs w.r.t. region changes. We show that previous approaches for single level-sets under via a minimum cost of region changes w.r.t. foreground and background, e.g. [12, 13] and recently [15], are a special case of the proposed theory.

The proposed principle can be reformulated as a spatially continuous Potts problem [26], i.e. a continuous multi-region min-cut problem, which we study here via convex relaxation under a continuous max-flow perspective.

### 2.1 Principle of Variational Region Competition

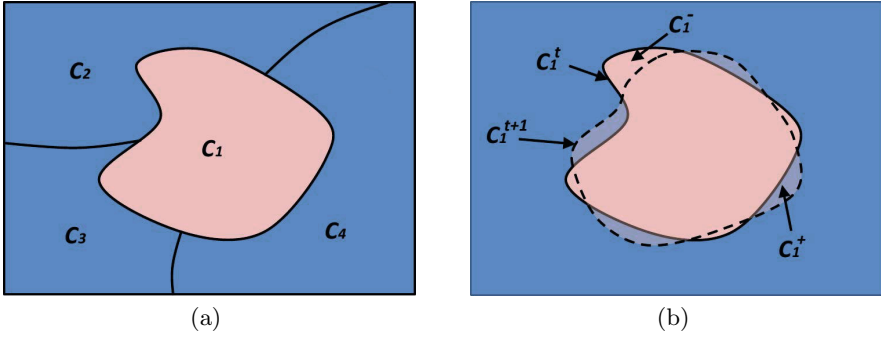
Consider the evolution of  $n$  regions,  $\mathcal{C}_i$ ,  $i = 1 \dots n$ , under the constraint:

$$\Omega = \cup_{i=1}^n \mathcal{C}_i, \quad \mathcal{C}_k \cap \mathcal{C}_l = \emptyset, \quad \forall k \neq l. \quad (7)$$

Let  $\mathcal{C}_i^t$ ,  $i = 1 \dots n$ , be the  $i$ -th region at the current time frame  $t$ , which moves to position  $\mathcal{C}_i^{t+1}$  at the next time frame  $t+1$ . For each region  $\mathcal{C}_i^t$  at time  $t$ , we define two types of difference regions with respect to  $\mathcal{C}_i^{t+1}$  (see Fig.1 for illustration):

1.  $\mathcal{C}_i^+$  indicates expansion of  $\mathcal{C}_i^t$  w.r.t.  $\mathcal{C}_i^{t+1}$ : for  $\forall x \in \mathcal{C}_i^+$ , it is outside  $\mathcal{C}_i^t$  at time  $t$ , but inside  $\mathcal{C}_i^{t+1}$  at  $t+1$ ; for such an expansion of  $x$ , with cost  $c_i^+(x)$ .
2.  $\mathcal{C}_i^-$  indicates shrinkage of  $\mathcal{C}_i^t$  w.r.t.  $\mathcal{C}_i^{t+1}$ : for  $\forall x \in \mathcal{C}_i^-$ , it is inside  $\mathcal{C}_i^t$  at time  $t$ , but outside  $\mathcal{C}_i^{t+1}$  at  $t+1$ ; for such a shrinkage of  $x$ , with cost  $c_i^-(x)$ .

With these definitions, we propose the variational principle as:



**Fig. 1.** A simple example of the evolution of 4 regions: (a) shows the 4 disjoint regions at the current time frame  $t$ ; (b) depicts the evolution of  $C_1^t$  from discrete time  $t$  to the next  $t + 1$ , i.e.  $C_1^{t+1}$ , which depicts region expansion  $C_1^+$  and shrinkage  $C_1^-$

**Variational Region Competition Principle 1.** For  $n$  disjoint regions  $C_i$ ,  $i = 1 \dots n$ , the evolution of each region over the discrete time frame from  $t$  to  $t + 1$  minimizes total cost of region changes. That is, the new optimal contours  $C_i^{t+1}$ ,  $i = 1 \dots n$ , minimize the energy:

$$\min_{C_i} \sum_{i=1}^n \left\{ \int_{C_i^-} c_i^-(x) dx + \int_{C_i^+} c_i^+(x) dx \right\} + \sum_{i=1}^n \int_{\partial C_i} g(s) ds \quad (8)$$

subject to (7), where  $g(s)$  is a weighting function acting as a cost along the contour boundaries.

We describe special applications of the proposed *variational region competition* principle in the following subsections:

**Application to Single Level-Set Evolution.** The mean-curvature-driven motion of the single contour  $C$ :

$$\partial_t C = -\kappa,$$

with  $C^t$  be the current level set. We define the cost functions  $c^-(x)$  and  $c^+(x)$  of region changes w.r.t.  $C_1^t = C^t$ , which are linear to the distance from  $x$  to its boundary  $\partial C^t$ , i.e.

$$c_1^-(x) = c_1^+(x) = \text{dist}(x, \partial C^t) / h, \quad (9)$$

where  $h > 0$  is constant.

This can be represented as the case of evolving two disjoint regions, i.e. region  $C_1 = C$  and its complementary region  $C_2 = \Omega \setminus C$ . Since the two regions  $C_1^t = C^t$  and  $C_2^t = \Omega \setminus C^t$  are complementary to each other, the region shrinkage of  $C_1^t$  corresponds to the region expansion of  $C_2^t$  and vice versa. Hence, the cost functions satisfy

$$c_1^-(x) = c_2^+(x), \quad c_1^+(x) = c_2^-(x).$$

Given the *variational region competition* principle (8) and

$$\int_{\partial\mathcal{C}_1(:=\mathcal{C})} ds = \int_{\partial\mathcal{C}_2(:=\Omega\setminus\mathcal{C})} ds = \int_{\partial\mathcal{C}} ds,$$

we can derive

$$\mathcal{C}^{t+h} := \min_{\mathcal{C}} \int_{\partial\mathcal{C}} ds + \int_{\mathcal{C}\Delta\mathcal{C}^t(:=\mathcal{C}_1^+\cup\mathcal{C}_1^-)} \frac{1}{h} \text{dist}(x, \partial\mathcal{C}^t) dx, \tag{10}$$

which is the variational formulation (3) with force  $f(x) = 0$ , proposed in [12]. It is also straight-forward to extend it to cases with a non-zero force term  $f(x) \neq 0$ .

**Application to Multiphase Level-Set Evolution.** Likewise, for the mean-curvature-driven evolution of multiple disjoint level-sets  $\mathcal{C}_i, i = 1 \dots n$ , we define the cost functions  $c_i^-(x)$  and  $c_i^+(x), i = 1 \dots n$ , to be proportional to the distance function from  $x$  to the current boundary  $\partial\mathcal{C}_i^t$  such that

$$c_i^-(x) = c_i^+(x) = \text{dist}(x, \partial\mathcal{C}_i^t)/h, \quad i = 1 \dots n. \tag{11}$$

Using the *variational region competition* principle (8), we have:

**Corollary 2.** *The mean-curvature-driven evolution of multiple disjoint level-sets  $\mathcal{C}_i, i = 1 \dots n$ , during time frame  $t$  to  $t + 1$  minimizes the cost w.r.t. region changes. The optimal new regions  $\mathcal{C}_i^{t+1}, i = 1 \dots n$ , therefore minimize:*

$$\min_{\mathcal{C}_i} \sum_{i=1}^n \int_{\mathcal{C}_i\Delta\mathcal{C}_i^t} \frac{1}{h} \text{dist}(x, \partial\mathcal{C}_i^t) dx + \sum_{i=1}^n \int_{\partial\mathcal{C}_i} ds \tag{12}$$

subject to the constraint (7).

**Application to Multiphase Image Segmentation.** For multiphase image segmentation, the level-set evolution is driven not only by the distance functions as above, but also by image features. In general, the cost functions  $c_i^-(x)$  and  $c_i^+(x), i = 1 \dots n$ , w.r.t. region changes are given by the combination of the image feature costs and the distance functions, i.e.

$$c_i^-(x) = \omega_1 f_i^-(x) + \omega_2 \frac{1}{h} \text{dist}(x, \partial\mathcal{C}_i^t), \quad \forall x \in \mathcal{C}_i^t, \text{ and}$$

$$c_i^+(x) = \omega_1 f_i^+(x) + \omega_2 \frac{1}{h} \text{dist}(x, \partial\mathcal{C}_i^t), \quad \forall x \notin \mathcal{C}_i^t;$$

where the weighting parameters are  $\omega_1, \omega_2 > 0, \omega_1 + \omega_2 = 1$ , and the cost functions  $f_i^-(x)$  and  $f_i^+(x), i = 1 \dots n$ , are derived according to the specified image features. The corresponding optimization formulation is then given by the *variational region competition* principle (8) directly, which is slightly different from (11). For the purpose of our experimental setup we define

$$f_i^-(x) = (I(x) - l_i)^2, \quad \forall x \in \mathcal{C}_i, \text{ and}$$

$$f_i^+(x) = (I(x) - l_i)^2, \quad \forall x \notin \mathcal{C}_i,$$

where  $l_i$  is the mean intensity  $I$  inside the contour  $\mathcal{C}_i, i = 1 \dots n$ .

### 2.2 Spatially Continuous Potts Model

In this section, we show that the variational problem (8) introduced by the *variational region competition* principle can be equally reformulated as the Potts problem [26]. For this purpose, we define two cost functions  $D_i^s(x)$  and  $D_i^t(x)$  w.r.t. the current contour  $\mathcal{C}_i^t$ ,  $i = 1 \dots n$ , at time  $t$ :

$$D_i^s(x) := \begin{cases} c_i^-(x), & \text{where } x \in \mathcal{C}_i^t \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$D_i^t(x) := \begin{cases} c_i^+(x), & \text{where } x \notin \mathcal{C}_i^t \\ 0, & \text{otherwise} \end{cases} . \tag{14}$$

Let  $u_i(x) \in \{0, 1\}$ ,  $i = 1 \dots n$ , be the indicator function of the region  $\mathcal{C}_i$ . Therefore, the disjoint constraint in (7) can be represented by

$$\sum_{i=1}^n u_i(x) = 1; \quad u_i(x) \in \{0, 1\}, \quad \forall x \in \Omega. \tag{15}$$

Via the cost functions (13) and (14), we can prove

**Proposition 3.** *The variational formulation (8) associated with the variational region competition principle can be expressed as the Potts problem*

$$\min_{u_i(x) \in \{0,1\}} \sum_{i=1}^n \langle u_i, D_i^t - D_i^s \rangle + \sum_{i=1}^n \int_{\Omega} g(x) |\nabla u_i| dx \tag{16}$$

subject to the contour disjointness constraint (15), where the weighted length term in (8) is encoded by the weighted total-variation functions.

*Proof.* To see the equivalence between the two optimization problems (8) and (16), we first consider the total cost related to the region changes of the contour  $\mathcal{C}_i$ ,  $i = 1 \dots n$ , i.e.

$$\int_{\mathcal{C}_i^-} c_i^-(x) dx + \int_{\mathcal{C}_i^+} c_i^+(x) dx. \tag{17}$$

In view of the new cost functions (13) and (14), the above total cost (17) can be equally written as

$$\langle 1 - u_i, D_i^s \rangle + \langle u_i, D_i^t \rangle. \tag{18}$$

Summing (18) over the  $n$  contours, we obtain

$$\sum_{i=1}^n \left\{ \langle 1 - u_i, D_i^s \rangle + \langle u_i, D_i^t \rangle \right\} = \sum_{i=1}^n \langle u_i, D_i^t - D_i^s \rangle + \sum_{i=1}^n \int_{\Omega} D_i^s(x) dx \tag{19}$$

where the last term is constant.

Also, we can formulate the weighted perimeter term in (8) by means of the total variation function such that

$$\int_{\partial \mathcal{C}_i} g(s) ds = \int_{\Omega} g(x) |\nabla u_i| dx. \tag{20}$$

By combining (19) and (20), the equivalence between (8) and (16) is proved.

**Potts Formulation to Multiphase Level-Set Evolution.** From the mean-curvature-driven evolution of multiple disjoint level-sets  $\mathcal{C}_i, i = 1 \dots n$ , the cost functions (13) and (14), we have the functions  $D_i^s(x)$  and  $D_i^t(x), i = 1 \dots n$ , by the definition of (11). The difference  $D_i^t(x) - D_i^s(x), i = 1 \dots n$ , defines the respective signed distance functions

$$D_i^t(x) - D_i^s(x) = \frac{1}{h} \text{sdist}(x, \partial \mathcal{C}_i^t) = \begin{cases} -\text{dist}(x, \partial \mathcal{C}_i^t)/h, & \text{where } x \in \mathcal{C}_i^t \\ \text{dist}(x, \partial \mathcal{C}_i^t)/h, & \text{otherwise} \end{cases} \quad (21)$$

Invoking Prop. 3, we have:

**Corollary 4.** *The variational problem (12) of the mean-curvature-driven evolution of multiple disjoint level-sets  $\mathcal{C}_i, i = 1 \dots n$ , can be identically formulated as the Potts problem with respect to the minimum total signed distances:*

$$\min_{u_i(x) \in \{0,1\}} \sum_{i=1}^n \frac{1}{h} \langle u_i, \text{sdist}(x, \partial \mathcal{C}_i^t) \rangle + \sum_{i=1}^n \int_{\Omega} g(x) |\nabla u_i| \, dx \quad (22)$$

subject to the contour disjointness constraint (15).

### 2.3 Continuous Max-flow Approach and Dual Optimization

The resulting formulation (16) gives rise to a challenging combinatorial optimization problem. From recent developments of convex optimization [19, 24, 1, 20], its global optimum can be approximated efficiently through convex relaxation, i.e.

$$\min_{u(x) \in \Delta_+} \sum_{i=1}^n \langle u_i, D_i^t - D_i^s \rangle + \sum_{i=1}^n \int_{\Omega} g(x) |\nabla u_i| \, dx \quad (23)$$

where  $\Delta_+$  is the simplex set

$$\text{for } \forall x \in \Omega, \quad \sum_{i=1}^n u_i(x) = 1; \quad u_i(x) \in [0, 1], \quad i = 1 \dots n.$$

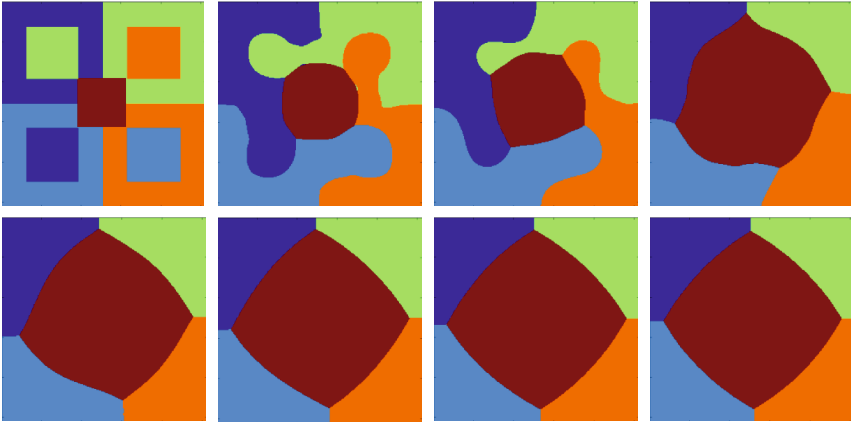
The minimization problem (23) is a special case of that studied in [1] where  $\rho(i, x) = D_i^t - D_i^s$ . To solve (23), we use an efficient algorithm proposed in [1], which solves a max-flow formulation of (23) by the augmented Lagrangian method. We refer to [1] for details.

## 3 Experiments

In this section we conduct a series of numerical experiments to assess several aspects of the performance of the proposed method. We demonstrate the ability to evolve multiple contours in a synthetic example converging to the minimum-length partition between all segments. Furthermore, we implement the proposed method in a massively parallelized manner and test it against a sequential implementation on a single core. We also compare both implementations, in terms of run times and convergence, against the classical multi-region level-set evolution. Finally, we demonstrate its applicability in 3D medical volume segmentation.

### 3.1 Experiments of Level-Set Evolution and Image Segmentation

By setting up the energy terms proposed in Coro. 4, we minimize the variational problem in (22) with dual optimization. We employed the initialization in Figure 2 to demonstrate this behavior on five regions. The contour evolves from an initial higher energetic state towards the optimum, where the total perimeter of all partitions is minimized. Figure 2 depicts the course of 100 iterations from the initialization (top left) to the final evolution status. Fig. 3 shows two examples of multi-region image segmentation by the proposed method given in Sec. 2.1.



**Fig. 2.** Five initial regions (top left) are propagated via the proposed contour evolution method. From left to right:  $t = 0, 1, 4, 10, 25, 50, 100$ ,  $h = 25$ ,  $g(x) = 10$ .

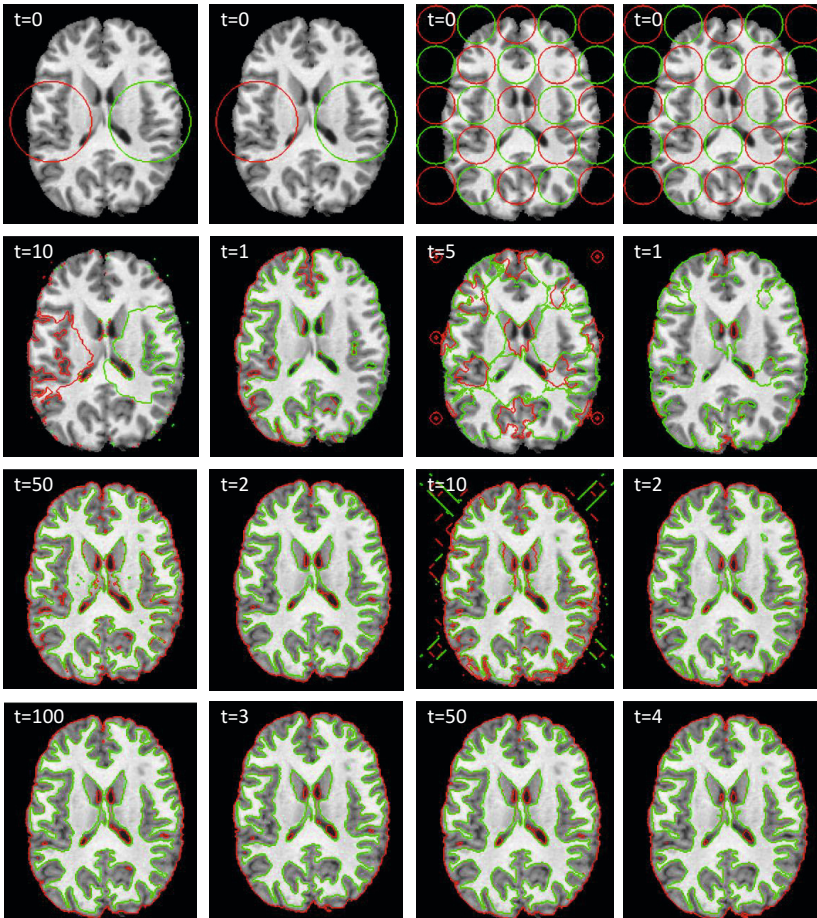


**Fig. 3.** Supervised example segmentation of two natural images from the Berkeley database with 6 and 3 regions, respectively. The contours are initialized via seeds placed by a user and convergence achieved if less than 50 pixels change between iterations. From left to right: Image and corresponding segmentation.



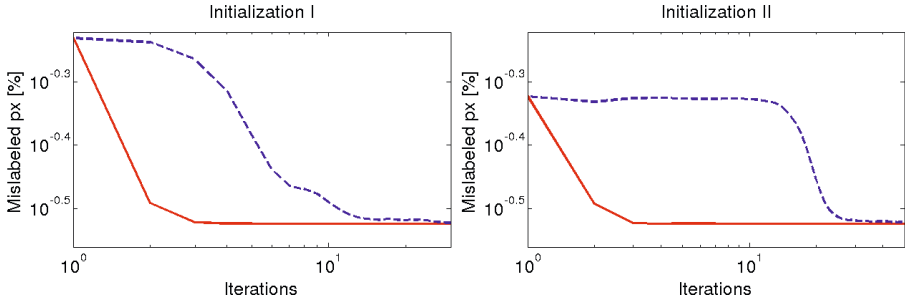
### 3.2 Comparison against Classical Multiphase Level-Sets

We tested the proposed method against a recent classical multi-region level set (MRLS) implementation [27] for segmentation of brain tissue in a T1-weighted magnetic resonance image, using two different initializations (Figure 4, row 1). We defined the three regions (red, green and background regions) as converged if fewer than 5 pixels change over an iteration. We used the same cost function (from section 2.1) in both methods and individually adjusted parameters for rapid convergence. The proposed method converged after 3 and 4 iterations (column 2 & 4) and MRLS after 100 and 50 iterations (column 1 & 3).



**Fig. 4.** Comparison of the proposed method (column 2 & 4) against MRLS [27] (column 1 & 3) for unsupervised brain tissue segmentation from a magnetic resonance image. Two initializations (row 1), and propagation over time (rows 2-4) are shown.

Additionally, we calculated the percentage of mislabeled pixels against a ground truth from manual segmentation at each iteration for both methods and initialization patterns (Fig. 5). The lowest mislabeling rate is reached far earlier by the proposed method than by MRLS, due to the larger time step-sizes allowed.



**Fig. 5.** Log-log plotted percentage of mislabeled pixels with time steps corresponding to Fig. 4. Initialization I (row 1, column 1 & 2) and II (row 1, column 3 & 4) for the proposed method (red) and MRLS [27] (blue).

### 3.3 Computation Performance and Parallelized Implementation

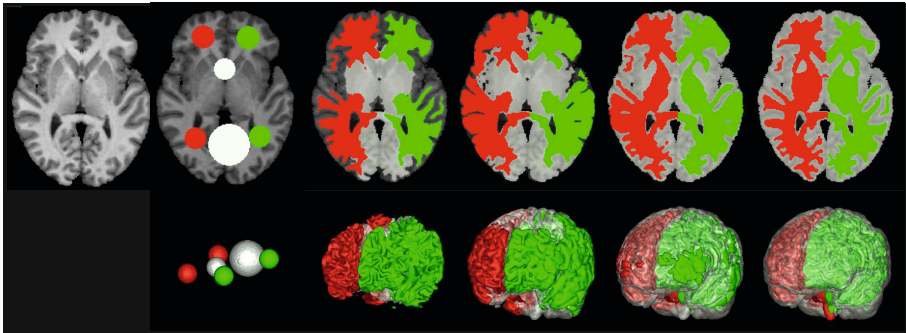
We employ General-Purpose Programming on Graphics Processing Units (GPGPU) via the CUDA (v6.0, NVIDIA, Santa Clara, CA) architecture to parallelize the underlying continuous max-flow optimization and test its impact on overall run times against the C++ implementation. Table 1 summarizes the run times of the experiments in Figure 4 using a Ubuntu 64-bit workstation, with 144GB memory and an NVIDIA Tesla C2070 (6GB) graphics card.

**Table 1.** Run time experiments for the brain example in Figure 4 until convergence at different resolutions: 202x170 (original image), 256x256 and 512x512 px

Method	Architecture	Dimensions	Iterations	Total run time [s]
Proposed	CPU	202x170	3	3.9
		256x256	3	5.9
		512x512	3	8.0
	GPU	202x170	3	<b>0.4</b>
		256x256	3	<b>0.5</b>
		512x512	3	<b>0.9</b>
Ben Ayed et al. [27]	CPU	202x170	100	22.9
		256x256	100	24.7
		512x512	100	33.1

### 3.4 Application to 3D Medical Image Segmentation

Fig. 6 demonstrates an application in 3D medical image segmentation, specifically of multiple brain tissue regions. In contrast to the complex implementation and low performance of classical 3D level-sets, the 3D implementation of the proposed multiphase level-sets is simple and fast, in particular when using modern parallelized computing platforms. The example took 8 iterations to converge to the segmentation shown (202x170x158 voxels).



**Fig. 6.** Application in 3D brain MRI segmentation. From left to right: the image, initialization, iteration 1,2,4 and 8. Axial slices (top) and surface renderings (bottom) are shown. Computation took 265.7 s (202x170x158 voxels) on GPU.

## 4 Discussion and Conclusions

We propose a new time-implicit multiphase level-set evolution method based on the spatially continuous Potts model, and demonstrate its performance and applicability in image processing. Due to the implicit evolution of contours to optimal positions at each time step, which allows for large step-sizes, the proposed method leads to a simple algorithmic scheme taking advantage of recent developments in convex optimization. The proposed variational method and the derived Potts problem could also be addressed by discrete optimization methods, e.g. graph-cuts [21, 22], but solving the Potts formulation in a spatially continuous setting [18–20] successfully avoids metrification artifacts and can be easily implemented on GPUs to significantly improve computational efficiency.

We compare the proposed variational time-implicit level-sets method against the classical multi-region level-set implementation used in [27]. Numerical results in Table 1 demonstrate that the proposed method converges quicker and with fewer iterations, and both its sequential and parallelized implementations outperform the slower and more complex classical approach, [27]. Convergence rates of both compared methods for the problem in Fig. 4 are depicted in log-log plots in Fig. 5. We demonstrate that the proposed method not only converges with fewer iterations, but reaches the same mislabeling error as classical MLS given the same initialization. This benefits 3D/4D medical imaging applications

(see Fig. 6 as an example) in particular, which often require rapid computation as not to impede clinical workflow. Future directions potentially include recently studied appearance and label ordering constraints [28–31], which could be readily applied to the proposed framework. Lastly, we provide the implementation, as well as the Matlab prototype used in this study, to the community. The proposed method is included into the Advanced Segmentation Tools (ASeTs) repository (<http://sourceforge.net/projects/asets/>).

## References

1. Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A continuous max-flow approach to potts model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 379–392. Springer, Heidelberg (2010)
2. Paragios, N., Chen, Y., Faugeras, O.: Handbook of Mathematical Models in Computer Vision. Springer-Verlag New York, Inc., Secaucus (2005)
3. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Applied Mathematical Sciences, vol. 153. Springer, New York (2003)
4. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321–331 (1988)
5. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision* 72, 195–215 (2007), doi:10.1007/s11263-006-8711-1
6. Mitiche, A., Ayed, I.B.: Variational and Level Set Methods in Image Segmentation (Springer Topics in Signal Processing), 2011th edn. Springer (2010)
7. Zhao, H.K., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *Journal of Computational Physics* 127(1), 179–195 (1996)
8. Paragios, N., Deriche, R.: Coupled geodesic active regions for image segmentation: A level set approach. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 224–240. Springer, Heidelberg (2000)
9. Brox, T., Weickert, J.: Level set segmentation with multiple regions. *IEEE Transactions on Image Processing* 15(10), 3213–3218 (2006)
10. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision* 50, 271–293 (2002)
11. Almgren, F., Taylor, J.E., Wang, L.: Curvature-driven flows: a variational approach. *SIAM J. Control Optim.* 31(2), 387–438 (1993)
12. Luckhaus, S., Sturzenhecker, T.: Implicit time discretization for the mean curvature flow equation. *Calc. Var. Partial Differential Equations* 3(2), 253–271 (1995)
13. Boykov, Y., Kolmogorov, V., Cremers, D., Delong, A.: An integral solution to surface evolution PDEs via geo-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 409–422. Springer, Heidelberg (2006)
14. Chambolle, A.: An algorithm for mean curvature motion. *Interf. Free Bound.* 6, 195–218 (2004)
15. Yuan, J., Ukwatta, E., Tai, X.C., Fenster, A., Schnoerr, C.: A fast global optimization-based approach to evolving contours with generic shape prior. Technical report CAM-12-38, UCLA (2012)
16. Yuan, J., Ukwatta, E., Qiu, W., Rajchl, M., Sun, Y., Tai, X.-C., Fenster, A.: Jointly segmenting prostate zones in 3D MRIs by globally optimized coupled level-sets. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) EMMCVPR 2013. LNCS, vol. 8081, pp. 12–25. Springer, Heidelberg (2013)

17. Chan, T.F., Esedoğlu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* 66(5), 1632–1648 (2006) (electronic)
18. Yuan, J., Bae, E., Tai, X.: A study on continuous max-flow and min-cut approaches. In: *CVPR 2010* (2010)
19. Pock, T., Chambolle, A., Bischof, H., Cremers, D.: A convex relaxation approach for computing minimal partitions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida (2009)
20. Lellmann, J., Breitenreiter, D., Schnörr, C.: Fast and exact primal-dual iterations for variational problems in computer vision. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part II*. LNCS, vol. 6312, pp. 494–505. Springer, Heidelberg (2010)
21. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001)
22. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 359–374 (2001)
23. Bresson, X., Chan, T.F.: Active contours based on Chambolle’s mean curvature motion. In: *IEEE International Conference on Image Processing, ICIP 2007*, September 16–October 19, vol. 1, pp. 33–36 (2007)
24. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained total variation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) *SSVM 2009*. LNCS, vol. 5567, pp. 150–162. Springer, Heidelberg (2009)
25. Lellmann, J., Lenzen, F., Schnörr, C.: Optimality bounds for a variational relaxation of the image partitioning problem. In: Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F.R. (eds.) *EMMCVPR 2011*. LNCS, vol. 6819, pp. 132–146. Springer, Heidelberg (2011)
26. Potts, R.B.: Some generalized order-disorder transformations. In: *Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 106–109 (1952)
27. Ayed, I.B., Mitiche, A., Belhadj, Z.: Multiregion level-set partitioning of synthetic aperture radar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(5), 793–800 (2005)
28. Rajchl, M., Yuan, J., Ukwatta, E., Peters, T.: Fast interactive multi-region cardiac segmentation with linearly ordered labels. In: *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1409–1412. *IEEE Conference Publications* (2012)
29. Rajchl, M., Yuan, J., White, J., Ukwatta, E., Stirrat, J., Nambakhsh, C., Li, F., Peters, T.: Interactive hierarchical max-flow segmentation of scar tissue from late-enhancement cardiac mr images. *IEEE Transactions on Medical Imaging* 33(1), 159–172 (2014)
30. Baxter, J.S., Rajchl, M., Yuan, J., Peters, T.M.: A continuous max-flow approach to general hierarchical multi-labelling problems. *arXiv preprint arXiv:1404.0336* (2014)
31. Baxter, J.S., Rajchl, M., Yuan, J., Peters, T.M.: A continuous max-flow approach to multi-labeling problems under arbitrary region regularization. *arXiv preprint arXiv:1405.0892* (2014)

# An Efficient Curve Evolution Algorithm for Multiphase Image Segmentation

Günay Doğan

Theiss Research,  
National Institute of Standards and Technology,  
100 Bureau Dr., Stop 8910,  
Gaithersburg, MD 20899-8910, USA  
`gunay.dogan@nist.gov`

**Abstract.** We propose a novel iterative algorithm for multiphase image segmentation by curve evolution. Specifically, we address a multiphase version of the Chan-Vese piecewise constant segmentation energy. Our algorithm is efficient: it is based on an explicit Lagrangian representation of the curves and it converges in a relatively small number of iterations. We devise a stable curvature-free semi-implicit velocity computation scheme. This enables us to take large steps to achieve sharp decreases in the multiphase segmentation energy when possible. The velocity and curve computations are linear with respect to the number of nodes on the curves, thanks to a finite element discretization of the curve and the gradient descent equations, yielding essentially tridiagonal linear systems. The step size at each iteration is selected using a non-monotone line search algorithm ensuring rapid progress and convergence. Thus, the user does not need to specify fixed step sizes or iteration numbers. We also introduce a novel dynamic stopping criterion, robust to various imaging conditions, to decide when to stop the iterations. Our implementation can handle topological changes of curves, such as merging and splitting as well. This is a distinct advantage of our approach, because we do not need to know the number of phases in advance. The curves can merge and split during the evolution to detect the correct regions, especially the number of phases.

## 1 Introduction

The goal of this work is to devise an efficient algorithm for segmentation of approximately piecewise constant images. We are given a possibly noisy and degraded image  $I$  with domain composed of distinct regions  $\{\Omega_l\}_{l=0}^{n_\Omega}$  each with homogeneous image intensity, i.e.,  $I|_{\Omega_l} \approx c_l$  (see Figure 1), and we would like to extract the boundaries of all the regions in the image and the average values  $\{c_l\}$  of image intensity for all regions. We express this problem as an energy minimization problem, in which a curve  $\Gamma = \bigcup_{l=1}^{n_\Omega} \Gamma_l$ , which is the union of a set of simple closed curves, and a set of region averages  $\{c_l\}_{l=0}^{n_\Omega}$ , also the number of regions  $n_\Omega$  are the unknowns. This can be considered as a more general version of two-phase segmentation problem solved by Chan and Vese in [5]. Given an

image  $I : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  defined on a bounded image domain  $D$ , we seek to minimize the following energy

$$J(\Gamma, \{c_l\}) = \frac{1}{2} \sum_{l=0}^{n_\Omega} \int_{\Omega_l} \chi_D(x) (I(x) - c_l)^2 dx + \mu \int_\Gamma d\Gamma, \quad \mu > 0 \quad (1)$$

where  $\chi_D$  is the indicator function for the image domain  $D$ ; it is included to account for the situations when the curves (and the enclosed regions) in the geometric model extend beyond  $D$ , but the image data is available only on  $D$ . The first term in the energy (1) is a data fidelity term so that the optimal curves match the boundaries of the homogeneous regions in the given image. The second term is a length penalty and favors shorter and smoother curves, so that the optimal curves do not fit insignificant variations or noise in the image.

Many different approaches have been proposed to address the problem of multiphase image segmentation. Recent notable approaches are based on graph formulations [1,3,16,23], convex relaxations [2,4,17,21], variational formulations [15,22] and level sets [5,7,11,27,26]. Our algorithm is closer to the level set approach, but it is Lagrangian and offers several advantages that we explain below.

To develop our segmentation algorithm, we study the multiphase energy (1), derive its shape derivative (or first variation) and propose a semi-implicit gradient descent algorithm (in Section 2) that enables large steps (hence fewer iterations) in the minimization. Then we propose a numerical realization (in Section 3) using explicit (but nonparametric) Lagrangian curves to represent the region boundaries, and the finite element method to compute the gradient descent velocity in linear time with respect to the number of nodes on the curves, thereby to perform the curve updates very efficiently at each iteration of the minimization. We also introduce numerical procedures to ensure robustness and reliability in execution, and address issues of practical importance, such as automation of step size and stopping criterion, ensuring adequate distribution of nodes in curve representation, topological changes, i.e., merging and splitting of curves during the curve evolution.

We emphasize the following main contributions of our work:

- Formulation and implementation of the multiphase segmentation problem with only *a set of explicit polygonal curves*, in a way that does not require several level set functions, label grids or indicator functions to represent multiple regions. Moreover we do not need to know the number of regions or phases a priori, this number can change during the minimization.
- A *curvature-free semi-implicit gradient descent scheme* that is unconditionally stable with respect to step size, so that we can take *large steps* at each iteration and converge to the minimum of the energy (1) faster. Curvature is a second order differential geometric quantity and is difficult to handle in both parametric and level set approaches; therefore, not having to compute the curvature to obtain the gradient descent velocity is a critical ingredient in the efficiency and stability of our algorithm.
- A *linear time algorithm for velocity computation* through finite element discretization. This way we obtain sparse matrices that we need to invert at

each iteration to compute the gradient descent velocity. The sparse matrices consist of circulant tridiagonal blocks on the diagonal and are inverted in  $O(n)$  time where  $n$  is the number of curve nodes.

The latter two features of our approach are central to its efficiency. The memory footprint of the curves and the associated data structures, such as the vectors and matrices, is proportional to the number of nodes, and the time complexity consists of pixel summations and other curve procedures, with cost linearly proportional to the number of nodes.

Additionally, our algorithm requires minimal input from the user for execution. It is fully automatic; the user only sets the smoothing parameter  $\mu$  in (1) and specifies the initial curve(s) and our algorithm performs the minimization without requiring the step size, the number of iterations or other stopping parameters from the user. We use line search to choose the right step size and follow the norm of the shape gradient with a novel dynamic tolerance formula to determine convergence (see Section 2).

Our algorithm currently has two limitations:

- It does not handle junctions. We assume that the boundaries of the homogeneous regions are simple curves that do not intersect with each other. This is not true for some images. Handling the cases with junctions requires some changes to our model and we will address this in future work.
- It does not guarantee convergence to a global minimum, and it can converge to a local minimum. But this can be alleviated with a good initialization scheme, such as topology optimization [12].

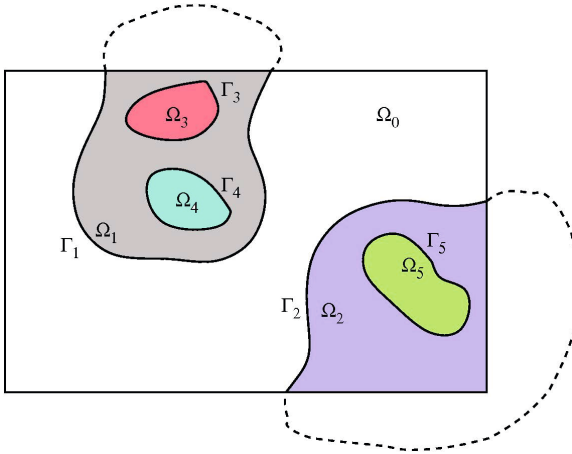
## 2 Gradient Descent Algorithm

The geometry of the problem is specified by a set of disjoint domains  $\{\Omega_l\}_{l=0}^{n_\Omega}$  that are used to cover the image domain  $D$ , namely, we have  $D \subset \bigcup_{l=0}^{n_\Omega} \Omega_l$  (note that a domain  $\Omega_l$  may extend beyond  $D$ , as illustrated in Figure 1). The boundaries  $\{\partial\Omega_l\}_{l=0}^{n_\Omega}$  of the domains make up the curve  $\Gamma$ , which is the free variable in this problem. The curve  $\Gamma$  is the union of a set of simple (non-intersecting) closed curves  $\{\Gamma_l\}_{l=1}^{n_\Omega}$ . The numbering or indexing of the domains and curves is such that a simple curve  $\Gamma_l$  gives the outer boundary of domain  $\Omega_l$ . We distinguish between two cases:

- If  $\Omega_l$  has no interior boundary due to a hole (i.e. another domain inside  $\Omega_l$ ), then  $\partial\Omega_l = \Gamma_l$ , namely, the boundary  $\partial\Omega_l$  of  $\Omega_l$  is equal to  $\Gamma_l$ .
- If  $\Omega_l$  encloses some other domains  $\{\Omega_k\}$  inside, then the boundary  $\partial\Omega_l$  of  $\Omega_l$  includes the outer boundaries  $\{\Gamma_k\}$  of  $\{\Omega_k\}$ , namely  $\partial\Omega_l = \Gamma_l \cup \bigcup_k \Gamma_k$ .

See Figure 1 for an illustration of the domains, curves and their numbering. Note that the previous work on variational multiphase segmentation required multiple level set functions, labeled grids or indicator functions in 2d to represent multiphase partitioning of the image domain. This is in contrast with our approach, which is built on a set of 1d curves. Moreover, although the number of distinct





**Fig. 1.** Illustration of domains, boundaries and their numbering. The image domain  $D$  is covered by the regions  $\{\Omega_l\}_{l=0}^{n_\Omega}$ , specified by the curves  $\Gamma = \bigcup_{l=0}^{n_\Omega} \Gamma_l$ .

regions  $n_\Omega$  is a constant in our formulation (1), in the implementation it need not be known a priori, and it changes during the minimization process as the curves merge and split.

Before we continue to develop the gradient descent algorithm for the energy (1), we review some definitions and concepts from differential geometry. We denote the outer unit normal, the scalar curvature and the curvature vector of a curve  $\Gamma \in C^2$  by  $\mathbf{n}$ ,  $\kappa$ ,  $\boldsymbol{\kappa} (:= \kappa \mathbf{n})$  respectively. For a given function  $f \in C^2(D)$ , we define tangential gradient  $\nabla_\Gamma f$  and tangential Laplacian  $\Delta_\Gamma f$ :

$$\nabla_\Gamma f = \left( \nabla f - \frac{\partial f}{\partial \mathbf{n}} \mathbf{n} \right) \Big|_\Gamma, \quad \Delta_\Gamma f = \left( \Delta f - \mathbf{n} \cdot D^2 f \cdot \mathbf{n} - \kappa \frac{\partial f}{\partial \mathbf{n}} \right) \Big|_\Gamma.$$

If the function  $f$  is defined on  $\Gamma$  only, then we consider a normal extension of  $f$  and use the same definitions for the tangential derivatives.

**Shape Derivative of the Energy.** We use the concept of shape derivatives to understand the change in the energy induced by a given velocity field  $\mathbf{V}$ . Once we have the means to evaluate how any given velocity affects the energy, we can choose from the space of admissible velocities the particular velocity that decreases the energy (1) for a given  $\Gamma$ . We define the shape derivative of an energy  $J(\Gamma)$  at  $\Gamma$  with respect to a velocity field  $\mathbf{V}$  as the limit

$$dJ(\Gamma; \mathbf{V}) = \lim_{t \rightarrow 0} \frac{1}{t} (J(\Gamma_t) - J(\Gamma)),$$

where  $\Gamma_t = \{x(t, X) : X \in \Gamma\}$  is the deformation of  $\Gamma$  by  $\mathbf{V}$  via the ordinary differential equation  $\frac{dx}{dt} = \mathbf{V}(x(t))$ ,  $x(0) = X$ . Shape derivative of energies  $J(\Omega)$  depending on domains or regions  $\Omega$  are defined similarly. We refer to the book [8] for more information on shape derivatives.

**Lemma 1 ([8]).** *The shape derivative of curve length  $J(\Gamma) = |\Gamma| = \int_{\Gamma} d\Gamma$  with respect to velocity field  $\mathbf{V}$  is  $dJ(\Gamma; \mathbf{V}) = \int_{\Gamma} \kappa V d\Gamma$ , where  $V = \mathbf{V} \cdot \mathbf{n}$  is the normal component of the vector velocity.*

**Lemma 2 ([5,8,12]).** *The shape derivative of the data fidelity term*

$$J(\Omega_l) = \frac{1}{2} \int_{\Omega_l} \chi_D(x)(I(x) - c_l)^2 dx$$

from energy (1) for domain  $\Omega_l$  with respect to velocity field  $\mathbf{V}$  is

$$dJ(\Omega_l; \mathbf{V}) = \frac{1}{2} \int_{\partial\Omega_l} \chi_D(x)(I(x) - c_l)^2 V dx, \tag{2}$$

where  $V = \mathbf{V} \cdot \mathbf{n}$  is the normal component of the vector velocity.

We will use Lemma 1 and Lemma 2 next to write the shape derivative for our energy (1) that is based on multiple curves and multiple regions.

**Theorem 1.** *The shape derivative of the energy (1) for  $\Gamma = \cup_{l=1}^{n_{\Omega}} \Gamma_l \in C^2$  with respect to a given velocity field  $\mathbf{V}$  is*

$$dJ(\Gamma; \mathbf{V}) = \int_{\Gamma} G V d\Gamma, \quad G = \mu \kappa + f(\Gamma), \tag{3}$$

where  $G$  is the shape gradient,  $V = \mathbf{V} \cdot \mathbf{n}$  is the normal component of the velocity and the image-based force term  $f$  is defined by

$$f|_{\Gamma_l} = (c_l^{out} - c_l^{in}) \chi_D(x) \left( I(x) - \frac{1}{2}(c_l^{in} + c_l^{out}) \right), \tag{4}$$

in which  $c_l^{in} = c_l = \frac{1}{|\Omega_l \cap D|} \int_{\Omega_l \cap D} I(x) dx$  is the average image intensity in the region  $\Omega_l \cap D$  enclosed by  $\Gamma_l$ , and  $c_l^{out} = c_k = \frac{1}{|\Omega_k \cap D|} \int_{\Omega_k \cap D} I(x) dx$  is the average over the outer region  $\Omega_k$  enclosing both  $\Omega_l$  and  $\Gamma_l$ . We can write the shape derivative more explicitly as

$$dJ(\Gamma; \mathbf{V}) = \mu \int_{\Gamma} \kappa V d\Gamma + \sum_{l=1}^{n_{\Omega}} (c_l^{out} - c_l^{in}) \int_{\Gamma_l} \chi_D(x) \left( I(x) - \frac{1}{2}(c_l^{in} + c_l^{out}) \right) V d\Gamma$$

*Proof.* The Euler-Lagrange equation of the energy (1) with respect to the unknown  $c_l$  gives  $c_l = \frac{1}{|\Omega_l \cap D|} \int_{\Omega_l \cap D} I(x) dx$ . We use Lemmas 1, 2 and write the shape derivative of the energy (1) as

$$dJ(\Gamma; \mathbf{V}) = \mu \int_{\Gamma} \kappa V d\Gamma + \frac{1}{2} \sum_{l=0}^{n_{\Omega}} \int_{\partial\Omega_l} \chi_D(x)(I(x) - c_l)^2 V d\Gamma. \tag{5}$$

Note that the boundary  $\Omega_l$  consists of the curves  $\Gamma_l$  and  $\{\Gamma_k : k \in IN(l)\}$ , where  $IN(l)$  is the set of indices of the curves immediately inside  $\Gamma_l$ . The domain  $\Omega_0$

does not have an outer boundary, it has only interior boundaries given by the curves  $\{\Gamma_k : k \in IN(0)\}$ . Thus we rewrite the shape derivative as follows

$$\begin{aligned}
 dJ(\Gamma; \mathbf{V}) &= \mu \int_{\Gamma} \kappa V d\Gamma + \frac{1}{2} \sum_{k \in IN(0)} \int_{\Gamma_k} \chi_D(x) (I(x) - c_k)^2 \mathbf{V} \cdot (-\mathbf{n}_k) d\Gamma \\
 &\quad + \frac{1}{2} \sum_{l=1}^{n_{\Omega}} \left( \int_{\Gamma_l} \chi_D(x) (I(x) - c_l)^2 \mathbf{V} \cdot \mathbf{n}_l d\Gamma \right. \\
 &\quad \left. + \sum_{k \in IN(l)} \int_{\Gamma_k} \chi_D(x) (I(x) - c_k)^2 \mathbf{V} \cdot (-\mathbf{n}_k) d\Gamma \right). \tag{6}
 \end{aligned}$$

Each simple curve  $\Gamma_l$  appears only twice in the expression (6), because  $\Gamma_l$  is the outer boundary of the domain  $\Omega_l$  and it is an inner boundary of the domain  $\Omega_m$ , namely  $l \in IN(m)$ . So we can collect all the integrals of  $\Gamma_l$  together and reorganize (6) as

$$\begin{aligned}
 dJ(\Gamma; \mathbf{V}) &= \mu \int_{\Gamma} \kappa V d\Gamma + \frac{1}{2} \sum_{l=1}^{n_{\Omega}} \int_{\Gamma_l} \chi_D(x) \left( (I(x) - c_l^{out})^2 - (I(x) - c_l^{in})^2 \right) \mathbf{V} \cdot \mathbf{n}_l d\Gamma \\
 &= \mu \int_{\Gamma} \kappa V d\Gamma + \sum_{l=1}^{n_{\Omega}} (c_l^{out} - c_l^{in}) \int_{\Gamma_l} \chi_D(x) \left( I(x) - \frac{1}{2}(c_l^{in} + c_l^{out}) \right) \mathbf{V} \cdot \mathbf{n}_l d\Gamma,
 \end{aligned}$$

where  $c_l^{in} = c_l$  is the constant value for the domain  $\Omega_l$ , for which  $\partial\Omega_l$  is the outer boundary, and  $c_l^{out}$  is the constant value for the enclosing domain  $\Omega_m$ , for which  $\Gamma_l$  is an inner boundary. This concludes our proof.

**Gradient Descent for Minimization.** Theorem 1 enables us to derive a gradient descent velocity, so that we can evolve a given initial curve  $\Gamma^0$  continuously in a manner that decreases its energy  $J(\Gamma)$  and drives it to a minimum of the energy (1). For this, we simply set  $\mathbf{V} = -(\mu\boldsymbol{\kappa} + f(\Gamma)\mathbf{n})$  following the shape derivative equations (3), (4). By substituting this velocity in  $V = \mathbf{V} \cdot \mathbf{n}$  and then in (3), we can verify that

$$dJ(\Gamma; \mathbf{V}) = \int_{\Gamma} (\mu\kappa + f)V d\Gamma = - \int_{\Gamma} (\mu\kappa + f)^2 d\Gamma \leq 0,$$

by recalling  $\boldsymbol{\kappa} = \kappa\mathbf{n}$ . Thus,  $\mathbf{V}$  is indeed a gradient descent velocity, and to pursue the minimization, one can conceive a curve evolution scheme as follows: Start with initial curve  $\Gamma^0$  and update the curve iteratively by

$$\mathbf{V}^k = -\mu\boldsymbol{\kappa}^k - f(\Gamma^k)\mathbf{n}^k, \quad \mathbf{X}^{k+1} = \mathbf{X}^k + \tau^k \mathbf{V}^k, \quad \forall \mathbf{X}^k \in \Gamma^k.$$

This update scheme can be viewed as an explicit time discretization of the evolution equation  $\frac{dx}{dt} = \mathbf{V}(x) = -\mu\boldsymbol{\kappa} - f(\Gamma)\mathbf{n}$ ; we use the curve  $\Gamma^k$  at the current step to compute the geometric quantities  $\mathbf{n}, \boldsymbol{\kappa}$  and the data term  $f(\Gamma)$ ; we then compute the velocity  $\mathbf{V}$ , and finally update  $\Gamma^k$  with  $\mathbf{V}$  to obtain  $\Gamma^{k+1}$ . The fact

that the scheme is explicit and contains the curvature term creates stability issues resulting in spurious oscillations on the curve as we iterate. This is a feature of this geometric update scheme and it is there regardless of the representation, whether it is parametric curves or level sets. This instability can be prevented by taking small steps  $\tau$ , but this leads to too many iterations and increased computation time. To circumvent these difficulties, we propose a semi-implicit update scheme:

$$\mathbf{V}^{k+1} + \mu \boldsymbol{\kappa}^{k+1} = -f(\Gamma^k) \mathbf{n}^k, \quad \mathbf{X}^{k+1} = \mathbf{X}^k + \tau^k \mathbf{V}^{k+1}, \quad \forall \mathbf{X}^k \in \Gamma^k,$$

where we choose to evaluate the curvature term  $\boldsymbol{\kappa}$  in the next iteration rather than the current iteration. We recall  $\boldsymbol{\kappa} = -\Delta_\Gamma \mathbf{X}$ , so we can write  $\boldsymbol{\kappa}^{k+1} = -\Delta_\Gamma \mathbf{X}^{k+1} = -\tau^k \Delta_\Gamma \mathbf{V}^{k+1} + \Delta_\Gamma \mathbf{X}^k$ . Hence we obtain the following form of the semi-implicit update, in which the curvature does not appear explicitly

$$\begin{aligned} (Id - \mu \tau^k \Delta_\Gamma) \mathbf{V}^{k+1} &= -\mu \Delta_\Gamma \mathbf{X}^k - f(\Gamma^k) \mathbf{n}^k, \\ \mathbf{X}^{k+1} &= \mathbf{X}^k + \tau^k \mathbf{V}^{k+1}, \quad \forall \mathbf{X}^k \in \Gamma^k. \end{aligned} \tag{7}$$

Now computing the velocity  $\mathbf{V}^{k+1}$  at each iteration requires inverting a second order tangential differential operator, namely computing

$$-(Id - \mu \tau^k \Delta_\Gamma)^{-1} (\mu \Delta_\Gamma \mathbf{X}^k - f(\Gamma^k) \mathbf{n}^k). \tag{8}$$

We will see that this operation can be done in linear time with respect to the number points on the curve; therefore, the semi-implicit step is asymptotically as efficient as the explicit step. Moreover, it is unconditionally stable with respect to step size  $\tau^k$ , so we can take steps as large as we need in order to achieve significant energy decrease and to approach the minimum in few iterations.

Equation (8) can also be viewed as preconditioning the gradient descent velocity with a smoothing operator, and this is helpful in understanding the stability of this scheme. In other words, we are solving for a gradient descent velocity in  $H^1(\Gamma)$  space and this has been shown to have favorable properties in recent works [6,24,25]. Moreover, an  $H^1$  gradient descent scheme seems to appear naturally when we use the second shape derivative to derive gradient descent velocities for curve integrals [13,14]. Using the semi-implicit updating scheme (7), we devise the Algorithm 1 for iterative minimization.

At this point, we should point out two important pieces of Algorithm 1: the stopping criterion and step size selection. In many implementations of the Chan-Vese segmentation algorithm [5] with curve evolution and its variants, users fix the step size  $\tau$  and take a fixed number of iterations. This is not a good approach, because sometimes the number of iterations may not be enough to reach the minimum, or sometimes one keeps taking many unnecessary iterations even when the curve is already at the minimum. The solution implemented in [10,14] is to select the step that satisfies the Armijo energy decrease criterion [19] at each iteration and to stop when the norm of the shape gradient falls below a stopping tolerance. At each iteration, we choose a step size  $\tau^k$  that satisfies the

---

**Algorithm 1.** Gradient Descent Algorithm

---

```

set initial curve  $\Gamma^0$  and  $k = 0$ 
repeat
  mark domains  $\{\Omega_l\}$  on pixels
  sum up pixels in  $\Omega_l$  as  $\text{SUM}_l$  and compute average  $c_l = \frac{1}{|\Omega_l \cap D|} \text{SUM}_l$ 
  compute energy  $J^k = J(\Gamma^k)$ 
  solve  $(Id - \mu\tau^k \Delta_\Gamma)\mathbf{V}^{k+1} = -\mu\Delta_\Gamma \mathbf{X}^k - f(\Gamma^k)\mathbf{n}^k$ 
  test step size  $\tau^k$ , modify if necessary to ensure energy decrease
  update  $\mathbf{X}^{k+1} = \mathbf{X}^k + \tau^k \mathbf{V}^{k+1}$ ,  $\forall \mathbf{X}^k \in \Gamma^k$ 
until  $\|G^{k+1}\|_{L^2} < \text{tol}(\Gamma^k, \{c_l^k\})$ 

```

---

nonmonotone energy decrease condition proposed by Zhang and Hager in [28] for continuous optimization

$$J(\Gamma^{k+1}) < C^k + \alpha\tau^k dJ(\Gamma^k; \mathbf{V}^k), \tag{9}$$

where  $C^k = (\eta Q^{k-1} C^{k-1} + J(\Gamma^k))/Q^k$ ,  $C^0 = J(\Gamma^0)$ ,  $Q^k = \eta Q^{k-1} + 1$ ,  $Q^0 = 0$ . We set  $\alpha = 10^{-4}$ ,  $\eta = 0.2$  in our experiments. We found that the nonmonotone energy decrease condition (9) is more robust and more efficient than the monotone counterpart. It allows energy increases in some iterations, but ensures good progress towards the minimum.

We found that a fixed tolerance on the norm of the shape gradient was not a robust stopping criterion across various image examples, e.g., poor contrast, unbalanced region sizes, etc. Thus we derived our novel dynamic cutoff tolerance. We impose the following relative tolerance on pointwise values of the data fidelity term (setting  $\mu = 0$  in the shape gradient  $G$  (3)),

$$\left| I(x) - \frac{1}{2}(c_{in} + c_{out}) \right| < \varepsilon \frac{1}{2} |c_{in} - c_{out}|. \tag{10}$$

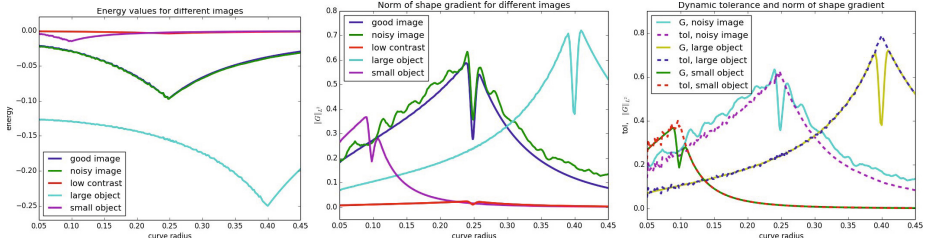
The term  $\frac{1}{2}|c_{in} - c_{out}|$  gives a measure of contrast between neighboring regions. Noting  $G|_{\Gamma_l} = (c_{in}^l - c_{out}^l)(I(x) - \frac{1}{2}(c_{in}^l + c_{out}^l))$ , we multiply (10) by  $(c_{in}^l - c_{out}^l)$  and integrate over  $\Gamma$  to obtain the  $L^2$  norm of the shape gradient, thereby getting the following condition as a stopping criterion:

$$\|G\|_{L^2} = \left( \int_{\Gamma} |G|^2 d\Gamma \right)^{1/2} < \text{tol}(\Gamma^k, \{c_l^k\}) = \frac{1}{2} \min_l (|c_{in}^l - c_{out}^l|)^2 |\Gamma|^{1/2} \varepsilon. \tag{11}$$

Figure 2 illustrates the effectiveness of the dynamic cutoff tolerance in identifying the dip in  $\|G\|_{L^2}$  in different versions of the same segmentation problem under varying conditions. A fixed tolerance would give either premature termination or no termination for these images.

### 3 Discretization and Numerical Solution

The works building on the Chan-Vese approach [5] to image segmentation represent the curves as level set functions. The main advantage of the level set



**Fig. 2.** Illustration of dynamic cutoff tolerance for different versions of a simple image example, a filled white circle centered at  $(0.5,0.5)$  in the foreground on a square image domain  $[0, 1]^2$  with zero background. Good, noisy and low contrast images have a circle of radius 0.25, large object has radius 0.40, small object has radius 0.10. Low contrast has gray-scale value 0.1 in the circle, whereas the others have gray-scale value 1.0. The noisy image has uniform noise added to all pixels with values between  $[-0.5,0.5]$ . The plots show the energy values, the  $L^2$  norm of the shape gradient, the dynamic cutoff tolerance for circle curves centered at  $(0.5,0.5)$  on these images. The dip in the shape gradient norms signal the optimal circle radius in the minimization and cannot be captured with a fixed tolerance or criterion. On the other, the dynamic cutoff tolerance (shown with dotted curves on the right) tracks the norm of shape gradient very well. We can stop our iterative algorithm when the norm of the shape gradient falls significantly below the dynamic tolerance.

approach is that it can handle topological changes, such as splitting and merging of curves, easily without additional work. The main disadvantage of the level set approach is the high computational cost of introducing an additional dimension to represent a 1d object, often using the image grid itself as the basis of a 2d array representation for the curve. The level set representation can be difficult to maintain through the evolution and may require costly reinitialization or other regularization schemes [20]. Moreover, representing more than two phases with level sets requires using more than one level set function, thus increasing the computational cost further.

We choose to work with explicit Lagrangian representations of curves, because it is much more efficient with respect to memory use and running time than the level set approach; *all our velocity computations and curve updates have linear time complexity with respect to the number of points on the curve*, and the number of points used to represent the curves is much fewer than the number of pixels that the image contains; therefore, the number of variables that we need to deal with is much lower than it would be in the case of a level set approach (including narrow-band level set representation).

**Curve Representation and Adaptivity.** We approximate a continuous curve  $\Gamma$  as a polygonal curve  $\Gamma^h$  that consists of linear curve elements  $\{\Gamma_i^h\}_{i=1}^m$ . Thus the curve approximation is piecewise linear. The curve element  $\Gamma_i^h$  is a segment connecting a point  $(x_{1,i}, x_{2,i})$  to the next point  $(x_{1,i+1}, x_{2,i+1})$ . This way the discrete curve  $\Gamma^h$  is stored as an ordered list of curve nodes  $\{(x_{1,i}, x_{2,i})\}_{i=1}^m$  and *this representation, nor the velocity computations described below, does not*

require a parameterization. Therefore our approach is explicit Lagrangian, but not parametric.

An important issue in realizing the Lagrangian curve representation is where to put the nodes and how to distribute them, especially after the curve is deformed by the iterations in unpredictable ways. A reasonable strategy is to equidistribute the nodes yielding uniform element length, but this approach is suboptimal. Ideally we would like to distribute the nodes in an economical way, just enough to capture the geometry and the image faithfully, but not use more nodes than necessary in order to control the computational cost; namely we put more nodes where the geometry and image vary more and few nodes in flat regions. Thus, the curve representation is spatially adaptive and it changes dynamically during the gradient descent evolution following our adaptivity criteria. We realize this through two atomic operations on the curve:

- *Coarsening*: Combines two consecutive elements  $\Gamma_i^h, \Gamma_{i+1}^h$  into one by removing the shared node.
- *Refinement*: Splits an element  $\Gamma_i^h$  into two elements by adding a new node  $(x_{1,i+\frac{1}{2}}, x_{2,i+\frac{1}{2}})$  in the middle. The new node is displaced in the normal direction to match the average of the curvatures at the two nodes of  $\Gamma_i^h$ . (Curvature at a node is estimated by fitting an osculating circle to the node and its two neighbors.)

The decision to refine or coarsen an element is based on the following two criteria:

- *Geometric criterion*: The error on an element  $\Gamma_i^h$  for piecewise linear approximation is bounded by  $\max_{\Gamma_i^h} \kappa |\Gamma_i^h|$ . If this error estimate is high, we refine. If it is very small, then we coarsen.
- *Data criterion*: We aim to evaluate if the element resolves the underlying image data. For this, we compare two numerical approximations of the integral  $\int_{\Gamma_i^h} I(x) d\Gamma$  by a low order and a high order numerical quadrature rule. If the difference is large, we refine. If it is too small, we coarsen.

Since we have two criteria to guide adaptivity, we refine if one of the rules mark the element for refinement, and coarsen only if both rules mark the element and its neighbor for coarsening. We have observed that our approach to curve adaptivity works very well in a diverse set of scenarios. The curve dynamically adjusts the number of nodes during the minimization process capturing complicated geometries and images in an efficient and reliable manner. In addition to curve adaptivity, we have implemented topological changes for curves; we detect curve intersections and split or merge the curves if needed. Explaining the details of topological changes is beyond the scope of this paper and this will be pursued in a separate paper. Descriptions of other methods for topological changes of Lagrangian curves can be found in [9,18]. We perform the procedures for curve adaptivity and topological changes at each iteration after the curve has been moved by the new velocity.

**Finite Element Method for Velocity.** Computing the update velocity using our semi-implicit scheme at each iteration requires solving the following velocity PDE for  $\mathbf{V}^{k+1}$

$$-\mu\tau^k \Delta_\Gamma \mathbf{V}^{k+1} + \mathbf{V}^{k+1} = -\mu\Delta_\Gamma \mathbf{X}^k - f(\Gamma^k)\mathbf{n}^k \quad \text{on } \Gamma. \tag{12}$$

Note that since  $\mathbf{V}, \mathbf{X}, \mathbf{n}$  are vectors, Equation (12) is actually two PDEs to be solved on  $\Gamma^k$ , one for each component of these vector functions. To solve the PDE (12), we discretize it using the finite element method (FEM) on curves. For this, we first write its weak form: we multiply the PDE (12) with suitable test functions  $\phi$ , integrate over  $\Gamma^k$ , then integrate the tangential Laplacian term  $\Delta_\Gamma \mathbf{V}$  by parts

$$\langle \mathbf{V}^{k+1}, \phi \rangle + \mu\tau^{k+1} \langle \nabla_\Gamma \mathbf{V}^{k+1}, \nabla_\Gamma \phi \rangle = \mu \langle \nabla_\Gamma \mathbf{X}, \nabla_\Gamma \phi \rangle - \langle f(\Gamma^k)\mathbf{n}^k, \phi \rangle. \tag{13}$$

The brackets  $\langle \cdot, \cdot \rangle$  denote integrals on the curve  $\Gamma^k$ , i.e.  $\langle f, g \rangle = \int_{\Gamma^k} f(x)g(x)d\Gamma$  and  $\langle \mathbf{f}, \mathbf{g} \rangle = \begin{pmatrix} \langle f_1, g_1 \rangle \\ \langle f_2, g_2 \rangle \end{pmatrix}$ . Next we choose a finite set of nodal basis functions  $\{\phi_i\}$  to discretize the weak form of the velocity PDE (13). We use piecewise linear vector functions  $\phi_i = (\phi_i, \phi_i)$ , such that  $\phi_i$  is nonzero only on the elements  $\Gamma_{i-1}^h, \Gamma_i^h$ , but zero on the other elements, and satisfies

$$\phi_i(X_i) = 1, \quad \phi_i(X_{i-1}) = \phi_i(X_{i+1}) = 0.$$

We expand the unknown velocity in terms of the basis functions:  $\mathbf{V}(X) = \sum_i \mathbf{V}_i \phi_i(X)$  (omit the iteration index  $k$  to simplify notation), so that our new unknown becomes a finite vector of nodal velocity coefficients  $\mathbf{V} = \{\mathbf{V}_i\}_{i=1}^m$ . We expand the position vector  $\mathbf{X} = \sum_i \mathbf{X}_i \phi_i$  as well and obtain

$$\sum_j \mathbf{V}_j (\langle \phi_i, \phi_j \rangle + \mu\tau \langle \nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j \rangle) = -\langle \phi_i, f(\Gamma^h)\mathbf{n} \rangle + \mu \sum_j \mathbf{X}_j \langle \nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j \rangle.$$

We define the corresponding matrices  $\mathbf{M}_{ij} = \langle \phi_i, \phi_j \rangle$ ,  $\mathbf{A}_{ij} = \langle \nabla_\Gamma \phi_i, \nabla_\Gamma \phi_j \rangle$  and vector  $\mathbf{f}_i = \langle \phi_i, f(\Gamma^h)\mathbf{n} \rangle$ , and obtain the compact linear system that we need to solve to compute velocity at each iteration:

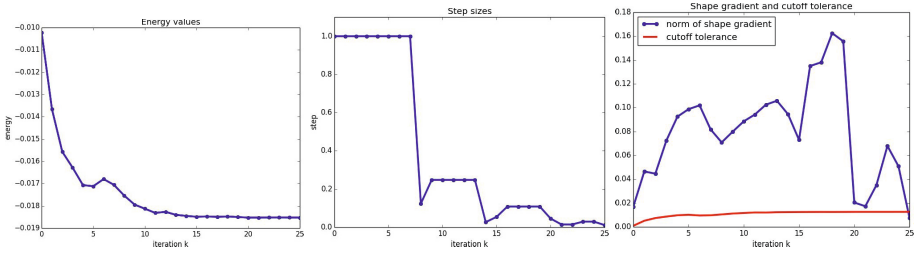
$$(\mathbf{M} + \mu\tau\mathbf{A})\mathbf{V} = \mu\mathbf{A}\mathbf{X} - \mathbf{f}.$$

Note that we actually have  $\mathbf{M} = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}$ ,  $\mathbf{A} = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$ ,  $\mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}$ . Since the basis functions are piecewise linear, the entries of the matrices  $M, A$  can be computed easily and are given by the following expressions:

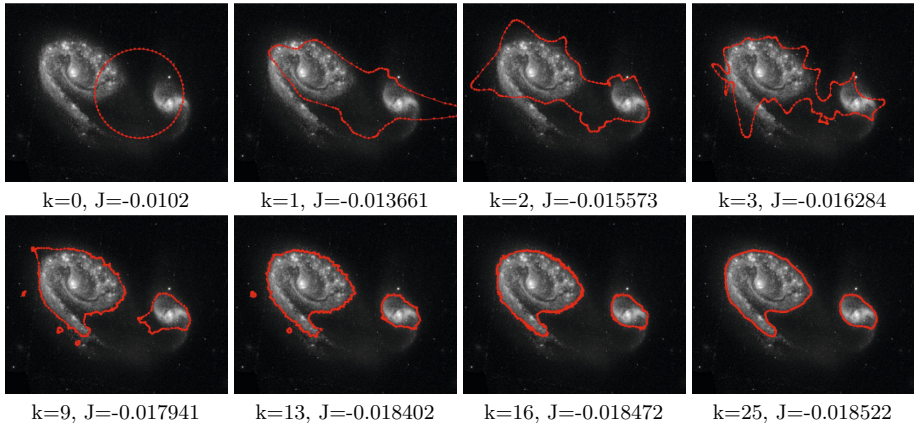
$$M_{ij}, A_{ij} = \begin{cases} d_i/3, & 2/d_i, & \text{if } i = j, \\ d_i/6, & -1/d_i, & \text{if } i = j - 1, j + 1 \text{ mod } m, \\ 0, & 0, & \text{otherwise,} \end{cases}$$

where  $d_i = |\Gamma_i^h|$  is the length of the  $i^{th}$  curve element and  $m$  is the number of nodes on the curve. The entries of the vector  $\mathbf{f}$ , on the other hand, depend on the image and can be computed with numerical quadrature by interpolating





**Fig. 3. The minimization dashboard.** The shape energy (left), the step sizes (middle),  $L^2$  norm of shape gradient (right, blue curve), the values of the dynamic cutoff tolerance (right, red curve) from the iterations of a segmentation example with galaxy image (Figure 4). The nonmonotone step size criterion allows some increases in energy to encourage large steps as much as possible. The dynamic cutoff tolerance tracking the  $L^2$  norm of the shape gradient signals when to stop the iterations.



**Fig. 4. Segmentation of galaxy image.** We observe large displacements in the first iterations to achieve sharp decreases in the energy and small displacements in the final iterations to ensure a good fit. The curves adapt to the new configurations easily by adding and subtracting nodes, always ensuring a good representation of the geometry.

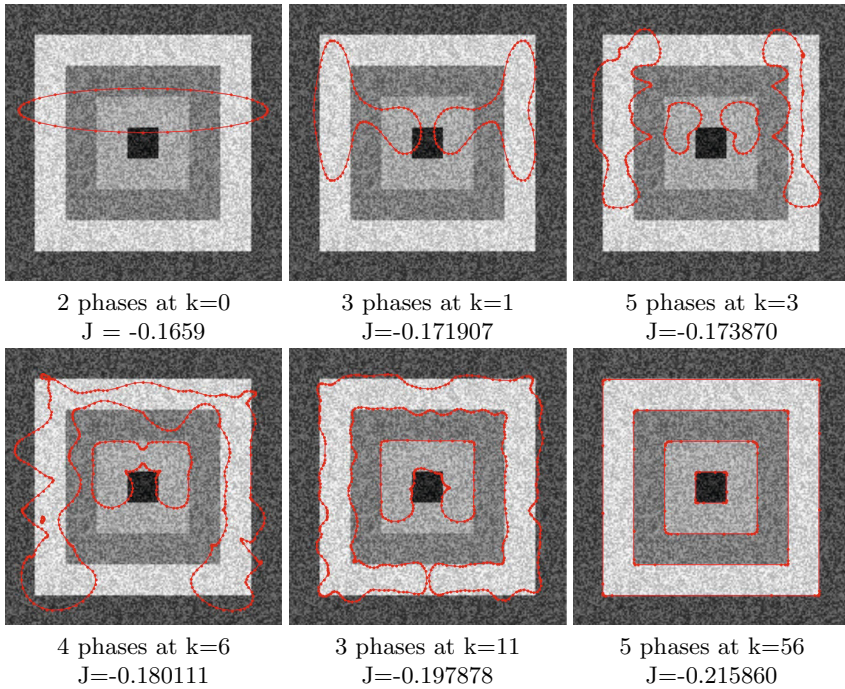
the image on each element  $\Gamma_i^h$ . The matrices  $M, A$  are circulant tridiagonal for a simple curve; therefore, they can be inverted in linear time with respect to the number of nodes on the curve using the Thomas algorithm. In the case of multiple simple curves, the matrices  $M, A$  consist of circulant tridiagonal blocks on their diagonals and zero elsewhere. The inversion is still linear time as each block is inverted independently.

## 4 Experiments

In this section, we present several numerical experiments demonstrating important aspects of our algorithm. We have confirmed the convergence and reliability

of our algorithm in many experiments. Figure 3 shows the computed energy values, step sizes, the norm of the shape gradient and the dynamic cutoff criterion from one of our experiments (segmentation of the galaxy image). The energy goes down steadily on average, but the nonmonotone step size criterion sometimes allows increases in the energy. We see the step sizes are kept large as long as possible, but they decrease as we get close the minimum to ensure a good fit in the final positioning of the curves at the minimum. The  $L^2$  norm of the shape gradient fluctuates through iterations. Our dynamic cutoff tolerance is small at the beginning and prevents premature termination. It increases gradually and helps detect the right dip in the norm of the shape gradient.

We also show two segmentation examples. One is a two-phase segmentation of the galaxy image shown in Figure 4. This example shows the judicious progress of the curve evolution. The curves take large steps in the first few iterations and achieve rapid energy decreases. The steps get smaller gradually to ensure convergence and a good fit at the minimum. The step sizes are determined by the



**Fig. 5. Segmentation of synthetic multiphase image.** This example shows segmentation of a noisy multiphase image. The curves take large steps and achieve large energy decreases at the beginning. They take small conservative steps close to the minimum to ensure a good fit and convergence. The number of phases changes during the evolution and finalizes in five phases. The number of nodes on the curves changes adaptively as well, increasing or decreasing as needed. The final curves have few nodes, because the region boundaries are straight.

nonmonotone step size selection criterion. The number of nodes on the curves changes at each iteration to ensure good representation of the geometry and resolving the image features. The curves have high quality representations at all times. They do not suffer from entanglements or other problems typically expected in front-tracking techniques. Moreover, topological changes, such as merging and splitting, are handled in a graceful manner by our topology surgery routines.

The second segmentation example is a noisy synthetic multiphase image, shown in Figure 5. The behaviour of the curve evolution is similar to that we observe for the galaxy image. In this image, we additionally observe adaptation of the phases. We start with two phases, increase to three phases right away in the first iteration, then increase to five, back to four, and finally settle at five phases when we terminate at the minimum. We see that the algorithm finds the correct number of phases without the user specifying this number a priori.

## References

1. Bae, E., Tai, X.-C.: Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) *SSVM 2009*. LNCS, vol. 5567, pp. 1–13. Springer, Heidelberg (2009)
2. Bae, E., Yuan, J., Tai, X.-C.: Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision* 92(1), 112–129 (2011)
3. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision* 70(2), 109–131 (2006)
4. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics* 66(5), 1632–1648 (2006)
5. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* 10(2), 266–277 (2001)
6. Charpiat, G., Maurel, P., Pons, J.-P., Keriven, R., Faugeras, O.: Generalized gradients: Priors on minimization flows. *IJCV* 73(3), 325–344 (2007)
7. Chung, G., Vese, L.A.: Image segmentation using a multilayer level-set approach. *Computing and Visualization in Science* 12(6), 267–285 (2009)
8. Delfour, M.C., Zolésio, J.-P.: *Shapes and Geometries*. Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2001)
9. Delingette, H., Montagnat, J.: Shape and topology constraints on parametric active contours. *Computer Vision and Image Understanding* 83(2), 140–171 (2001)
10. Doğan, G., Morin, P., Nochetto, R.H.: A variational shape optimization approach for image segmentation with a Mumford-Shah functional. *SIAM J. Sci. Comput.* 30(6), 3028–3049 (2008)
11. Dubrovina, A., Rosman, G., Kimmel, R.: Active contours for multi-region image segmentation with a single level set function. In: Pack, T. (ed.) *SSVM 2013*. LNCS, vol. 7893, pp. 416–427. Springer, Heidelberg (2013)
12. Hintermüller, M., Laurain, A.: Multiphase image segmentation and modulation recovery based on shape and topological sensitivity. *J. Math. Imaging Vision* 35(1), 1–22 (2009)

13. Hintermüller, M., Ring, W.: A second order shape optimization approach for image segmentation. *SIAM J. Appl. Math.* 64(2), 442–467 (2003)
14. Hintermüller, M., Ring, W.: An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *J. Math. Imaging Vision* 20(1-2), 19–42 (2004); Special issue on mathematics and image analysis
15. Jung, Y.M., Kang, S.H., Shen, J.: Multiphase image segmentation via modica-mortola phase transition. *SIAM Journal on Applied Mathematics* 67(5), 1213–1232 (2007)
16. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(2), 147–159 (2004)
17. Lellmann, J., Schnörr, C.: Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences* 4(4), 1049–1096 (2011)
18. McInerney, T., Terzopoulos, D.: T-Snakes: Topology Adaptive Snakes. *Medical Image Analysis* 4(2), 73–91 (2000)
19. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York (1999)
20. Osher, S., Fedkiw, R.: *Level set methods and dynamic implicit surfaces*. Applied Mathematical Sciences, vol. 153. Springer, New York (2003)
21. 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, CVPR 2009*, pp. 810–817. IEEE (2009)
22. Sandberg, B., Kang, S.H., Chan, T.F.: Unsupervised multiphase segmentation: A phase balancing model. *IEEE Transactions on Image Processing* 19(1), 119–130 (2010)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
24. Sundaramoorthi, G., Yezzi, A., Mennucci, A.C.: Sobolev active contours. *IJCV* 73(3), 345–366 (2007)
25. Sundaramoorthi, G., Yezzi, A., Mennucci, A.C., Sapiro, G.: New possibilities with Sobolev active contours. In: *Proceedings of the 1st International Conference on Scale Space Methods and Variational Methods in Computer Vision* (2007)
26. Tai, X.-C., Christiansen, O., Lin, P., Skjælaaen, I.: Image segmentation using some piecewise constant level set methods with mbo type of projection. *International Journal of Computer Vision* 73(1), 61–76 (2007)
27. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision* 50(3), 271–293 (2002)
28. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization* 14(4), 1043–1056 (2004)

# A Tensor Variational Formulation of Gradient Energy Total Variation

Freddie Åström<sup>1,2</sup>, George Baravdish<sup>3</sup>, and Michael Felsberg<sup>1</sup>

<sup>1</sup> Computer Vision Laboratory, Linköping University, Sweden

<sup>2</sup> Center for Medical Image Science and Visualization (CMIV), Linköping University

<sup>3</sup> Department of Science and Technology, Linköping University, Sweden

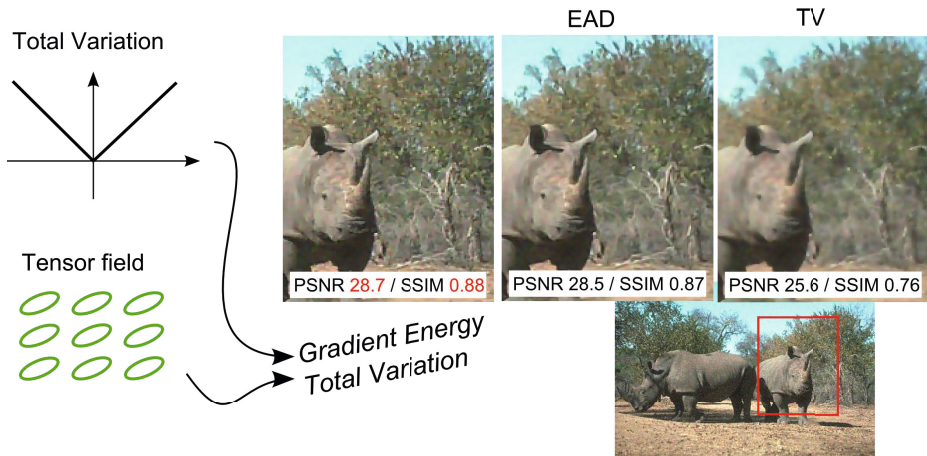
**Abstract.** We present a novel variational approach to a tensor-based total variation formulation which is called *gradient energy total variation*, GETV. We introduce the gradient energy tensor [6] into the GETV and show that the corresponding Euler-Lagrange (E-L) equation is a tensor-based partial differential equation of total variation type. Furthermore, we give a proof which shows that GETV is a convex functional. This approach, in contrast to the commonly used *structure tensor*, enables a formal derivation of the corresponding E-L equation. Experimental results suggest that GETV compares favourably to other state of the art variational denoising methods such as *extended anisotropic diffusion* (EAD) [1] and *total variation* (TV) [18] for gray-scale and colour images.

## 1 Introduction

The variational approach to image diffusion is to model an energy functional  $E(u) = F(u) + \lambda R(u)$  where  $F(u)$  is a fidelity term. The positive constant  $\lambda$  determines the influence of  $R(u)$ , the regularization term describing smoothness constraints on the solution  $u^*$  that minimizes  $E(u)$ . In this work we are interested in tensor-based formulations of the regularization term and we introduce the functional *gradient energy total variation* (GETV).

A basic approach to remove additive image noise is to convolve the image data with a low-pass filter *e.g.* a Gaussian kernel. The approach has the advantage that noise is eliminated, but so is image structure. To tackle this drawback, Perona and Malik [16] introduced an edge stopping function to limit the filtering where the image gradient takes on large values. A tensor-based extension of the Perona and Malik formulation was presented by Weickert [20] in the mid 90s which we refer to, in accordance to Weickert, as anisotropic diffusion.

The principle of anisotropic diffusion is that smoothing of the image is performed parallel to image structure. The concept is based on the structure tensor [2,7], a windowed second moment matrix, which describes the local orientation in terms of a tensor field. To smooth the image parallel to the image structure, the tensor field is transformed by using a non-linear diffusivity function. The transformed tensor field is then used in the diffusion scheme and the resulting tensor is commonly denoted as the diffusion tensor. In this paper we propose to replace the structure tensor and introduce the *gradient energy tensor* (GET) [6] into the



**Fig. 1.** In this work we derive a gradient energy tensor-total variation (GETV) scheme. In this example, our approach clearly boosts the visual impression, PSNR and SSIM performance over EAD and TV in a colour image *auto-denoising* application.

regularization term of a total variation energy functional. The formulation allows us to consider *both* the eigenvalues and eigenvectors, in contrast to previous work which only considers the eigenvalues of the structure tensor. We formulate a gradient energy total variation functional and show significant improvements over current variational state-of-the-art denoising methods. In figure 1 we illustrate the denoising result obtained by the proposed gradient energy total variation formula compared to extended anisotropic diffusion (EAD) [1] and total variation (TV) [18], note that our approach obtain higher error measures and sharper edges.

## 1.1 Related Works

The linear diffusion scheme (convolution with a Gaussian kernel) is the solution of a partial differential equation (PDE) and it is closely related to the notion of scale-space [11]. Therefore it has been of interest to investigate also the Perona and Malik formulation and its successors of adaptive image filtering in terms of a variational framework.

In the linear diffusion scheme the regularization terms are given by  $R(u) = \int_{\Omega} |\nabla u|^2 dx$  and in total variation (TV)  $R(u) = \int_{\Omega} |\nabla u| dx$ , see [18]. The total variation formulation is of particular interest since it has a tendency to enforce piecewise smooth surfaces, however it is also the drawback since it produces cartoon-like images.

Several works have investigated generalizations of the standard total variation approach to tensor-based formulations. Roussos and Maragos [17], Lefkimmatis et al. [14] and Grasmair and Lenzen [9], all consider the *structure tensor* and

define objective functions in terms of the tensor eigenvalues. The difference from those work compared to our presentation is that our formulation allows us to consider both eigenvalues and eigenvectors of the *gradient energy tensor*.

Roussous and Maragos [17] considered a functional which indirectly takes the eigenvalues of the structure tensor into account and ignores the eigenvectors. They considered the regularization  $R(u) = \int_{\Omega} \psi(\mu_1, \mu_2) dx$ , where  $\mu_{1,2}$  are the eigenvalues of the structure tensor. They remark that standard variational calculus tools are not applicable to derive the Euler-Lagrange equation. The problem arise when computing the structure tensor where a smooth kernel is convolved with the image gradients. Furthermore, Lefkimmiatis et al. which, similar to Roussous and Maragos, considered Schatten-norm of the structure tensor eigenvalues. Grasmair and Lenzen [9] defined  $R(u) = \int_{\Omega} \sqrt{\nabla^t u A(u) \nabla u} dx$ , where  $A(u)$  is the structure tensor with remapped eigenvalues. The objective function is then solved using a finite element method instead of deriving a variational solution. Krajsek and Scharr [13], linearized the diffusion tensor thus they obtained a linear anisotropic regularization term resulting in an approximate formulation of a tensor-valued functional for image diffusion.

The common formulation by the aforementioned works is that they use the structure tensor which does not allow for an explicit formal derivation of the Euler-Lagrange equation. Our framework does.

## 1.2 Contributions

The approach we take in this work is to introduce the *gradient energy tensor* (GET) [6] into the regularization term of the proposed functional. We give a proof which shows that the GETV is a convex functional. Our formulation allows us to differentiate *both* the eigenvalues and eigenvectors since the GET does *not* (in this work) contain a post-convolution of its components. The following major contributions are presented

- We present a novel objective function *gradient energy total variation* which models a tensor-based total variation diffusion scheme by using the gradient energy tensor in section 4.
- In section 5, we show that the new scheme combines EAD [1] and TV [18] achieving highly competitive results for grey and colour image denoising.

## 2 Variational Approach to Image Enhancement

### 2.1 Energy Minimization

The variational framework of image diffusion is based on functionals of the form

$$E(u) = \int_{\Omega} (u - u^0)^2 dx + \lambda R(u), \quad (1)$$

where  $x = (x_1, x_2)^t \in \Omega \subset \mathbb{R}^2$ ,  $\Omega$  is the image size in pixels and  $u^0$  is the observed noisy image. The first term in (1) is the fidelity term  $F(u)$  and the

second term is the regularization term,  $R(u)$ . The constant  $\lambda > 0$  determines the amount of regularization. The stationary point that minimizes  $E(u)$  is given by the Euler-Lagrange (E-L) equation

$$\lim_{\varepsilon \rightarrow 0} \frac{\partial E(u + \varepsilon v)}{\partial \varepsilon} = 0, \quad \text{in } \Omega, \quad (2)$$

where  $v$  is a test-function. The corresponding boundary condition to (2) is a homogeneous Neumann condition *e.g.*  $\nabla u \cdot n = 0$  where  $n$  is the normal vector on the boundary  $\partial\Omega$  and  $\nabla = (\partial_{x_1}, \partial_{x_2})^t$  is the gradient operator. The E-L equation for total variation [18] is

$$\begin{cases} u - u^0 - \lambda \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) = 0 & \text{in } \Omega \\ \nabla u \cdot n = 0 & \text{on } \partial\Omega, \end{cases} \quad (3)$$

and the  $u$  which minimizes (3) can be obtained by solving a parabolic initial value problem (IVP) to get the diffusion equation. Alternatively, TV is often solved by using primal-dual formulations [4] or by modifying its norm to include a constant offset to avoid discontinuous solutions.

## 2.2 Tensor-Based Anisotropic Diffusion

In order to filter parallel to the image structure, a tensor-based anisotropic diffusion scheme was introduced by Weickert [20]. The filtering scheme is defined as the partial differential equation (PDE)

$$\operatorname{div}(D(T)\nabla u) = 0, \quad \text{in } \Omega. \quad (4)$$

The adaptivity of the filter is determined by the structure tensor

$$T = w * (\nabla u \nabla^t u), \quad (5)$$

where  $*$  is a convolution operator and  $w$  is a Gaussian kernel [7,2]. The tensor is a windowed second moment matrix, thus it estimates the local variance and can be thought of as describing a covariance matrix [8]. The eigenvector of  $T$  corresponding to the largest eigenvalue is aligned orthogonal to the image structure. Therefore, to avoid blurring of image structures, the diffusion tensor  $D$  is computed as  $D(T) = U^T g(\Lambda) U$ , where  $U$  is the eigenvectors and  $\Lambda$  the eigenvalues of  $T$  [5]. We require  $g(r) \rightarrow 1$  as  $r \rightarrow 0$  and  $g(r) \rightarrow 0$  as  $r \rightarrow \infty$  and a common choice is the negative exponential function  $g(r) = \exp(-r/k)$  where  $k$  is an unknown edge-stopping parameter.

## 3 Gradient Energy Tensor

The gradient energy tensor is a real-valued and symmetric tensor and it determines the directional energy distribution of the signal gradient [6]. In contrast



to the structure tensor (5) it does *not* require a post-convolution of the tensor-components to form a rank 2 tensor. Note that due to the convolution operator, the structure tensor is not sensitive to structures smaller than the width of the averaging filter used to compute it. The classical GET is defined in terms of the image data in  $u$  [6]. Here we use an alternative, but equivalent, formulation of GET expressed in the image gradient. Let  $H = \nabla\nabla^t$  be the Hessian and  $\nabla\Delta u = \nabla\nabla^t\nabla u$ , then we define the GET as

$$GET(\nabla u) = HuHu - \frac{1}{2}(\nabla u[\nabla\Delta u]^t + [\nabla\Delta u]\nabla^t u). \tag{6}$$

The presence of second and third-order derivatives in GET makes it sensitive to noise, however, it allows us to capture orientation of structures that are not possible to detect with the structure tensor. In general the GET is not positive semi-definite. An investigation of the positivity of the 1-dimensional energy operator was done in [3]. In the two-dimensional case, the positivity of the operator is reflected in the sign of the eigenvalues. Let the components of the GET be  $a, b$  and  $c$  *i.e.*  $GET := \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ , then the GET is positive semi-definite if the condition in Lemma 1 is satisfied.

**Lemma 1.** *The GET is positive semi-definite if  $\text{tr}(HuHu) - \nabla^t u \nabla\Delta u \geq \sqrt{l}$  where  $l = \text{tr}(GET)^2 - 4\det(GET) \geq 0$ .*

*Proof.* Since GET is symmetric it has real eigenvalues. Thus by its eigenvalue decomposition it is sufficient to show that  $\text{tr}(GET) \geq \sqrt{l}$  in order for GET to be positive semi-definite.  $l$  is necessarily positive since  $l = (a - c)^2 + 4b^2 \geq 0$ .

Since  $GET$  is not necessarily positive semi-definite we define  $GET^+$ , in the below definition, which is a positive semi-definite tensor.

**Definition 1.** *The positive semi-definite tensor  $GET^+$  is*

$$GET^+(\nabla u) = V^t \begin{pmatrix} |\iota_1| & 0 \\ 0 & |\iota_2| \end{pmatrix} V \tag{7}$$

where  $V$  is the eigenvectors and  $\iota_{1,2}$  are eigenvalues of  $GET$ .

## 4 Introducing Gradient Energy Total Variation

In this section we introduce the proposed energy functional which results in the gradient energy tensor-based total variation scheme. The regularization term we consider is given in the following definition

**Definition 2.** *The gradient energy total variation functional ( $GETV$ ) is*

$$R(u) = \int_{\Omega} \nabla^t u S(\nabla u) \nabla u \, dx, \tag{8}$$

where  $S(\nabla u) \in \mathbb{R}^{2 \times 2}$  is a symmetric positive semi-definite tensor.

#### 4.1 Variational Formulation of Gradient Energy Total Variation

In this section we will study properties and interpretation of the GETV before deriving its corresponding Euler-Lagrange equation in the next section.

We begin our analysis by putting  $S(\nabla u) \in \mathbb{R}^{2 \times 2}$  to be the symmetric positive semi-definite tensor in (8) with eigenvalues  $\mu_{1,2}$  and orthonormal eigenvectors  $v, w$  then

$$S(\nabla u) = vv^t \mu_1 + ww^t \mu_2.$$

Furthermore, we define a tensor  $W(\nabla u) \in \mathbb{R}^{2 \times 2}$  which also is symmetric positive semi-definite with corresponding eigenvectors to  $S(\nabla u)$  *i.e.*

$$W(\nabla u) = vv^t \lambda_1 + ww^t \lambda_2,$$

and  $\lambda_{1,2}$  are the eigenvalues. In particular we will consider  $W(\nabla u)$  of the form

$$W(\nabla u) = |\nabla u| S(\nabla u), \quad (9)$$

such that (8) is convex (see Corollary 1).

Start by expressing the quadratic form defined in  $S(\nabla u)$  by its eigendecomposition and rearranging the resulting vectors such that

$$\begin{aligned} \nabla^t u S(\nabla u) \nabla u &= \nabla^t u [\mu_1 vv^t + \mu_2 ww^t] \nabla u \\ &= \mu_1 v^t [\nabla u \nabla^t u] v + \mu_2 w^t [\nabla u \nabla^t u] w, \end{aligned} \quad (10)$$

The product  $\nabla u \nabla^t u$  is a rank-1 tensor with orthonormal eigenvectors  $p = (p_1, p_2)^t$  and  $p^\perp = (p_2, -p_1)^t$  such that  $P = (p, p^\perp)$  and  $\Lambda$  has the corresponding eigenvalues  $\kappa_1$  and  $\kappa_2$  on its diagonal. Note that, by the spectral theorem, the eigendecomposition of  $\nabla u \nabla^t u$  is always well-defined, *i.e.* the eigenvector  $p$  is not singular. This is shown by the generalized definition of  $p$ , *i.e.* in the case of  $|\nabla u| \neq 0$  then  $p = \nabla u / |\nabla u|$ , and in the case of  $|\nabla u| = 0$  we let  $P = I$  where  $I$  is the identity matrix

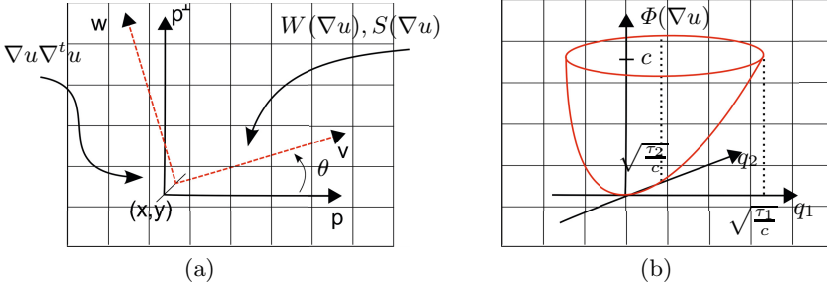
In the following we substitute the eigendecomposition of  $\nabla u \nabla^t u = P^t \Lambda P$  into (10):

$$\begin{aligned} \nabla^t u S(\nabla u) \nabla u &= (\mu_1 v^t P \Lambda P^t v + \mu_2 w^t P \Lambda P^t w) \\ &= \left( \mu_1 v^t P \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} P^t v + \mu_2 w^t P \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} P^t w \right) \end{aligned} \quad (11)$$

and insert  $\kappa_1 = |\nabla u|^2$  and  $\kappa_2 = 0$  and use the eigenvalue relation from (9) *i.e.*  $\lambda_1 = \mu_1 |\nabla u|$  and  $\lambda_2 = \mu_2 |\nabla u|$ . Then after rewriting (11) we obtain

$$\nabla^t u S(\nabla u) \nabla u = \lambda_1 |\nabla u| (v \cdot p)^2 + \lambda_2 |\nabla u| (w \cdot p)^2. \quad (12)$$

The interpretation of (12) is that  $v, p, w$  are normalized eigenvectors such that the scalar products defines the rotation of  $W(\nabla u)$  and  $S(\nabla u)$  in relation to the image gradients direction. This can be illustrated by using the definition of the scalar product *i.e.*  $v \cdot p = \cos(\theta)$  and  $w \cdot p = \sin(\theta)$ , where  $\theta$  is the rotation



**Fig. 2.** (a) Illustration of eigenvector basis at coordinate  $(x, y)$ , the dashed (red) arrows indicate the eigenvectors of  $W(\nabla u)$  and  $S(\nabla u)$  and thick (black) arrows  $\nabla u \nabla^t u$ . (b) Illustration of the paraboloid (15) where we set  $c = \tau_1 q_1^2 + \tau_2 q_2^2$ .

angle as shown in figure 2 a. Note that, if  $W(\nabla u)$  describes the local directional information its eigenvectors will be parallel to the orthonormal eigenvectors of  $\nabla u \nabla^t u$ , i.e.  $v \parallel p$  and  $w \parallel p^\perp$  if  $\theta = 0$ .

In the following we set  $W(\nabla u)$  in (9) as

$$W(\nabla u) = \exp(-GET^+(\nabla u)/k), \tag{13}$$

where  $\exp$  is the matrix exponential function such that  $\lambda_i = \exp(-|\iota_i|/k)$  for  $i = 1, 2$  and  $k > 0$ , the eigenvalues  $\iota_i$  were defined in (7). In the below Corollary we put  $W(\nabla u)$  as (13) and show that  $\Phi(\nabla u)$  is convex and thereby  $R(u)$  is convex in  $u$ .

**Corollary 1.** *The GETV functional,  $R(u)$ , is convex w.r.t.  $u$ .*

*Proof.* To prove the convexity of  $R(u)$  we write  $\Phi(u) = \nabla^t u S(\nabla u) \nabla u$  in terms of the eigenvectors and eigenvalues of  $W(\nabla u)$ . Then, from (12) it follows that

$$\begin{aligned} \Phi(\nabla u) &= |\nabla u|(\lambda_1 p^t (v v^t) p + \lambda_2 p^t (w w^t) p) \\ &= |\nabla u| p^t (\lambda_1 v v^t + \lambda_2 w w^t) p \\ &= (V p)^t \begin{pmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{pmatrix} V p, \end{aligned} \tag{14}$$

where  $V = (v, w)$  and  $\tau_i(\nabla u) = |\nabla u| \lambda_i \geq 0$  for  $i = 1, 2$ . Let  $q = V p = (q_1, q_2)^t$ , then

$$\Phi(\nabla u) = \tau_1 q_1^2 + \tau_2 q_2^2, \tag{15}$$

is a quadratic form in the basis of orthonormal eigenvectors  $V$  and  $\tau_i(\nabla u)$ . This quadratic form is always well-defined due to the spectral theorem. Since the paraboloid (15) has positive curvature everywhere, and  $u$  maps continuously to the paraboloid,  $R(u)$  is convex in  $u$  which concludes the proof.  $\square$

We illustrate  $\Phi(\nabla u)$  by the paraboloid in figure 2 b.

*Remark 1.*  $\Phi(\nabla u)$  can also be expressed in the Schatten-1 norm [10] (pp. 441) i.e.  $\|A\|_1 = \sum_i^n |\sigma_i(A)|$  where  $\sigma_i(A)$  denotes the  $i$ 'th singular value of a tensor  $A$ . Since  $\|A\|_1$  is a norm it has the important properties of positivity and convexity. However, in our case it is not obvious that the convexity follow directly from the norm due to the non-linearity of  $W(\nabla u)$ , see Corollary 1. From (12) we have that

$$\nabla^t u S(\nabla u) \nabla u = \|A(\nabla u)\|_1. \tag{16}$$

This means that  $A(\nabla u)$  has singular values  $\sigma_1(A) = \lambda_1 |\nabla u| (v \cdot p)^2$  and  $\sigma_2(A) = \lambda_2 |\nabla u| (w \cdot p)^2$  where  $\lambda_{1,2}$  are the eigenvalues of  $W(\nabla u)$  in (9).

*Remark 2.* The standard total variation formulation is obtained from (9) by setting  $W(\nabla u)$  as the identity matrix  $I$ , then we have  $\Phi(\nabla u) = \nabla^t u \frac{I}{|\nabla u|} \nabla u = \frac{|\nabla u|^2}{|\nabla u|} = |\nabla u|$ . Notice that we can derive the same result from (16). Suppose that  $W(\nabla u) = I$ , then  $\lambda_{1,2} = 1$  and  $\sigma_1(A) = |\nabla u|$  and  $\sigma_2(A) = 0$ , thus we obtain that  $\Phi(\nabla u) = \|A(\nabla u)\|_1 = |\nabla u|$ .

### 4.2 A Formal Minimizer of Gradient Energy Total Variation

In the previous section we defined the GETV in definition 2 by putting  $S(\nabla u)$  according to (9) with  $W(\nabla u)$  as in (13). In order to minimize our proposed functional we use a result from [1] which derived the corresponding Euler-Lagrange equation for a functional with a quadratic form. Therefore we use this result to directly minimize (17) in the below Theorem 1 in order to compute the Euler-Lagrange equation of (8). Note that Theorem 1 is restated from [1] but with the difference that the tensor  $S$  is symmetric.

**Theorem 1.** *Let the regularization term  $R(u)$  in the functional (1), be given by*

$$R(u) = \int_{\Omega} \nabla^t u S(\nabla u) \nabla u \, dx, \tag{17}$$

where  $u \in \mathcal{C}^2$  and  $S(\nabla u) \in \mathbb{R}^{2 \times 2}$  is a tensor-valued function  $\mathbb{R}^2 \rightarrow \mathbb{R}^{2 \times 2}$ . Set  $\mathbf{s} = \nabla u$  and define

$$S_{\mathbf{s}}(\mathbf{s}) = \begin{pmatrix} \nabla^t u S_{s_1}(\mathbf{s}) \\ \nabla^t u S_{s_2}(\mathbf{s}) \end{pmatrix}. \tag{18}$$

where  $S_{s_1}$  is defined as the component-wise differentiation of  $S$  with respect to  $s_1$ . Then the corresponding E-L equation is given by

$$\begin{cases} \operatorname{div}(B \nabla u) = 0 \text{ in } \Omega \\ \mathbf{n} \cdot B \nabla u = 0 \text{ on } \partial \Omega, \end{cases} \tag{19}$$

where  $B = [2S + S_{\mathbf{s}}]_{\mathbf{s}=\nabla u}$ .

By using Theorem 1 we compute  $S_{\mathbf{s}}$  as

$$S_{\mathbf{s}} = -\frac{1}{|\nabla u|^3} \begin{pmatrix} u_x \nabla^t u W \\ u_y \nabla^t u W \end{pmatrix} + \frac{1}{|\nabla u|} \begin{pmatrix} \nabla^t u W_{u_x} \\ \nabla^t u W_{u_y} \end{pmatrix}, \quad (20)$$

so that the corresponding minimizer of the regularizer (17) is obtained by inserting (20) into (19)

$$\left\{ \begin{aligned} \operatorname{div} \left( \underbrace{\left[ 2W + \begin{pmatrix} \nabla^t u W_{u_x} \\ \nabla^t u W_{u_y} \end{pmatrix} - \frac{1}{|\nabla u|^2} \begin{pmatrix} u_x \nabla^t u W \\ u_y \nabla^t u W \end{pmatrix} \right]}_{=Q} \frac{\nabla u}{|\nabla u|} \right) &= 0 \text{ in } \Omega \\ n \cdot B \nabla u &= 0 \text{ on } \partial \Omega, \end{aligned} \right. \quad (21)$$

where the bracket, which we denote as  $Q$ , defines a weight controlling the anisotropy of the total variation scheme. We compute the component-wise derivative of  $W$  with respect to  $u_x$  and  $u_y$  by using an explicit eigendecomposition *i.e.*

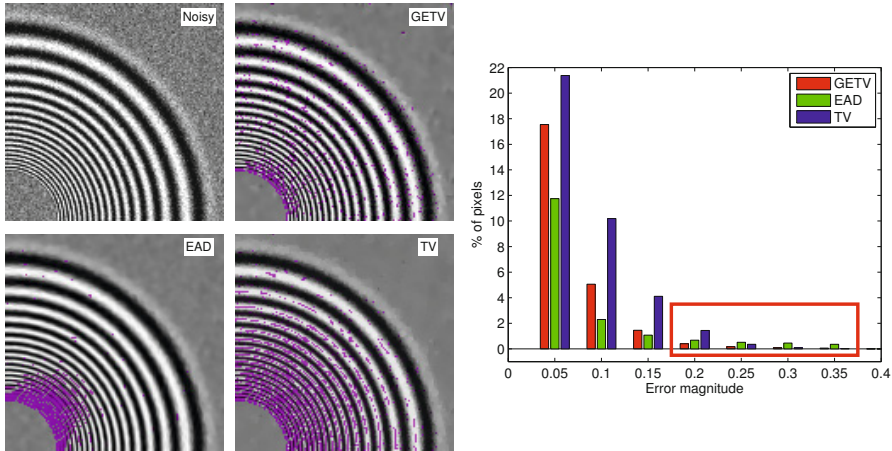
$$\begin{aligned} W_{u_x} &= \left[ \begin{pmatrix} 2v_1 & v_2 \\ v_2 & 0 \end{pmatrix} (\partial_{u_x} v_1) + \begin{pmatrix} 0 & v_1 \\ v_1 & 2v_2 \end{pmatrix} (\partial_{u_x} v_2) \right] \lambda_1 + v v^t (\partial_{u_x} \lambda_1) \\ &+ \left[ \begin{pmatrix} 2w_1 & w_2 \\ w_2 & 0 \end{pmatrix} (\partial_{u_x} w_1) + \begin{pmatrix} 0 & w_1 \\ w_1 & 2w_2 \end{pmatrix} (\partial_{u_x} w_2) \right] \lambda_2 + w w^t (\partial_{u_x} \lambda_2), \end{aligned} \quad (22)$$

with the corresponding orthonormal eigenvectors  $v$  and  $w$ . The general expressions for the derivatives of the eigenvalues and eigenvectors are given in the supplementary material.

The most intuitive interpretation of the GETV is to consider the eigendecomposition of  $W(\nabla u)$ . Thus given eigenvalues  $\lambda_{1,2} = \exp(-|\iota_{1,2}|/k)$  of  $W(\nabla u)$  where  $\iota_{1,2}$  are eigenvalues computed from the gradient energy tensor, the exponential function will adapt the filtering to be parallel to the image structures *i.e.* close to an image structure  $\lambda_1$  will be small and  $\lambda_2$  larger. Since the gradient energy tensor does not contain a post-convolution of the tensor-components, our formulation allows us to better preserve fine details in the image structure, than if we would use the structure tensor, as we show in the numerical experiments.

### 4.3 Discretization

The proposed PDE (21) is solved with a forward Euler-scheme and the image derivatives are approximated by using regularized finite differences [21]. The numerical approximation of the divergence operator is based on the expansion  $\operatorname{div}(M \nabla u) = \partial_x(M_{11} \partial_x u) + \partial_x(M_{12} \partial_y u) + \partial_y(M_{21} \partial_x u) + \partial_y(M_{22} \partial_y u)$ . The first and last term in the previous equation are computed by averaging the forward  $\partial^+$  and backward  $\partial^-$  finite difference operators, and the mixed derivatives are computed with central differences. The final E-L equation that we solve is (21) but with regularized derivatives, *i.e.* let  $\beta$  denote regularization with a small positive constant such that the denominators are expressed as  $|\nabla u|_{\beta} = \sqrt{|\nabla u|^2 + \beta^2}$ .



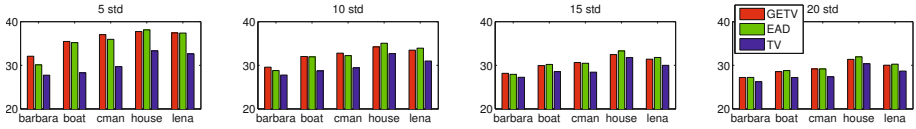
**Fig. 3.** A test image corrupted with 20 standard deviation additive Gaussian noise and the corresponding denoised results. Observe that the errors larger than 10% (magenta) are considerably less concentrated at high frequencies for the GETV than the other methods.

Furthermore, we are required to compute third-order derivatives (in GET) for terms such as  $\partial_x \Delta u$ , and we found that it is appropriate to directly approximate these higher-order derivatives with central differences of the Laplacian. In practice, to avoid numerical instabilities, it is sufficient to regularize the first and second order derivatives with a Gaussian filter of standard deviation  $\sigma$  of 8/10. To compute the third-order term a Gaussian filter of standard deviation 3 was suitable for regularization. These filter sizes were kept constant for all images and all noise levels in the experimental evaluation, we fixed  $\beta = 10^{-4}$ .

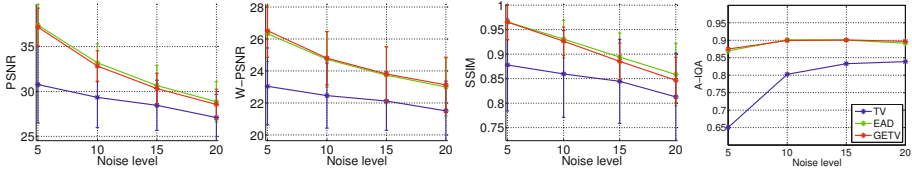
## 5 Application to Image Enhancement

We evaluate our approach with respect to extended anisotropic diffusion (EAD) [1] and a state-of-the-art primal-dual implementation of the Rudin-Osher-Fatemi [18] total variation model (TV) [4]<sup>1</sup>. In figure 3 we illustrate the behaviour of the three schemes on a radial test pattern consisting of increasingly high-frequency components. The histogram illustrates that the proposed gradient energy total variation (GETV) scheme in essence exhibits fewer large magnitude errors than the other methods, this is marked by the red box. The EAD scheme shows errors in high-frequency areas as illustrated with the magenta colour, whereas standard total variation gives errors for all frequencies due to the tendency of enforcing piece-wise constant surfaces.

<sup>1</sup> Code (14-02-17) [gpu4vision.icg.tugraz.at/index.php?content=downloads.php](http://gpu4vision.icg.tugraz.at/index.php?content=downloads.php)



(a) PSNR values for the grey-scale images.



(b) Mean and standard deviation of error measures for the Berkeley dataset.





**Fig. 4.** Error measures, a higher W-PSNR indicates better recovery of high-frequency regions. The visual appearance for selected images is shown in figure 6.

## 5.1 Experiments' Setup

**Datasets** that we consider are twofold. First we consider a number of standard grey-scale images *barbara*, *boat*, *cameraman* (*cman*), *house* and *lena*, each image is of size  $256 \times 256$ . The other dataset is the Berkeley database [15], where we randomly choose 50 colour images each of size  $481 \times 321$ . In the evaluation we corrupted each image with Gaussian additive noise of standard deviations 5, 10, 15 and 20. The images that we have used are listed in the supplementary material. In this work we use the decorrelation CIELAB transform, however other choices of colour spaces are possible as investigated in [1].

**Error measures** are in the image processing community recognised to not correlate with perceived image quality, therefore we investigate several error measures and consider the visual image quality. PSNR is widely used in the denoising literature so we report it, as well as, the structure similarity index (SSIM) [19] known to better reflect the true image quality. Also, since large homogeneous regions have more impact on the error measures than edges in the image do, we compute a weighted PSNR, W-PSNR, to assess preservation of edges in the images after filtering. The weight we use is given by the trace of the structure tensor and it is applied on the difference between the original and the enhanced image in the computation of the PSNR. Since the trace measures the magnitude of the gradient, the W-PSNR value correlates with a better preservation of edges than the PSNR measure does in relation to the noise-free image.

**Image auto-denoising** is used to optimize the selection of parameters in the different filtering schemes. It is a method which does not take the noise-free image into account when determining a quality measure. In this work we use the image auto-denoising metric proposed in [12], which we denote as A-IQA (auto-image quality assessment). The basic idea of A-IQA is that a high correlation score is obtained if the denoised image has smooth surfaces, but yet preserves boundaries. In the total variation scheme, we select the regularization parameter  $\lambda$  from the

Noisy	EAD	GETV	TV
			
A-IQA: -	<b>0.91</b>	0.90	0.84
PSNR: 24.84	30.5	<b>30.6</b>	28.4
PSNR-W: 63.30	63.44	<b>64.12</b>	60.43
SSIM: 0.21	0.91	<b>0.92</b>	0.88

**Fig. 5.** Example from the grey-scale dataset for 15 standard deviation of noise. GETV shows an improvement over EAD and TV, both in PSNR and preservation of fine image structures such as the camera handle. Also, note that the images obtained with GETV appear less blurry than EAD and TV.





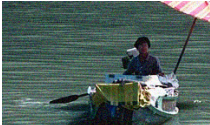



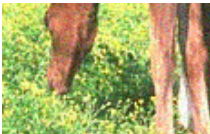

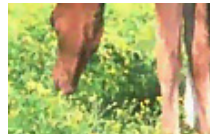
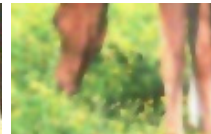
values 6, 8, 10, 12 and 14 based on maximum A-IQA. The control parameter  $k$  in the EAD scheme is computed according to  $k = (\exp(1) - 1)/(\exp(1) - 2)\sigma^2$  [5] where  $\sigma$  is the standard deviation of the added noise. The  $k$  obtained for EAD is also used in the proposed GETV scheme but scaled with a factor  $10^{-1}$ . The stopping time for all methods was determined by the maximum A-IQA value.

## 5.2 Result of Image Denoising

In figure 4 (a) we show the PSNR values that we have obtained for each grey-scale image and noise level. We observe that the standard TV formulation does not perform well compared to EAD and GETV in these cases. In figure 5 we show close-ups of *cameraman*. We note that in all cases the error measures are similar for the A-IQA values, however considering the visual quality it is obvious that more details are preserved in GETV, *i.e.* the presence of sharp edges in the *cameraman* image such as the handle of the camera.

With respect to the colour images, figure 6 shows examples from the Berkeley dataset and the corresponding error measures are given in figure 4 (b). By comparing EAD and GETV for lower noise levels (5-15 standard deviations) we see that the difference in PSNR and SSIM is at best marginal. However, considering the variance, GETV is more robust than EAD. In figure 6 the visual differences can be seen for some selected images. Note that it is primarily in the high-frequency regions that GETV excels, consider *e.g.* the clarity of the document, the visibility of waves and details in the grass in the horse image. For both grey-scale and colour images, EAD tends to oversmooth the images. Furthermore, it is obvious that the TV-method fails to handle these images when auto-tuning is used. By manually tweaking the regularization parameter of the methods we can improve the error measures for some images, however this approach is infeasible for a large amount of images.



Noisy	EAD	GETV	TV
			
A-IQA: -	0.85	<b>0.87</b>	0.76
PSNR: 25.01	<b>30.5</b>	30.4	28.3
PSNR-W: 22.57	23.08	<b>23.45</b>	21.89
SSIM: 0.65	<b>0.90</b>	0.89	0.86
			
A-IQA: -	0.81	<b>0.84</b>	0.67
PSNR: 24.93	29.5	<b>29.6</b>	26.8
PSNR-W: 22.54	23.22	<b>23.42</b>	21.50
SSIM: 0.68	<b>0.87</b>	<b>0.87</b>	0.77
			
A-IQA: -	0.79	<b>0.83</b>	0.72
PSNR: 24.73	27.4	<b>28.6</b>	23.9
PSNR-W: 22.41	22.22	<b>22.93</b>	20.00
SSIM: 0.69	0.84	<b>0.86</b>	0.72

**Fig. 6.** Results from the Berkeley colour-image dataset with 15 standard deviation of noise where GETV excels. Consider particularly the text on the document and the grass behind the horse on the last row. Note that GETV in general preserves more fine details than EAD and TV, which both tends to oversmooth the images.

## 6 Conclusion

In this work we have presented a novel variational approach to tensor-based total variation. In particular, we have proposed a *gradient energy total variation* functional which uses the gradient energy tensor. Our results suggest that the GETV formulation is suitable for images containing high-frequency information such as fine structures. Secondly, we showed by using the error measure A-IQA that the diffusion formulation performs well in denoising applications compared to EAD and TV.

**Acknowledgement.** This research has received funding from the Swedish Foundation for Strategic Research through the grant VPS and from Swedish Research Council through grants for the projects energy models for computational cameras (EMC<sup>2</sup>), Visualization-adaptive Iterative Denoising of Images (VIDI) and Extended Target Tracking (ETT), all within the Linnaeus environment CADICS and the excellence network ELLIIT.

## References

1. Åström, F., Baravdish, G., Felsberg, M.: On Tensor-Based PDEs and Their Corresponding Variational Formulations with Application to Color Image Denoising. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 215–228. Springer, Heidelberg (2012)
2. Bigun, J., Granlund, G.H.: Optimal Orientation Detection of Linear Symmetry. In: Proceedings of the IEEE First ICCV, pp. 433–438 (1987)
3. Bovik, A., Maragos, P.: Conditions for positivity of an energy operator. IEEE Transactions on Signal Processing 42(2), 469–471 (1994)
4. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. JMIV 40(1), 120–145 (2011)
5. Felsberg, M.: Autocorrelation-driven diffusion filtering. Image Processing 20(7), 1797–1806 (2011)
6. Felsberg, M., Köthe, U.: GET: The connection between monogenic scale-space and Gaussian derivatives. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) Scale-Space 2005. LNCS, vol. 3459, pp. 192–203. Springer, Heidelberg (2005)
7. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: ISPRS Intercommission, Workshop, Interlaken, pp. 149–155 (1987)
8. Gårding, J., Lindeberg, T.: Direct computation of shape cues using scale-adapted spatial derivative operators. IJCV 17(2), 163–191 (1996)
9. Grasmair, M., Lenzen, F.: Anisotropic Total Variation Filtering. Applied Mathematics & Optimization 62(3), 323–339 (2010)
10. Horn, R.A., Johnson, C.R. (eds.): Matrix Analysis. Cambridge University Press, New York (1986)
11. Iijima, T.: Basic Theory on Normalization of a Pattern (in Case of Typical One-Dimensional Pattern). Bulletin of the Electrotechnical Lab 26, 368–388 (1962)
12. Kong, X., Li, K., Yang, Q., Liu, W., Yang, M.H.: A new image quality metric for image auto-denoising. In: ICCV (2013)
13. Krajssek, K., Schar, H.: Diffusion Filtering Without Parameter Tuning: Models and Inference Tools. In: CVPR 2010, San Francisco, pp. 2536–2543 (2010)
14. Lefkimmatis, S., Roussos, A., Unser, M., Maragos, P.: Convex Generalizations of Total Variation Based on the Structure Tensor with Applications to Inverse Problems. In: Pack, T. (ed.) SSVM 2013. LNCS, vol. 7893, pp. 48–60. Springer, Heidelberg (2013)
15. 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: Proc. ICCV, vol. 2, pp. 416–423 (July 2001)
16. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. PAMI 12, 629–639 (1990)
17. Roussos, A., Maragos, P.: Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals. In: ICIP, pp. 4141–4144 (2010)
18. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1–4), 259–268 (1992)
19. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. TIP 13(4), 600–612 (2004)
20. Weickert, J.: Anisotropic Diffusion In Image Processing. ECMI Series. Teubner-Verlag, Stuttgart (1998)
21. Weickert, J.: Nonlinear diffusion filtering. In: Jähne, B., Haussecker, H., Beissler, P. (eds.) Signal Processing and Pattern Recognition. Handbook of Computer Vision and Applications, ch. 15, pp. 423–451. Academic Press (1999)

# Color Image Segmentation by Minimal Surface Smoothing<sup>\*</sup>

Zhi Li<sup>1</sup> and Tiejong Zeng<sup>2</sup>

<sup>1</sup> Department of Mathematics, Hong Kong Baptist University, Hong Kong, China  
zhi\_li@life.hkbu.edu.hk

<sup>2</sup> Department of Mathematics, Hong Kong Baptist University, Hong Kong, China  
zeng@hkbu.edu.hk

**Abstract.** In this paper, we propose a two-stage approach for color image segmentation, which is inspired by minimal surface smoothing. Indeed, the first stage is to find a smooth solution to a convex variational model related to minimal surface smoothing. The classical primal-dual algorithm can be applied to efficiently solve the minimization problem. Once the smoothed image  $u$  is obtained, in the second stage, the segmentation is done by thresholding. Here, instead of using the classical K-means to find the thresholds, we propose a hill-climbing procedure to find the peaks on the histogram of  $u$ , which can be used to determine the required thresholds. The benefit of such approach is that it is more stable and can find the number of segments automatically. Finally, the experiment results illustrate that the proposed algorithm is very robust to noise and exhibits superior performance for color image segmentation.

**Keywords:** image segmentation, minimal surface, primal-dual method, total variation.

## 1 Introduction

Image segmentation is a fundamental and challenging topic in image processing, and the minimal surface theory has been applied in this area for many years [6, 10, 11, 39]. To begin with, we will briefly review some related image segmentation models, such as edge-based and region-based approaches.

The Geodesic Active Contour (GAC) model [10, 11] is a classical edge-based model. This model employs the intrinsic geometry nature of the image to compute minimal distance curves. The authors smooth the image before performing the segmentation, and put considerable efforts to construct appropriate terms with minimal surface property to advance the level set function. Basically, they first apply a simple smooth procedure to the image, then provide the level set framework to solve a sophisticated model. The re-initialization procedure is usually used to prevent the level set function generating undesired shapes during the

---

<sup>\*</sup> This work is supported in part by the National Science Foundation of China (11271049), by HKRGC 211710 and 211911, by the FRGs of Hong Kong Baptist University, and by the HKPFS from HKRGC.

curve evolution. However, the re-initialization process has its own side effects, Li et al. [24] suggested a new variational formulation to avoid this procedure.

One of the most popular region-based models is the Chan-Vese (CV) model [15] based on the piecewise constant Mumford-Shah (MS) functional [15], where Chan and Vese implement the level set method.

Moreover, in order to segment a given image  $f$ , the MS model [27] deals with the minimization of the following energy,

$$E(u, \Gamma) = \int_{\Omega} (u - f)^2 dx + \mu \int_{\Omega - \Gamma} \|\nabla u\|^2 dx + \lambda |\Gamma|,$$

where  $u$  approximates  $f$ ,  $\mu, \lambda$  are positive parameters,  $\Gamma$  is the collection of the boundaries of the partitions  $\Omega_i$  inside  $\Omega$ ,  $|\Gamma|$  denotes the summation of the lengths of  $\Omega_i$ ,  $\cup_{i=1}^K \Omega_i \cup \Gamma = \Omega$ , and  $\Omega_i \cap \Omega_j = \emptyset$ ,  $\forall i \neq j$ , and  $K$  is the number of partitions.

The MS model accomplishes two major purposes: one is to find a piecewise smooth approximation  $u$  of the image  $f$ . This means  $u$  is differentiable on  $\cup \Omega_i$ , the variation of  $u$  on each  $\Omega_i$  is small, and  $u$  can be discontinuous across the boundaries of each  $\Omega_i$ . The other purpose is to find an edge set  $\Gamma$  separating  $u$ , while  $|\Gamma|$  is as short as possible.

A simplified version of the MS model is the piecewise constant MS model, and this is also known as the cartoon limit problem. The energy is rewritten as

$$E(c_i, \Gamma) = \sum_{k=1}^K \int_{\Omega_i} (c_i - f)^2 dx + \lambda \sum_{k=1}^K |\partial \Omega_i|, \quad (1)$$

where  $K$  is the number of partitions,  $|\partial \Omega_i|$  measures the perimeter of each  $\Omega_i$ ,  $u = c_i$  is a constant on each open set  $\Omega_i$ .

The CV model is equivalent to minimizing (1) with an additional area-penalizing term. However, the model is hard to solve in consequence of the two difficulties that (i) the non-convexity of the energy, (ii) solving the two unknowns (a function  $u$  and a contour  $\Gamma$ ) with different natures at the same time.

The difficulty (i) may lead to a local minimum, and due to the difficulty (ii), the CV model has to solve  $m$  ODEs for  $n = 2^m$  partitions, because  $m$  level set functions  $\varphi_j$  ( $j = 1, \dots, m$ ) must be used. During each iteration time step, and in each ODE, the non-smooth high-order derivative term,  $\operatorname{div}(\frac{\nabla \varphi_j}{|\nabla \varphi_j|})$  has to be computed. Practically speaking, as a component of the 'image force', this term is implemented by a matrix (the same size as the image) contains the mean curvature of each level set function approximated by a finite difference scheme.

Cai et al. [9] suggested a two-stage segmentation method to conquer the two difficulties. In the first stage, they minimize a convex functional based on the MS energy to obtain  $u$  (the smoothed version of the grayscale image  $f$ ), and K-means in the second stage will choose  $K - 1$  thresholds and segment  $u$  into  $K$  parts. Most importantly, the authors reveal the connection between image segmentation and image restoration, and it is the first time to handle the difficulty (ii) of the CV model properly. Until recently, combined with the 'four color theorem',

the updated CV model [38] can segment any number of regions with four indicator functions. But implementation of the 'four color theorem' will add extra workload. Moreover, the users have to compute the whole model again, if they want different segmentation results, while in the two-stage method [9], since the smoothed  $u$  is already computed in the first stage, the user can adjust the second stage to give new thresholds for a different segmentation very quickly. For the undesired non-convexity of the CV model, there are continuing works [4, 8, 14] to improve the CV model based on different convex relaxation methods.

Besides of edge-based models and region-based models, there are also other important segmentation methods, such as graph-based models. Yuan et al. [36, 37] proposed the continuous max-flow and min-cut approach. This method can get a global minimum solution, and includes the above-mentioned [14] as a special case, but in the max-flow algorithm [37], the indicator function is implemented by a 3-dimensional matrix, called the label function. The size of the first 2 dimensions of this matrix is the same as the size of the image which related to each pixel, and the size of the third dimension is equal to the number of segments. This matrix has to be updated during each iteration, thus this algorithm is not an easy breezy for the multiphase case. And it worth mentioning that Boykov [6] extended the minimal surface idea to graph cuts.

In this paper, we adopt the strategy in [9] to propose a new two-stage segmentation method. We improve [9] by three aspects. First, instead of two parameters in the first step in [9], we use the minimal surface energy which has only one parameter. Second, we use the concept of vectorial total variation (VTV) to handle the color image segmentation problem, which is ignored in [9]. Third, in the second step, we propose to use the hill-climbing method to replace the K-means and thus we can find the required thresholds automatically and stably.

The rest of the paper is organized as follows. In Section 2, we review the VTV and some thresholding methods. Then we present our method stage by stage in Section 3. We suggest the convex model with the minimal surface property for the first stage, and show that the model has a unique solution, then the primal-dual schema is adopted to solve the discrete version of the convex model. For the second stage, we give the detailed implementation of the hill-climbing algorithm. In Section 4, we compare our method with a nonconvex segmentation model [33]. Numerical experiments conform that the smoothing process is good enough, and a straightforward segment method can give satisfactory results from the smoothed image. Conclusions are given in Section 5.

## 2 Preliminaries

In this section, we recall the concept of VTV and the hill-climbing method.

### 2.1 Vectorial Total Variation

The vectorial total variation [7] is useful for color image processing. Let  $\Omega$  be a bounded convex region in  $R^2$ , a given function  $u : \Omega \rightarrow R^M$ . The VTV for a color image  $u$  can be defined as:

$$\int_{\Omega} |Du| := \sup_{v \in V} \left\{ \int_{\Omega} -\langle u, \operatorname{div} v \rangle dx \right\}, \quad (2)$$

where  $V = \{v | v := (v_1, \dots, v_M), v \in C_c^1(\Omega; R^{M \times 2}) : |v| \leq 1\}$ , refer to [3, 7] for more details. The divergence operator  $\operatorname{div} v := (\operatorname{div} v_1, \dots, \operatorname{div} v_M) : \Omega \rightarrow R^M$ . The gradient operator  $\nabla u := ((\nabla u)^1, (\nabla u)^2) : \Omega \rightarrow R^{M \times 2}$ .

## 2.2 Some Thresholding Methods

Now we review some thresholding methods. The image is smoothed delicately in the first stage, and in the second stage, we plan to adopt a simple algorithm to segment the well smoothed image. Thresholding is considered as the simplest image segmentation method, we will revisit some thresholding algorithms, and show that the hill-climbing method is a more suitable one.

We compare the famous K-means [9, 20, 31] with the hill-climbing method [2, 17, 29], and outline the advantages of the hill-climbing method. First, the K-means model is NP-hard, and for efficiency consideration, it usually comes with parallel execution support. The hill-climbing algorithm we consider detects peaks of the color histogram, thus it is easier to compute because of dealing with limited number of bins not all the pixels. Second, the K-means algorithm may stick in a local minimum because of the nonconvexity of the model. For this reason, it usually suffers from bad results because of the required misleading initial values. While the hill-climbing algorithm will find all of the fixed peaks, once the histogram is fixed. Third, K-means has to take another input parameter  $K$ , and clusters the pixels into  $K$  groups. The hill-climbing algorithms only need the histogram of the image, initial center values and the number of thresholds are not required as input parameters. Considering the drawbacks of K-means, the hill-climbing method is a better choice.

There are other histogram based thresholding methods. The image histogram acting as a representation of pixel distribution (e.g., the gray value of the pixels in the gray scale image) can be a useful tool for thresholding, for example, the Otsu's method [30], and it was extended to deal with color images in [28]. The authors separate the RGB channels and smooth them independently, after apply their multi-level thresholding method, a region merging algorithm has to be used to overcome over segmentation. While the hill-climbing algorithm [29] deals with the 3D histogram instead of the independent channels, only those peaks approved by all 3 channels are recorded, then the over-cut is avoided. Starting from a random bin, the algorithm compares with the neighborhood bins to find the uphill direction and moves along that direction until a peak is reached. The bins that lead to the same peak are placed in the same group.

## 3 The Proposed Two-Stage Method

In this section, we will show the details of our two-stage segmentation method. We will not try to achieve the two goals of finding the smooth approximation

$u$  of the image  $f$  and an edge set separating  $u$  at the same time, such as the MS model. Alternatively, we first apply a sophisticated smoothing process to clear out the singularities and useless details from the image, and our convex model leads to a unique smooth image  $u$ . Here we will solve a single saddle-point problem not an ODE system like in the CV model. What's more, since we do not consider the segmentation in this step, there is no large storage matrix for implementation of the indicator functions, such as the max-flow algorithm. In the second stage, we employ a simple hill-climbing algorithm to search the  $K$  peaks, and use them as thresholds to segment  $u$  into  $K$  parts, no complex computations involved in this step.

### 3.1 First Stage

Most images are deteriorated, and contain useless details for image segmentation. For this reason, we will choose a convex model which is robust to noise like the CV model, and we search the image manifold for the surface with a minimal area, as a result, singularities are smoothed, while main features are kept.

Inspired by the two-stage algorithm [9] and MS functional [27], we consider the model

$$E(u) = \lambda \|u - f\|_2^2 + \int_{\Omega} \sqrt{1 + |\nabla u|^2} dx, \quad (3)$$

where  $\Omega \subset R^2$  is a bounded open connected set (in most cases, is a rectangle),  $u$  is the approximation of the image  $f : \Omega \rightarrow R^M$ ,  $M = 1, 3$ . The first term is the data fidelity term. It provides an appropriate measure of the difference between the image  $f$  and the approximation image  $u$ , and the second term gives  $u$  reasonable smoothness [5].  $\lambda$  is a positive parameter to make sure the solution does not deviate too much from the measured data  $f$ , and also drops some information to get a simple  $u$ . This process can be seen as applying image restoration ideas to image segmentation. This model is first brought up in [7] as an extension of the vectorial ROF model for image denoising.

The first three parts of section 3.1 are basically following [1] and [7]. We briefly show how to include minimal surface property to the VTV, and we call it the QMSVTV (Quasi-Minimal Surface Vectorial Total Variation). Then we consider the dual form of the QMSVTV, and show the model (3) has a unique solution. Finally, we employ the primal-dual approach to solve discrete setting of the convex model in the last part of section 3.1.

**Quasi-Minimal Surface Vectorial Total Variation.** As pointed out in [7, 11, 21], coupling all the color information is very important for the VTV. First, we assume that all channels contribute equally to the VTV. Second, all channels gain smoothness in scale, which implies that we need to minimize energy function involving the gradient of all channels. Last but not the least, we argue that easy-to-compute is also a significant consideration in constructing the energy.

Starting from modern Riemannian differential geometry, Kimmel et al. [21] considered images as Riemannian manifolds. In this way, the image can be seen as a 2-dimensional surface embedded in a  $(2 + M)$ -dimensional space:  $(x, y) \rightarrow (x, y, \beta u_1(x, y), \dots, \beta u_M(x, y))$ . By taking account of the metric, they proposed the area norm [21, 32] as follow

$$S = \int_{\Omega} \sqrt{1 + \beta^2 \sum_{i=1}^M |\nabla u_i|^2 + \beta^4 \frac{1}{2} \sum_{i,j=1}^M (\nabla u_i, \nabla u_j)^2},$$

where  $\beta$  is a constant scale factor multiplying the intensity values. In this paper, we use

$$S := \int_{\Omega} \sqrt{1 + \sum_{i=1}^M |\nabla u_i|^2}, \tag{4}$$

and call it the QMSVTV, because of the ease of implementation and each channel is equally smoothed, most importantly, the minimal surface property is well preserved.

There are other literatures on the minimal area/mean curvature, for example, Zhu, Tai and Chan [39] proposed a denoising model employing the mean curvature of the image surface, and this model can preserve corners and image contrast. With  $\Gamma$ -convergence technique and the minimal surfaces theory, Kluzner et al. [22, 23] provided an alternative for the MS functional.

**Dual of Quasi-Minimal Surface Vectorial Total Variation.** Because the Fenchel transform of the convex functional  $f(x) = \sqrt{1 + |x|^2}$  is

$$\sqrt{1 + |x|^2} = \sup \left\{ x \cdot y + \sqrt{1 - |y|^2} : y \in R^M \ |y| \leq 1 \right\},$$

and the supremum is attained when  $y = \frac{x}{\sqrt{1+|x|^2}}$  [16]. Similar to Section 2.1, we can define [1]

$$J(u) := \sup_{v \in V} \int_{\Omega} \left( -u \operatorname{div} v + \sqrt{1 - |v|^2} \right) dx. \tag{5}$$

From Theorem 2.1 in [1], we know

$$J(u) = \int_{\Omega} \sqrt{1 + |\nabla u|^2}, \quad u \in W^{1,1}(\Omega).$$

**Existence and Uniqueness of the Model (3)**

**Proposition 1.** *The energy functional  $E$  in (3) is weakly lower semicontinuous with respect to the  $L^1$  topology.*



*Proof.* Let  $u_n \rightharpoonup \bar{u}$  (weak convergence in  $L^1(\Omega)$ ). For any  $v \in V$ ,  $\operatorname{div} v \in C(\Omega)$ , and from Fatou’s lemma,

$$\begin{aligned} & \lambda \|\bar{u} - f\|_2^2 + \int_{\Omega} \left( -\bar{u} \operatorname{div} v + \sqrt{1 - |v|^2} \right) \\ &= \lim_{n \rightarrow \infty} \lambda \|\bar{u} - f\|_2^2 + \int_{\Omega} \left( -u_n \operatorname{div} v + \sqrt{1 - |v|^2} \right) \\ &\leq \liminf_{n \rightarrow \infty} \lambda \|u_n - f\|_2^2 + J(u_n). \end{aligned}$$

Taking the supremum over  $v \in V$ , and referring to (3), we get

$$\begin{aligned} E(\bar{u}) &= \lambda \|\bar{u} - f\|_2^2 + \sup_{v \in V} \int_{\Omega} \left( -\bar{u} \operatorname{div} v + \sqrt{1 - |v|^2} \right) \\ &\leq \liminf_{n \rightarrow \infty} \lambda \|u_n - f\|_2^2 + J(u_n). \end{aligned}$$

□

**Proposition 2.** *The energy functional  $E$  in (3) is strictly convex.*

*Proof.* The proof is obvious from the definitions. □

**Theorem 1.** (*Existence-uniqueness*) *The minimization problem  $\inf_{u \in BV(\Omega)} E(u)$  has a unique minimizer.*

*Proof.* From Theorem 3.1 and Theorem 4.1 in [1], we know there is a  $u^* = \operatorname{argmin}_{u \in L^1(\Omega)} E(u)$ , and  $E(u) = \lambda \|u - f\|_2^2 + \int_{\Omega} \sqrt{1 + |\nabla u|^2} dx < \infty$ , then  $\int_{\Omega} \sqrt{1 + |\nabla u^*|^2} < \infty$ .

We use the dual relationship, and get  $\sup_{v \in V} \int_{\Omega} \left( -u^* \operatorname{div} v + \sqrt{1 - |v|^2} \right) dx < \infty$ , then  $\sup_{v \in V} \int_{\Omega} (-u^* \operatorname{div} v) dx < \infty$ , hence  $u^* \in BV(\Omega)$ . Since  $BV(\Omega) \subset L^1(\Omega)$ ,  $u^* = \operatorname{argmin}_{u \in BV(\Omega)} E(u)$ . Uniqueness of minimizers follows immediately from strict convexity. □

**The Discrete Setting and Implementation Details.** In this section, we will give the discrete version of the model (3), and solve it with the primal-dual algorithm to get the smoothed image.

The discrete setting of the model (3) is

$$\min_u \lambda \|u - f\|_2^2 + \sqrt{1 + \|\nabla u\|_2^2}, \tag{6}$$

the detailed definitions of the operators can be found in section 2.1.

There are many algorithms, such as [12, 18], can be adopted to solve this convex minimization problem, similar to [13], we choose the primal-dual algorithm [12] because of its easy implementation, and we can easily proof the convergence of the algorithm.

We write the saddle-point formulation of (6)

$$\max_q \min_{u,p} \lambda \|u - f\|_2^2 + \sqrt{1 + \|p\|_2^2} + \langle p - \nabla u, q \rangle. \tag{7}$$

Applying the primal-dual method, we arrive at the following 3 minimization subproblems:

$$\begin{aligned} q^{k+1} &= \operatorname{argmax}_q \left\{ \langle \bar{p}^k - \nabla \bar{u}^k, q \rangle - \frac{1}{2\sigma} \|q - q^k\|_2^2 \right\} \\ u^{k+1} &= \operatorname{argmin}_u \left\{ \lambda \|u - f\|_2^2 + \langle -\nabla u, q^{k+1} \rangle + \frac{1}{2\tau} \|u - u^k\|_2^2 \right\} \\ p^{k+1} &= \operatorname{argmin}_p \left\{ \sqrt{1 + \|p\|_2^2} + \langle p, q^{k+1} \rangle + \frac{1}{2\tau} \|p - p^k\|_2^2 \right\} \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k \\ \bar{p}^{k+1} &= 2p^{k+1} - p^k. \end{aligned}$$

Algorithm 1 summarizes the procedure of solving the 3 minimization subproblems.

---

**Algorithm 1.** Solving (7) by the primal-dual algorithm

---

**1. Initialization**

**2. Solve the 3 subproblems:**

Do  $k = 0, 1, \dots$ , until  $\frac{\|u^{k+1} - u^k\|}{\|u^{k+1}\|} < \epsilon$

- 1) solve the 1st subproblem:  $q^{k+1} = \sigma (\bar{p}^k - \nabla \bar{u}^k) + q^k$
- 2) solve the 2nd subproblem:  $u^{k+1} = \frac{1}{2\lambda\tau + 1} (2\lambda\tau f - \tau \operatorname{div} q^{k+1} + u^k)$
- 3) solve the 3rd subproblem: using the Newton iteration
- 4) update  $\bar{u}^{k+1}, \bar{p}^{k+1}$

**3. Output:**  $u$

---

**3.2 Second Stage**

In this stage, the hill-climbing method finds thresholds from the smoothed image  $u$  got from the first stage. The image matrix  $u$  is grouped into a 3D histogram, then the hill-climbing algorithm attempts to locate the peaks on the color histogram. These peaks are used as thresholds to segment the image based on different distance measures. The advantage of this step is that, for different segmentation results, just simply regroup  $u$  into a different histogram, and the

hill-climbing algorithm will find a new set of peaks to thresholding the smoothed image data  $u$ . Figure 1 shows the peak for the 1 channel case, for a single bin, it only has up to two neighbors (Figure 2).

The hill-climbing method in this paper is presented in Algorithm 2. A few modification is made to the hill-climbing algorithm in [29]. (i) We loop through each bin of the 3D histogram to find the peaks, if the number of pixels falling into the current bin is larger than all it neighbors', then we consider there is a peak in this bin [19], and the possible neighbors for the 3 channels case is shown in Figure 2. In [29], the authors use a Dijkstra-like algorithm to move along the uphill direction until a peak was reached, then start search again from an unclimbed bin until all bins have been visited. This algorithm requires extra data structures, such as the stack, to store those unclimbed bins, and this will cause inefficiency for the numerical experiments here. (ii) We average the intensity values of the pixels falling into the peak bin, and use this average value as the peak. After getting all the peaks, we compute the distance between these peaks and each pixel in the image under certain measure. And the pixel is associated with the nearest peak. While in [29], they do not compute the peaks, all bins leading to certain peak bin are grouped together, then it is unfair for those pixels falling into the valley bin, because some of them may belong to the other group.

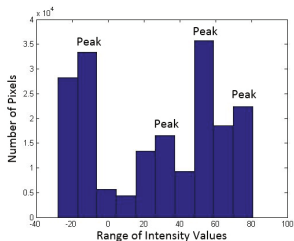


Fig. 1. histogram for 1 channel

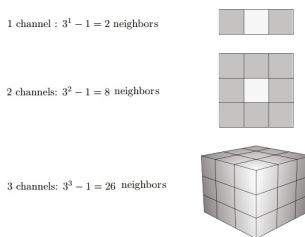


Fig. 2. number of neighbors

---

**Algorithm 2** The hill-climbing algorithm for the color image

---

**1. Initialization:**

Group image  $I$  into matrix  $A(N, N, N)$  ( $N$  is the number of bins per channel)

$A(i, j, k)$  = the number of pixels fall into the  $(i, j, k)$  bins/matrix

**2. Find peaks of  $A$ :**

For all  $A(i, j, k)$

If  $A(i, j, k) >$  all its neighbors

1) average the intensity values of those pixels in the  $(i, j, k)$  bin

2) store this average value as a peak

EndIf

EndFor **3. Output:** peaks  $c$

---

## 4 Experimental Results

In this section, to demonstrate the superior performance of our convex model, we compare the proposed method with two state of the art methods: the SW-Potts model [33] and the FRC (Fuzzy Region Competition) model [25]. The SW-Potts model uses the dynamic programming and the ADMM algorithm to minimize the Potts model which is non-convex. And the FRC model is solved by the alternative minimization method. To standardize the experiments, all test images are transferred to the Lab space from the RGB color space except in the 3-SHAPE example, because we add noise to each RGB channel of the 3-SHAPE image.

### 4.1 2-phase Segmentation

The PLANE image in the first row of Figure 3 is from the Berkeley segmentation dataset [26]. It can be seen that our method cuts the plane from the background sky, and the picture is divided into two reasonable parts, while the FRC model fails and the SW-Potts model over cuts the image into 5 parts.

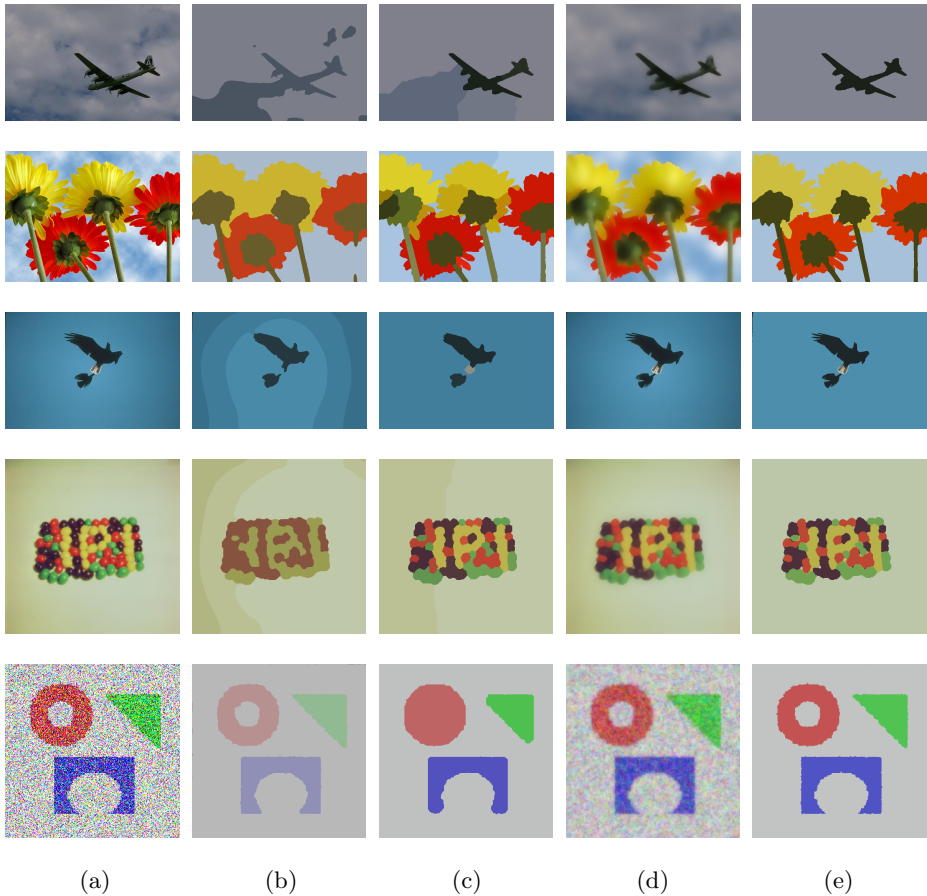
### 4.2 Multi-phase Segmentation

The FLOWER image in the second row of Figure 3 is a commonly used image for testing variational image processing techniques. Our method successfully segments the picture into 4 parts: the sky is grouped into one part, the flowers are well cut by colors, and the stems and torus are count as a whole. But the FRC model loses some detail features and the SW-Potts model divides the image into 51 segments after we carefully choose the parameter. Because of the convexity of the target energy, and taking the advantage of the uniqueness for the minimizer, our algorithm finds the optimal solution, instead of getting stuck in one of the local minima.

The BIRD image in the third row of Figure 3 is also from the Berkeley segmentation dataset. Our segmentation result exhibits the gray color on the tails and also the detailed shape of feathers on wings. The FRC model fails to show the tails. Although the SW-Potts model also divides the image into 4 parts, the result is vague.

The PELLET image in the fourth row of Figure 3 is from the USC-SIPI image database [35]. Our result presents decent segmentation based on the color of the pellets, but the other two models do not get reasonable segments, for example, the SW-Potts model gets 37 different colors.

The 3-SHAPE image in the fifth row of Figure 3 is an extension of the grayscale example in [34]. The target image is corrupted by strong Gaussian noise with  $m = 20$ ,  $\sigma = 200$ . We use our model (6) to preprocess it, and notice that the majority of noise is smoothed out. Then we use the hill-climbing algorithm to segment the smoothed image into 4 parts separately, and the basic important colors and shapes are revealed. The FRC model recovers and separates the 3 different shapes, but messes up on the image boundary. The SW-Potts model

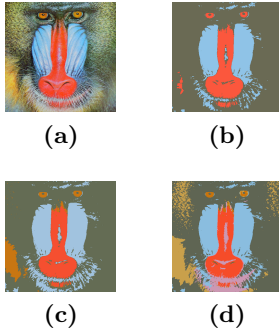


**Fig. 3.** Segmentation comparison: (a) given image, (b) FRC, (c) SW-Potts, (d) solution  $u$  from (6), (e) our method

fails to segment the red ring object and a few details are missing even it also gets 4 segments.

The Table 1 shows the comparison of the CPU time, it is obvious that our method is faster than the FRC model and the SW-Potts model.

Finally, we will show 'one smoothing multiple segmentations', this means we smooth the image once, then adjust the 'light-weight' hill-climbing method to cut the image into different number of parts. The baboon image in Figure 4 is from the USC-SIPI image database, and Figure 4 reveals different levels of features. For instance, compared with Figure 4b (3 segments) and Figure 4c (4 segments), Figure 4d (6 segments) distinguishes between the eyes and the cheek.



**Fig. 4.** Multiple segmentation results

**Table 1.** CPU time (seconds) for Figure 3

Model \ Fig	FRC	SW-Potts	Ours
PLANE	16.45	23.56	<b>7.45</b>
FLOWER	48.41	44.79	<b>9.16</b>
BIRD	34.58	20.32	<b>1.87</b>
PELLET	20.47	8.95	<b>3.77</b>
3-SHAPE	18.96	14.87	<b>0.85</b>

## 5 Conclusion

In this paper, we have proposed a convex image smoothing model followed by a thresholding algorithm which can divide images efficiently. The fast primal-dual algorithm was presented to minimize the convex functional with minimal surface properties. The attractive features of the model guarantee a unique and smooth solution. We have successfully demonstrated our approach on various images. For color images, we improve the hill-climbing algorithm to find more precise centers. And the future work could be using mixed norms for the data-fidelity term to yield better results.

## References

1. Acar, R., Vogel, C.R.: Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems* 10(6), 1217–1229 (1994)
2. Achanta, R., Estrada, F.J., Wils, P., Süsstrunk, S.: Salient region detection and segmentation. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) *ICVS 2008*. LNCS, vol. 5008, pp. 66–75. Springer, Heidelberg (2008)
3. Ambrosio, L., Fusco, N., Pallara, D.: *Functions of bounded variation and free discontinuity problems*, vol. 254. Clarendon Press Oxford (2000)
4. Bae, E., Lellmann, J., Tai, X.-C.: Convex relaxations for a generalized chan-veese model. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMM-CVPR 2013*. LNCS, vol. 8081, pp. 223–236. Springer, Heidelberg (2013)
5. Bae, E., Yuan, J., Tai, X.C.: Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision* 92(1), 112–129 (2011)
6. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 26–33. IEEE (2003)
7. Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging* 2(4), 455–484 (2008)

8. Brown, E.S., Chan, T.F., Bresson, X.: Completely convex formulation of the Chan–Vese image segmentation model. *International journal of computer vision* 98(1), 103–121 (2012)
9. Cai, X., Chan, R., Zeng, T.: A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding. *SIAM Journal on Imaging Sciences* 6(1), 368–390 (2013)
10. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *International Journal of Computer Vision* 22(1), 61–79 (1997)
11. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces: A geometric three dimensional segmentation approach. *Numerische Mathematik* 77(4), 423–451 (1997)
12. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40(1), 120–145 (2011)
13. Chan, R., Yang, H., Zeng, T.: A two-stage image segmentation method for blurry images with Poisson or multiplicative gamma noise. *SIAM Journal on Imaging Sciences* 7(1), 98–127 (2014)
14. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics* 66(5), 1632–1648 (2006)
15. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* 10(2), 266–277 (2001)
16. Deimling, K.: *Nonlinear functional analysis*. Courier Dover Publications (2013)
17. Ding, Z., Jia, J., Li, D.: Fast clustering segmentation method combining hill-climbing for color image (2011)
18. Goldstein, T., Osher, S.: The split Bregman method for  $l_1$ -regularized problems. *SIAM Journal on Imaging Sciences* 2(2), 323–343 (2009)
19. Hu, Y.: Hill-climbing color image segmentation (2008), <http://www.mathworks.com/matlabcentral/fileexchange/22274-hill-climbing-color-image-segmentation>
20. Jin, R., Kou, C., Liu, R., Li, Y.: A color image segmentation method based on improved k-means clustering algorithm. In: Zhong, Z. (ed.) *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*. LNEE, vol. 217, pp. 499–505. Springer, Heidelberg (2013)
21. Kimmel, R.: *Numerical geometry of images: Theory, algorithms, and applications*. Springer (2004)
22. Kluzner, V., Wolansky, G., Zeevi, Y.Y.: Minimal surfaces, measure-based metric and image segmentation. Technion-IIT, Department of Electrical Engineering (2006)
23. Kluzner, V., Wolansky, G., Zeevi, Y.Y.: Geometric approach to measure-based metric in image segmentation. *Journal of Mathematical Imaging and Vision* 33(3), 360–378 (2009)
24. Li, C., Xu, C., Gui, C., Fox, M.D.: Level set evolution without re-initialization: a new variational formulation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 430–436. IEEE (2005)
25. Li, F., Ng, M.K., Zeng, T.Y., Shen, C.: A multiphase image segmentation method based on fuzzy region competition. *SIAM Journal on Imaging Sciences* 3(3), 277–299 (2010)

26. 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: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)
27. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42(5), 577–685 (1989)
28. Nimbarte, N.M., Mushrif, M.M.: Multi-level thresholding algorithm for color image segmentation. In: 2010 Second International Conference on Computer Engineering and Applications (ICCEA), pp. 231–233. IEEE (2010)
29. Ohashi, T., Aghbari, Z., Makinouchi, A.: Hill-climbing algorithm for efficient color-based image segmentation. In: IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 17–22 (2003)
30. Otsu, N.: A threshold selection method from gray-level histograms. *Automatica* 11(285-296), 23–27 (1975)
31. Singh, M., Patel, P., Khosla, D., Kim, T.: Segmentation of functional mri by k-means clustering. *IEEE Transactions on Nuclear Science* 43(3), 2030–2036 (1996)
32. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. *IEEE Transactions on Image Processing* 7(3), 310–318 (1998)
33. Storath, M., Weinmann, A.: Fast partitioning of vector-valued images. *SIAM Journal on Imaging Sciences* 7(3), 1826–1852 (2014)
34. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision* 50(3), 271–293 (2002)
35. Weber, A.: The usc-sipi image database. Signal and Image Processing Institute of the University of Southern California (1997), <http://sipi.usc.edu/services/database>
36. Yuan, J., Bae, E., Tai, X.C.: A study on continuous max-flow and min-cut approaches. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2217–2224. IEEE (2010)
37. Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A continuous max-flow approach to potts model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 379–392. Springer, Heidelberg (2010)
38. Zhang, R., Bresson, X., Chan, T.F., Tai, X.C.: Four color theorem and convex relaxation for image segmentation with any number of regions. *Inverse Problems and Imaging* 7(3), 1099–1113 (2013)
39. Zhu, W., Tai, X.C., Chan, T.: Augmented lagrangian method for a mean curvature based image denoising model. *Inverse Problems and Imaging* 7(4), 1409–1432 (2013)



# Domain Decomposition Methods for Total Variation Minimization

Huibin Chang<sup>1</sup>, Xue-Cheng Tai<sup>2</sup>, and Danping Yang<sup>3</sup>

<sup>1</sup> School of Mathematical Sciences, Tianjin Normal University, Tianjin, P.R. China  
changhuibin@gmail.com

<sup>2</sup> Department of Mathematics, University of Bergen, Bergen, Norway  
tai@math.uib.no

<sup>3</sup> Department of Mathematics, East China Normal University, Shanghai, P.R. China  
dpyang@math.ecnu.edu.cn

**Abstract.** In this paper, overlapping domain decomposition methods (DDMs) are used for solving the Rudin-Osher-Fatemi (ROF) model in image restoration. It is known that this problem is nonlinear and the minimization functional is non-strictly convex and non-differentiable. Therefore, it is difficult to analyze the convergence rate for this problem. In this work, we use the dual formulation of the ROF model in connection with proper subspace correction. With this approach, we overcome the problems caused by the non-strict-convexity and non-differentiability of the ROF model. However, the dual problem has a global constraint for the dual variable which is difficult to handle for subspace correction methods. We propose a stable unit decomposition, which allows us to construct the successive subspace correction method (SSC) and parallel subspace correction method (PSC) based domain decomposition. Numerical experiments are supplied to demonstrate the efficiency of our proposed methods.

**Keywords:** ROF Model, Dual Formulation, Domain Decomposition Methods (DDMs), Successive Subspace Correction (SSC), Parallel Subspace Correction (PSC).

## 1 Introduction

The ROF model [22] plays an important role in image restoration, which is also instrumental in boosting the use of total variation (TV)-regularization in other image processing tasks. Over the last two decades, many fast and efficient algorithms have been proposed to solve the ROF model.

In general, these algorithms can be classified into three categories based on the nature of manipulating the primal and dual variables and one can refer to [23]. The first category is the primal approach, such as the gradient descent method (cf. [1,19,22,30,31,32,33]). In order to accelerate these methods, the most popular algorithms were proposed based on Bregman iteration (cf. [14,34,35]), augmented Lagrangian methods (cf. [15,36]), graph cuts method, additive operator splitting (AOS), and multigrid method. A concise outline of these approaches

can be found in [37]. The second one is the dual approach. A typical and efficient approach (see, e.g., [5]) is to apply the KKT condition to the dual formulation for ROF model, which allows to solve the dual variable. The last one is the primal-dual approach. This type of approach was introduced in [2,3]. Extensive applications to image processing were studied in [4,39]. Domain decomposition methods (DDMs) and multigrid methods are known to be efficient for the computation of different partial differential equations. Their applications to image processing are still very limited. The DDMs can break down a large problem into a sequence of subproblems with much smaller sizes, so better-conditioned solvers can be constructed over each subdomain. They also allow for parallel computations with the pretty good load balance and speed-up efficiency. In this work, we shall use the space decomposition and subspace correction ideas proposed in [24,25,26,27] for general nonlinear minimization problems. Especially, we will use a parallel subspace correction (PSC) algorithm and a successive subspace correction (SSC) algorithm to get some efficient algorithms for solving the ROF models with an explicit estimate of the convergence rate.

Before explaining the details for the approaches we are going to use, we would like to review some existing studies using DDMs for image processing. In [13], DDMs with Dirichlet boundary condition were studied for image denoising related to Gaussian curvature. Overlapping DDMs were used there based on a primal-dual formulation for the anisotropic total variation problem [20]. The PSC and SSC were applied to variational image restoration and segmentation in [9,28,37]. In these applications, the original problem was successfully decomposed into smaller-size subproblems, which were solved in parallel. Especially in [37], a coarse mesh space correction was considered. Xu et al. [38] also applied the DDMs to image deblurring. Chang et al. [6] extended the DDMs for the nonlocal total variation(NLTV) based image restoration, where the authors also pointed out that their proposed DDMs for NLTV can be adopted to solve the ROF model directly. In addition to these works, some variants of the classical DDMs have been proposed in [10,11,12,17]. In [10,11,12], the authors introduced the surrogate functional to form an approximation(or iterative proximity-map) of the subproblems. Then the subproblems were solved by oblique thresholding. Their proposed algorithms were very efficient in image restoration and compress sensing. At the same time, the weak convergence of the algorithm was proved. In [17], they studied the  $TV - L1 - L2$  model. The convergence and monotone decay of the associated energy for SSC were guaranteed. We would also like to mention that a non-overlapping domain decomposition method was proposed by Hintermüller and Langer [18] for the dual formulation of the anisotropical total variation based image denoising. There, fast solvers for the subproblems and convergence analysis of the algorithm were given as well.

It is known that the minimization functional for the ROF model is non-differentiable. It is convex, but not strongly convex in the BV-space (space of functions with bounded variations). The associated partial differential equation is nonlinear degenerate with special anisotropic diffusion effect. We could use space decomposition ideas to get domain decomposition and mutligrd to work

for the ROF model. It is easy to prove convergence for the resulting algorithms, but it is difficult to prove the convergence rate. Normally, strong convexity is needed as in [24,27]. So far, there is no convergence rate estimate for applying DDMs to the ROF model. In this work, we shall rely on the dual model and use proper space decomposition techniques to get a domain decomposition method. We shall also estimate the convergence rate for the proposed algorithms.

In the following, we focus on solving the dual model using subspace correction methods in our paper. The paper is organized as follows. We introduce the dual formulation and then construct the overlapping DDMs in Section 2. Numerical examples for the proposed algorithms are listed in Section 3. At last we conclude this paper in Section 4.

## 2 Dual Formulation and the Overlapping DDMs

The ROF model is to restore a noisy image  $g$  on a domain  $\Omega$  (e.g. in  $\mathbb{R}^2$ ) through the following minimization problem:

$$\min_{u \in BV(\Omega)} \left\{ \lambda TV(u) + \frac{1}{2} \|u - g\|_{L^2(\Omega)}^2 \right\}, \tag{1}$$

where  $\lambda > 0$  and  $BV(\Omega)$  is the space of functions of bounded variation, and the total variation of  $u$  is defined as in [1] by

$$TV(u) := \sup_{\mathbf{p} \in \mathcal{K}} \int_{\Omega} u \operatorname{div} \mathbf{p} \, d\mathbf{x} \quad \text{with} \tag{2}$$

$$\mathcal{K} := \{ \mathbf{p} = (p_1, p_2) \in (C_0^1(\Omega))^2 : |\mathbf{p}| := (p_1^2 + p_2^2)^{1/2} \leq 1 \}. \tag{3}$$

As usual,  $\mathbf{p}$  is known as the dual variable, while  $u$  is the primal variable.

In Chambolle [5], it was observed that the solution for the ROF model can be obtained by solving the following dual problem:

$$\inf_{\mathbf{p} \in \mathcal{K}} \left\{ \int_{\Omega} (\lambda \operatorname{div} \mathbf{p} - g)^2 \, d\mathbf{x} \right\}, \tag{4}$$

and then get  $u$  through:

$$u = g - \lambda \operatorname{div} \mathbf{p}. \tag{5}$$

### 2.1 Problem Description

As the digital images consist of discrete pixels, we study the discrete model in this paper. Hereafter, we use the following notations. For simplicity, set  $\Omega := \{(i, j) \mid 0 \leq i \leq m, 0 \leq j \leq n\}$ , where the image  $g$  has the resolution of  $(m + 1) \times (n + 1)$ . Then define the gradient and divergence of each  $u_{i,j}$  over  $\Omega$  with the Neumann boundary condition as

$$(\nabla u)_{i,j} = ((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2), \quad \forall (i, j) \in \Omega,$$

where

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j}, & i < m, \\ 0, & i = m, \end{cases} \quad (\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j}, & j < n, \\ 0, & j = n. \end{cases}$$

The divergence of  $\mathbf{p} = (p^1, p^2) \in \mathcal{R}^2$  satisfying  $\operatorname{div} = -\nabla^*$  in the discrete form is defined as

$$(\operatorname{div} \mathbf{p})_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & (\text{as } 0 < i < m) \\ p_{i,j}^1 & (\text{as } i = 0) \\ -p_{i-1,j}^1 & (\text{as } i = m) \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & (\text{as } 0 < j < n), \\ p_{i,j}^2 & (\text{as } j = 0), \\ -p_{i,j-1}^2 & (\text{as } j = n). \end{cases}$$

Denote the discrete form for  $\mathcal{K}$  as

$$K := \{ \mathbf{p} : |\mathbf{p}_{i,j}| \leq 1, \forall (i,j) \in \Omega \}, \tag{6}$$

and the total variation in the discrete form as

$$TV_h(u) := \max_{\mathbf{p} \in K} \sum_{(i,j) \in \Omega} u_{i,j} (\operatorname{div} \mathbf{p})_{i,j}.$$

One can give an equivalent definition as

$$TV_h(u) := \sum_{(i,j) \in \Omega} |(\nabla u)_{i,j}|. \tag{7}$$

Then the discrete ROF model reads

$$\min_u \left\{ \lambda TV_h(u) + \frac{1}{2} \|u - g\|_\Omega^2 \right\}, \tag{8}$$

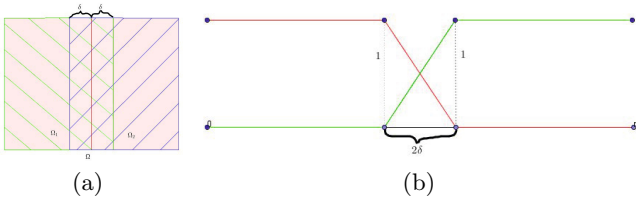
where the  $\|\cdot\|_\Omega$  denotes the discrete  $L^2$  norm over the index set  $\Omega$ . In view of this, we consider the following dual formulation of (8):

$$\min_{\mathbf{p} \in K} \left\{ D(\mathbf{p}) := \sum_{(i,j) \in \Omega} (\lambda (\operatorname{div} \mathbf{p})_{i,j} - (g)_{i,j})^2 \right\}. \tag{9}$$

Notice that the functional  $D(\mathbf{p})$  is convex but not strictly convex, so the problem (9) may have non-unique minimizers in  $K$ . To ensure the uniqueness, one may modify the energy functional (see, e.g., [8,16]) by adding the additional regularization term for  $\mathbf{p}$ . However, it will change the problem. For our proposed algorithms, the non-uniqueness of the minimizer for the dual problem does not pose a problem and we just need one of the minimizers  $\mathbf{p}^*$  to resolve the primal variable via  $u^* = g - \lambda \operatorname{div} \mathbf{p}^*$ . Note that the minimizer for the ROF model is unique.

### 2.2 General Setup for DDMs

To use DDMs, we first decompose the domain  $\Omega$  into subdomains and then use the decomposed subdomains to decompose the constraint set  $K$ . We illustrate a simple case in Figure 1 (a). Here, we first divide the domain into two non-overlapping subdomains by the center red line. Then we extend these two non-overlapping subdomains by including points that have a distant order to decompose the constraint set  $K$ , and we need to use the ‘‘Partition of unity functions’’ (PUFs) with respect to the overlapping subdomains. Figure 1 (b) shows the PUFs  $\theta_1$  and  $\theta_2$  for this special partition. In Figure 2 of Section 3, details about the decomposition used for the numerical experiments will be given.



**Fig. 1.** (a) Domain decomposition  $\Omega = \Omega_1 \cup \Omega_2$  with the overlapping size  $\delta$ ; (b) PUFs  $\theta_1$ (red line), and  $\theta_2$ (green line)

Hereafter, we assume that the discrete computational domain  $\Omega$  has been decomposed to  $M$  overlapping subdomains  $\Omega = \bigcup_{s=1}^M \Omega_s$  ( $M \geq 2$ ) and  $\theta_s$ ,  $s = 1, 2, \dots, M$ , are the PUFs satisfying the following properties:

$$(i) \quad \sum_{s=1}^M (\theta_s)_{i,j} \equiv 1, \quad (\theta_s)_{i,j} \geq 0, \quad \forall (i, j) \in \Omega; \tag{10}$$

$$(ii) \quad \text{supp}(\theta_s) \subset \Omega_s, \quad 1 \leq s \leq M; \tag{11}$$

$$(iii) \quad \|\nabla_h \theta_s\|_\infty \leq \frac{C_0}{\delta}, \quad 1 \leq s \leq M, \tag{12}$$

where  $\nabla_h$  is a proper discrete gradient operator (forward difference, for example),  $C_0$  is a positive constant independent of  $\delta$ , and  $\|\cdot\|_\infty$  is the  $L^\infty$ -norm for discrete functions with values at the grid points. See PUFs in Figure 1 (b) for 1-dimensional case in the discrete setting ( $M = 2$ ).

Using the PUFs, we can define the decomposed constraint sets as

$$K_s = \{\mathbf{p} : |(\mathbf{p})_{i,j}| \leq (\theta_s)_{i,j}, (i, j) \in \Omega\}, \quad 1 \leq s \leq M. \tag{13}$$

Then, it is easy to see that

$$K = \sum_{s=1}^M K_s. \tag{14}$$

This decomposition means that for any  $\mathbf{p} \in K$ , we can find  $\mathbf{p}_s \in K_s$  such that  $\mathbf{p} = \sum_{s=1}^M \mathbf{p}_s$ . In addition, for any  $\mathbf{p}_s \in K_s$ , we have  $\sum_{s=1}^M \mathbf{p}_s \in K$ .

### 2.3 Proposed DDMs

As we have finished the decomposition of domain  $\Omega$  and constraint set  $K$ , we are now ready to present the DDMs for the dual problem (9). Taking  $\alpha > 0$  as the relaxation parameter, the parallel subspace correction (Algorithm I) and successive subspace correction (Algorithm II) algorithms are given below.

---

#### Algorithm I. Parallel Subspace Correction (PSC) Method

1. Initialization: choose  $\mathbf{p}^0$  and select a relaxation parameter  $\alpha$ .
2. For  $n = 0, 1, \dots$ , find  $\mathbf{p}^{n+1}$  in the following two steps:
  - (i) Find  $\{\hat{\mathbf{q}}_s^n\}_{s=1}^M$  in parallel for  $s = 1, 2, \dots, M$ , such that

$$\hat{\mathbf{q}}_s^n = \arg \min_{\mathbf{v} \in K_s} D\left(\mathbf{v} + \sum_{t \neq s} \theta_t \mathbf{p}^n\right), \quad s = 1, 2, \dots, M. \quad (15)$$

- (ii) Compute  $\mathbf{p}^{n+1}$  by

$$\mathbf{p}^{n+1} = (1 - \alpha)\mathbf{p}^n + \alpha \sum_{s=1}^M \hat{\mathbf{q}}_s^n, \quad (16)$$

3. Endfor till some stopping criterion meets.
- 

#### Algorithm II. Successive Subspace Correction (SSC) Method

1. Initialization: choose  $\mathbf{p}^0$  and select a relaxation parameter  $\alpha$ .
2. For  $n = 0, 1, \dots$ , find  $\mathbf{p}^{n+1}$  in the following two steps:
  - (i) Find  $\{\hat{\mathbf{q}}_s^n\}_{s=1}^M$  sequentially for  $s = 1, 2, \dots, M$ , such that

$$\hat{\mathbf{q}}_s^n = \arg \min_{\mathbf{v} \in K_s} D\left(\mathbf{v} + \sum_{t < s} \mathbf{q}_t^n + \sum_{t > s} \theta_t \mathbf{p}^n\right), \quad (17)$$

and then define

$$\mathbf{q}_s^n = (1 - \alpha)\theta_s \mathbf{p}^n + \alpha \hat{\mathbf{q}}_s^n.$$

- (ii) Update

$$\mathbf{p}^{n+1} = (1 - \alpha)\mathbf{p}^n + \alpha \sum_{s=1}^M \hat{\mathbf{q}}_s^n.$$

3. Endfor till some stopping rule meets.

---

In order to guarantee  $\mathbf{p}^{n+1}$  be in  $K$ , it is sufficient to choose  $\alpha \in (0, 1]$ , for Algorithm II and then we have  $|\mathbf{p}^{n+1}| \leq (1 - \alpha)|\mathbf{p}^n| + \alpha \sum_{s=1}^M |\hat{\mathbf{q}}_s^n| \leq 1$ . For Algorithm I, a sufficient condition to guarantee  $\mathbf{p}^{n+1} \in K$  is to choose  $\alpha \in (0, 1/M]$ . In practice, it is possible to choose  $\alpha$  bigger and still retain convergence of the algorithms and guarantee that the solution at convergence is in  $K$ . For both algorithms, bigger relaxation parameter  $\alpha$  will give faster convergence.

We see that the original minimizing of (9), which is normally large in size, is now decomposed into a number of subproblems (15) and (17) with much smaller sizes. Moreover, we can use parallel processor for both algorithms. The parallel degree of Algorithm I is higher than Algorithm II, as the  $M$  sub-minimization problems can be computed in parallel. However, Algorithm I usually converges slower than Algorithm II, since it does not update the iterative solutions in a timely manner. Algorithm II can be also computed using parallel processor if the subdomains are colored properly, see section 3 for some more details. Indeed, subproblems defined on the subdomains with the same color are independent of each other, and therefore can be solved in parallel [6].

The algorithms proposed here is essential the algorithms proposed in [24] applied to decomposition (14) for the dual problem. The essential difference between the above algorithms with those in [24] mainly lies in two aspects. First, the minimization functional is neither strongly and nor strictly convex. Second, the sub-minimization problems (15) and (17) do not have unique minimizers. These bring about significant difficulties for the analysis and lead to different convergence behaviors as well. In a forthcoming paper, we will show some details about the convergence analysis of this algorithm. Letting  $\mathbf{p}^*$  be one of the exact minimizer of the dual problem (which is not unique as well), we can deduce the following results for Algorithm I and Algorithm II:

$$\|u^n - u^*\|_{L^2(\Omega)} \leq \frac{C}{\sqrt{n}}, \tag{18}$$

where  $u^n := g - \lambda \operatorname{div} \mathbf{p}^n$ ,  $u^* := g - \lambda \operatorname{div} \mathbf{p}^*$ ,  $C$  is a positive constant, which mainly depends on parameters  $C_0, M, \alpha, \delta, \lambda$  and the initial value  $\mathbf{p}^0$ . Obviously, the proposed DDMs are convergent with the rates  $O(n^{-1/2})$ .

The algorithm for subproblems of the proposed DDMs is given as follows. In order to solve (15) and (17), we can apply the gradient projection methods (GP) similarly to [5]. One can choose other solvers [7,21] for the subproblem. Here, the divergence for the subproblems is just the restriction of the divergence for the entire domain to the subdomains. We only present the algorithm to solve the subproblems for Algorithm I. Let  $\mathbf{p}$  be the given initial value for per iteration. Denoting  $\mathbf{q}_s^0 := \sum_{t \neq s} \theta_t \mathbf{p}$  and  $g_s = \frac{g}{\lambda} - \operatorname{div} \mathbf{q}_s^0$ , we shall solve the minimization problem as

$$\mathbf{v}_s^* = \min_{\mathbf{v} \in K_s} \left\{ D_s(\mathbf{v}) := \sum_{(i,j) \in \Omega_s} ((\operatorname{div} \mathbf{v})_{i,j} - (g_s)_{i,j})^2 \right\}.$$

By KKT conditions, there exists a Lagrange multiplier  $\mu_s \geq 0$ , such that  $D'_s(\hat{v}_s^* + \mathbf{q}_s^0) + 2\mu_s \hat{v}_s^* = 0$ , associated with either  $\mu_s > 0$  as  $|\hat{v}_s^*| = \theta_s$ , or  $\mu_s = 0$  as  $|\hat{v}_s^*| < \theta_s$ , where  $D'_s(\cdot)$  is the Gâteaux derivative. Thus

$$\theta_i(-\lambda^2 \nabla(\operatorname{div} \hat{v}_s^* - g_s)) + |-\lambda^2 \nabla(\operatorname{div} \hat{v}_s^* - g_s)| \hat{v}_s^* = 0.$$

Therefore, the iterative scheme of semi-implicit gradient descent method is constructed to obtain  $\mathbf{v}_s^*$  as follows:

$$\hat{v}_s^{n+1} = \frac{\hat{v}_s^n + \theta_s \tau (\nabla(\operatorname{div} \hat{v}_s^n - g_s))}{1 + \tau |\nabla(\operatorname{div} \hat{v}_s^n - g_s)|}, \quad n = 0, 1, \dots \tag{19}$$

with suitable iterative step  $\tau > 0$  from the initial value  $\hat{v}_s^0$ . One can readily obtains  $|(\hat{v}_s^{n+1})_{i,j}| \leq \theta_s, \forall (i, j) \in \Omega_s$ , i.e.  $\hat{v}_s^{n+1} \in K_s$ . Following [5], one can prove that the algorithm is convergent if  $\tau \theta_s \leq \frac{1}{8}$ .

### 3 Numerical Experiments

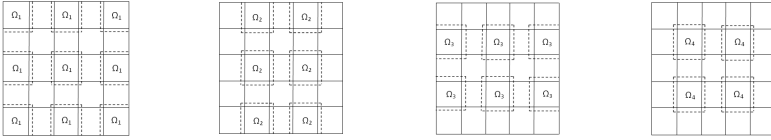
In this section, we supply several numerical experiments to show the performance and convergence of the proposed DDMs.

First, the domain  $\Omega$  is partitioned in a checkerboard-like overlapping-blocks, c.f. Figure 2. Then we extend the non-overlapping blocks to get overlapping blocks. We assume that each of the extended blocks can be painted with one color such that the blocks with the same color will not intersect each other. From the four-color theorem, we know that 4 colors are enough to paint the extended blocks if the overlapping size is not bigger than half of the size of non-overlapping blocks.

We define  $\Omega_i$  to the union of the blocks of the same color. Accordingly, we have  $\Omega = \bigcup_{i=1}^4 \Omega_i$ . Each subdomain  $\Omega_i$ , consisting of  $m_i$  disjoint blocks painted with the same color, i.e. the  $i$ th color, See Figure 2. Hence, the total number of blocks that cover  $\Omega$  is  $m_T = \sum_{i=1}^4 m_i$ . Here the blocks covering  $\Omega$  are defined by  $\{\Omega_j^{block}\}_{j=1}^{m_T}$ . Define  $\Omega_j^{in} = \Omega_j^{block} \setminus (\bigcup_{i \neq j} \Omega_i^{block})$ ,  $\delta_j = \operatorname{dist}(\partial \Omega_j^{block} \setminus \partial \Omega, \Omega_j^{in})$ , and overlapping size  $\delta = \min_{1 \leq j \leq m_T} \delta_j$ . In Figure 2, the blocks (defined by  $\tilde{\Omega}_i$ ) with solid lines are non-overlapping blocks, and the blocks with dotted lines are the overlapping extended blocks. Define *subsize* is the size of non-overlapping blocks, i.e.  $\textit{subsize} = \max_{1 \leq j \leq m_T} \min\{l_x^j, l_y^j\}$ , with the width of non-overlapping

blocks  $\tilde{\Omega}_j^{block}$  (the small regions in Figure 2 with solid lines) in horizontal or vertical directions  $l_x$  and  $l_y$ . If assuming that the original domain  $\Omega$  is square as  $[0, L]^2$  and blocks are decomposed to be squares with same sizes (except for the blocks adjacent to the boundaries of  $\Omega$ ), it holds  $\sqrt{m_T} \times \textit{subsize} = L$ . In the real computation, the domain  $\Omega$  is discretized to be the grid of size  $(W_1 - 1) \times (W_2 - 1)$  if the image has  $W_1 \times W_2$  pixels, i.e. each pixel is considered to be as one grid point





**Fig. 2.** Domain decomposition with coloring technique,  $M = 4, m_1 = 9, m_2 = 6, m_3 = 6, m_4 = 4$  and  $m_T = 25$ . The overlapping blocks (dotted lines) are generated by extending the non-overlapping blocks (solid lines).

and the distance of two adjacent pixels in the horizontal or vertical directions is set to be 1.

We give the numerical examples only by Algorithm II via coloring technique stated above. The image “Cameraman” in Figure 3(a) are tested. Define energy

$$E(\mathbf{p}) := \sum_{i=0}^m \sum_{j=0}^n (\lambda \operatorname{div} \mathbf{p} - g)_{i,j}^2. \tag{20}$$

The image resolution is  $(m + 1) \times (n + 1)$ . The following tests are qualified by the the decreasing of the energy defined by  $Energy := (E(\mathbf{p}_{DDM}) - E(\mathbf{p}^*)) / E(\mathbf{p}^*)$ , where  $\mathbf{p}^*$  is approximated by the GP methods [5] after  $1.0 \times 10^7$  iterations without DDM, and  $\mathbf{p}_{DDM}$  is computed by our proposed DDMs. Assume that the data  $g$  is contaminated by an additive zero-mean white Gaussian noise with standard deviation  $\sigma$ .  $N_{in}$  is the iteration number for subproblems. Set  $\sigma = 50$ ,  $\lambda = 80$ , and the overlapping size  $\delta = 4$ (unless specified differently).

### 3.1 Performance and Convergence of the Proposed DDMs

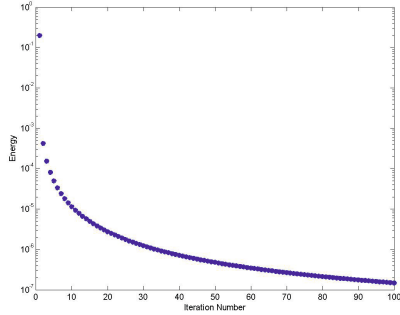
The restored images, and the differences between the solutions by proposed DDMs and the exact minimizer are shown in Figure 4 at iteration 103. The proposed DDMs (Figure 4 (c)) perform as good as the gradient projection method (Figure 4 (b)). Inferred from Figure 4, the denoising results are good enough within 3 iterations (Figure 4 (d)-(f)) of the proposed DDMs. Furthermore in Figure 4 (g), the differences located at the boundary of the subdomains are much bigger than that anywhere else after the first iteration. As the iteration goes on, the differences near the boundary of subdomains become small (Figure 4 (h)-(j)). Indeed, the proposed DDMs converge by observing Figure 3 (b), where we show the curve of the energy decay.

### 3.2 Convergence v.s. Overlapping Size $\delta$ , subsize and Relaxed Parameter $\alpha$

First, we test how the convergence performance relies on the overlapping size  $\delta$  by setting  $\delta = 4, 8, 16$ , and 32. By observing the results in Figure 5 (a), the DDMs



(a)



(b)

**Fig. 3.** (a) Cameraman with resolution  $256 \times 256$ ; (b) Energy decay v.s. iteration number, with  $\alpha = 1$ ,  $\tau = \frac{1}{4}$ ,  $subsize = 128$ ,  $m_T = 4$ ,  $\delta = 4$ ,  $\sigma = 50$ ,  $\lambda = 80$ ,  $N_{in} = 500$

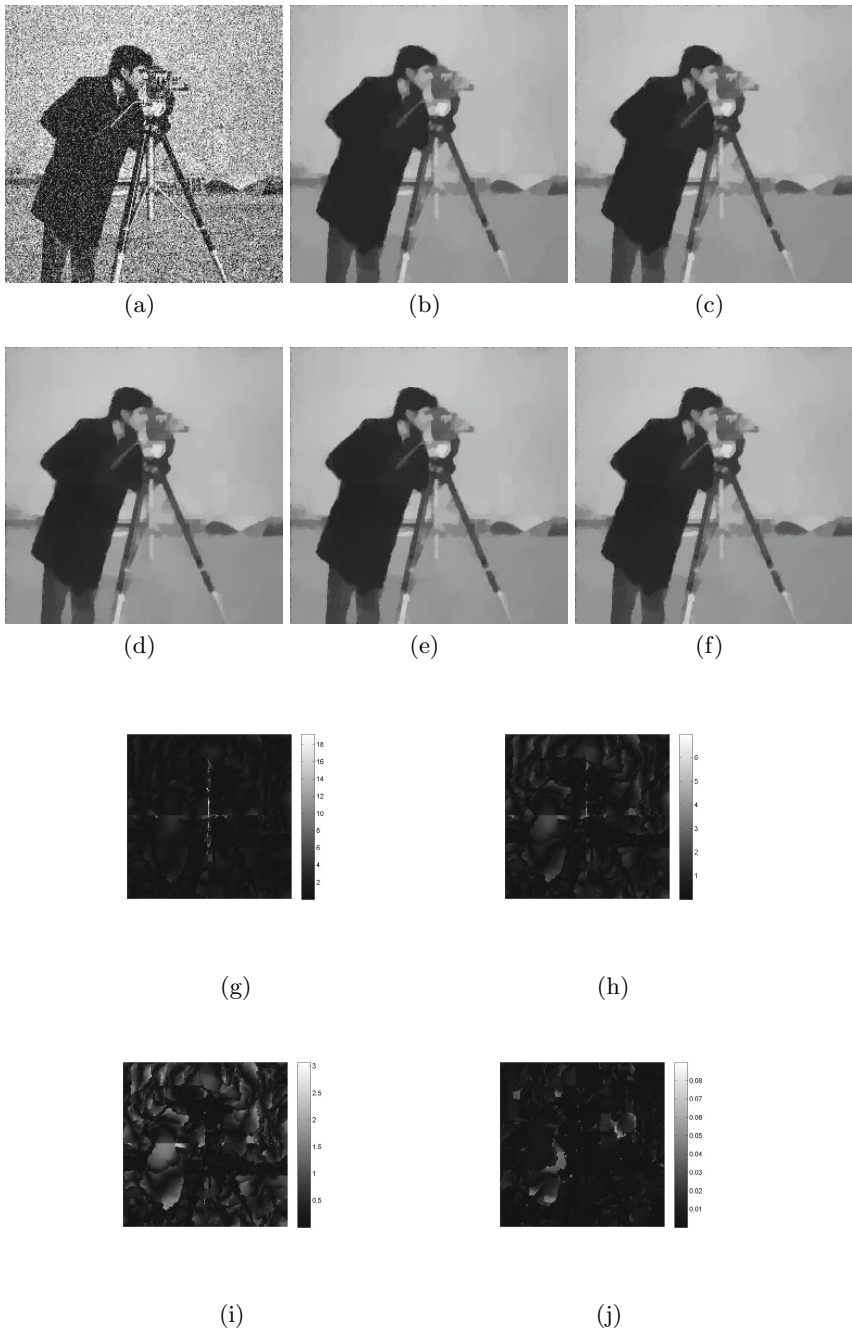
converge fast when overlapping size becomes large. Second, we test how the convergence rate relies on the number of subdomains by setting  $subsize = 8, 16, 32$  and  $64$  (number of blocks  $m_T = 32 \times 32, 16 \times 16, 8 \times 8$ , and  $4 \times 4$ ). The results are shown in Figure 5 (b). The convergence becomes fast when the size of subdomain becomes small. Since we use coloring techniques that fixes  $M = 4$ , the smaller size of the subdomain, the relative larger is the overlapping size. Third, we test our DDMs with respect to the relaxed parameter  $\alpha$  by setting  $\alpha = 1/8, 1/4, 1/2$  and  $1$ , and see the results in Figure 5 (c). The DDMs converge faster as  $\alpha$  is much closer to  $1$ , and one shall fix  $\alpha$  to be  $1$  in order to achieve better performance in real computation.

### 3.3 Sensitivity to the Regularized Parameter

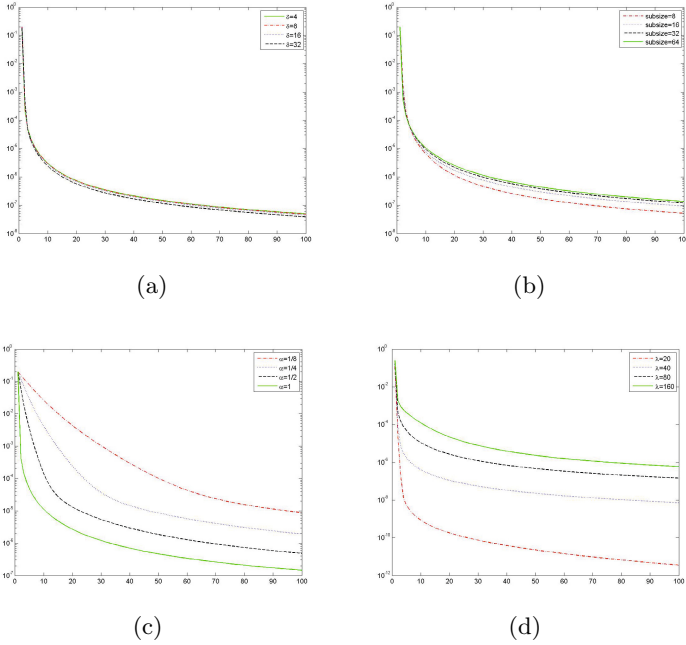
We test the performance of the proposed DDMs with respect to the regularized parameter  $\lambda$ . The energy decay is shown in Figure 5(d), that implies that our proposed DDMs are quite sensitive to the parameter  $\lambda$ . Therefore, the two-level method with additional coarse mesh shall be considered to accelerate the proposed DDMs.

### 3.4 Performance of Parallel Implementation

At last, we realize Algorithm II in parallel. Coloring technique is adopted as Figure 1, and the subproblems defined on the subdomains with the same color can be computed in parallel. The algorithm coded by C(OpenMP) runs on the workstation (Dell Precision-WorkStation-T7500) with Intel (R) Xeon(R) CPU X5650 2.67GHz $\times$ 2(12 cores), and 10G Ram. Each subproblem with the same color is computed using one single core. Therefore, there are at most 12 subproblems are computed in parallel at the same time. We consider the image “Lena” with



**Fig. 4.**  $\alpha = 1, \tau = \frac{1}{4}, \text{subsize} = 128, m_T = 4, \delta = 4, \sigma = 50, \lambda = 80, N_{in} = 500$ . Noised Image (a); Denoised image without DDMs (b) and denoised by proposed DDMs(c); The denoised images within first 3 iterations in (d)-(f); Errors between the iterative results(within the first 3 iterations) and the exact solution in (g)-(i); Errors between the solutions by DDMs after 100 iterations and the exact solution in (j).



**Fig. 5.** (a) Convergence v.s. overlapping size  $\delta$ ; (b) Convergence v.s. *subsize*(or number of blocks  $m_T$ ); (c) Convergence v.s. relaxed parameter  $\alpha$ ; (d) Energy decay with respect to the regularized parameter  $\lambda$ .

the resolution  $2048 \times 2048$ , and the noise level  $\sigma = 50$ . The parameters  $\lambda = 60$ , and  $\alpha = 1$ . The algorithm stops after  $n = 10$  outer iterations, while  $N_{in} = 500$ . For the domain decomposition, we set *subsize* = 16, and overlapping size  $\delta = 4$ . That is to say, there are  $m_T = 128 \times 128$  blocks. At most  $m_T/4$  subproblems can be computed independently, that also depends on the cores of the workstation.

We present the time table by using different cores in Table 1. Meanwhile, the speed-up ratio and speed-up efficiency are adopted to measure the performance of the parallel computing. The speed-up ratio is computed by the ratio of the elapsed time by multiple cores to the time by one single core, and the speed-up efficiency is the ratio of the speed-up ratio to the number of cores. Obviously, the speed-up ratio can not exceed the maximum number of cores, and the efficiency can not exceed 1. The bigger the values are, the better is the proposed algorithm.

**Table 1.** Parallel test: \* denotes blank

Number of Cores	1	2	4	8	12
Elapsed Time	1074	540	273	146	102
Speed-up Ratio	*	1.99	3.93	7.35	10.43
Speed-up Efficiency	*	0.99	0.98	0.92	0.88

Inferred from the table, the better speed-up ratio and efficiency are obtained. Especially, the speed-up efficiency is not less than 0.88. Therefore, our proposed algorithm is quite suitable for parallel computing.

## 4 Conclusion

We have proposed the efficient one-level DDMs for the dual formulation of ROF model, and the convergence rates are deduced as well. Extensive numerical experiments demonstrate the efficiency of the DDMs. In the future, the two-level corrections and multigrid methods should be adopted to accelerate and increase the robustness of the DDMs. The dual formulation based model with more applications in image processing should be considered as well.

**Acknowledgements.** The first author H. Chang was supported by PHD Programme 52XB1304 of Tianjin Normal University. The last author D. Yang was supported by Natural Science Foundation of China, Grant No. 11071080.

## References

1. Acar, R., Vogel, C.: Analysis of Bounded Variation Penalty Methods for Ill-Posed Problems. *Inverse Probl.*, 10,1217-1230(1994)
2. Appleton, B., Talbot, H.: Globally Optimal Geodesic Active Contours. *J. Math. Imaging Vis.*, 23, 67-86(2005)
3. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in Linear and Non-linear Programming. With contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford, Calif.(1958)
4. 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)
5. Chambolle, A.: An Algorithm for Total Variation Minimization and Applications. *Math. Imaging Vis.*, 20,89-97(2004)
6. Chang, H., Zhang, X., Tai, X.C., Yang, D.: Domain Decomposition Methods for Nonlocal Total Variation Image Restoration. *J. Sci. Comput.*, 60,79-100(2014)
7. Chen, K., Tai, X.C.: On Semismooth Newton's Methods for Total Variation Minimization. *J. Sci. Comput.*, 33, 115-138(2007)
8. Dong, Y., Hintermüller, M., Neri, M.: An Efficient Primal-Dual Method for  $L^1$ -TV Image Restoration. *SIAM J. Imaging Sci.*, 2, 1168-1189(2009)
9. Duan, Y., Tai, X. C.: Domain Decomposition Methods with Graph Cuts Algorithms for Total Variation Minimization. *Adv. Comput. Math.*, 36, 175-199(2012)
10. Fornasier, M., Langer, A., Schönlieb, C.: Domain Decomposition Methods for Compressed Sensing. ArXiv preprint, arXiv:0902.0124(2009)
11. Fornasier, M., Langer, A., Schönlieb, C.: A Convergent Overlapping Domain Decomposition Method for Total Variation Minimization. *Numer. Math.*, 116, 645-685(2010)
12. Fornasier, M., Schönlieb, C.: Subspace Correction Methods for Total Variation and  $L_1$ - Minimization. *SIAM J. Numer. Anal.*, 47, 3397-3428(2009)

13. Firsov, D., Lui, S. H.: Domain Decomposition Methods in Image Denoising Using Gaussian Curvature. *J. Comput. Appl. Math.*, 193, 460-473(2006)
14. Goldstein, T., Osher, S.: The Split Bregman Method for L1-Regularized Problems. *SIAM J. Imaging Sci.*, 2, 323-343(2009)
15. Glowinski, R., Tallec, P.: Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia(1989)
16. Hintermüller, M., Kunisch, K.: Total Bounded Variation Regularization as a Bilaterally Constrained Optimaization Problem. *SIAM. J. Appl. Math.*, 64, 1311-1333(2004)
17. Hintermüller, M., Langer, A.: Subspace Correction Methods for a Class of Non-smooth and Nonadditive Convex Variational Problems with Mixed  $L^1/L^2$  Data-Fidelity in Image Processing. *SIAM J. Imaging Sci.*, 6, 2134-2173(2013)
18. Hintermüller, M., Langer, A.: Non-overlapping Domain Decomposition Methods for Dual Total Variation Based Image Denoising. *J. Sci. Comput.*, DOI 10.1007/s10915-014-9863-8(2014)
19. Marquina, A., Osher, S.: Explicit Algorithms for A New Time Dependent Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal. *SIAM J. Sci. Comput.*, 22, 387-405(2000)
20. Müller, J.: Parallel Total Variation Minimization. University of Muenster, Diploma Thesis(2008)
21. Ng, M., Qi, L., Yang, Y., Huang, Y.: On Semismooth Newton's Methods for Total Variation Minimization. *J. Math. Imaging Vis.*, 27, 265-276(2007)
22. Rudin, L., Osher, S., Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D*, 60, 259-268(1992)
23. Scherzer, O.(editor): Handbook of Mathematical Methods in Imaging. Springer, New York(2011)
24. Tai, X.C.: Rate of Convergence for Some Constraint Decomposition Methods for Nonlinear Variational Inequalities. *Numer. Math.*, 93, 755-786(2003)
25. Tai, X.C., Espedal, M.: Applications of a Space Decomposition Method to Linear and Nonlinear Elliptic Problems. *Numer. Meth. Part. D. E.*, 14, 717-737(1998)
26. Tai, X. C., Espedal, M.: Rate of Convergence of Some Space Decomposition Methods for Linear and Nonlinear Problems. *SIAM J. Numer. Anal.*, 35, 1558-1570(1998)
27. Tai, X. C., Xu, J.: Global and Uniform Convergence of Subspace Correction Methods for Some Convex Optimization Problems. *Math. Comput.*, 71, 105-124(2002)
28. Tai, X. C., Duan, Y.P.: Domain Decomposition Methods with Graph Cuts Algorithms for Image Segmentation. *Int. J. Numer. Anal. Model.*, 8, 137-155(2011)
29. Toselli, A., Widlund, O.: Domain Decomposition Methods-Algorithm and Theory. Springer-Verlag Berlin Heidelberg (2005)
30. Vogel, C.: A Multigrid Method for Total Variation-Based Image Denoising. *Computation and Control IV, Progress in Systems and Control Theory*, 20, Birkhäuser Boston(1995)
31. Vogel, C.: Computational Methods for Inverse Problems. SIAM(2002)
32. Vogel, C., Oman, M.: Iterative Methods for Total Variation Denoising. *SIAM J. Sci. Comput.*, 17, 227-238(1996)
33. Vogel, C., Oman, M.: Fast, Robust Total Variation-Based Reconstruction of Noisy, Blurred Images. *IEEE Trans. Image Process.*, 7, 813-824(1998)
34. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A New Alternating Minimization Algorithm for Total Variation Image Reconstruction. *SIAM J. Imaging Sci.*, 1, 248-272(2008)

35. Wang, Y., Yin, W., Zhang, Y.: A Fast Algorithm for Image Deblurring with Total Variation Regularization. Rice University CAAM Technical Report TR07-10(2007)
36. 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, 300-339(2010)
37. Xu, J., Tai, X. C., Wang, L.L.: A Two-Level Domain Decomposition Method for Image Restoration. *Inverse Probl. Image*, 4, 523-545(2010)
38. Xu, J., Chang, H., Qin, J.: Domain Decomposition Method for Image Deblurring. *J. Comput. Appl. Math.*, 271, 401-414(2014)
39. Zhu, M., Chan, T.: An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration. UCLA, Center for Applied Math., CAM Reports No. 08-34(2008)

# A Convex Solution to Disparity Estimation from Light Fields via the Primal-Dual Method

Mahdad Hosseini Kamal<sup>1</sup>, Paolo Favaro<sup>2</sup>, and Pierre Vandergheynst<sup>1</sup>

<sup>1</sup> Ecole Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland  
{mahdad.hosseini.kamal,pierre.vandergheynst}@epfl.ch

<sup>2</sup> University of Bern, Bern 3012, Switzerland  
paolo.favaro@iam.unibe.ch

**Abstract.** We present a novel approach to the reconstruction of depth from light field data. Our method uses dictionary representations and group sparsity constraints to derive a convex formulation. Although our solution results in an increase of the problem dimensionality, we keep numerical complexity at bay by restricting the space of solutions and by exploiting an efficient Primal-Dual formulation. Comparisons with state of the art techniques, on both synthetic and real data, show promising performances.

**Keywords:** Light fields, multi-view stereo, primal-dual formulation.

## 1 Introduction

The estimation of a disparity map from multiple images is one of the very well studied problems in computer vision. Some of the most dramatic improvements in this field occurred with the introduction of novel numerical frameworks and their corresponding theory. A non-exhaustive list of such breakthroughs are the early work on space carving [20], the level set formulation and the variational framework [10], the Markov random field framework with polynomial-complexity solvers [6], the  $L_1$ -Total Variation optimization framework [35] and, more recently, convex formulations that aim for global optimality [25]. In this paper, we look at a novel approach based on recent primal-dual optimization techniques. Our approach is also convex as in the most recent developments, but we work with discrete labels (the possible disparity values).

Our formulation is based on a linear model of the data where a patch in an image is written as a linear combination of patches in other views. The key idea is that ideal Lambertian objects generate views that look alike (modulo foreshortening) and therefore corresponding patches live approximately on a 1D manifold. When objects are not Lambertian, they generate effects, such as specularities, that change with the pose of the camera. One can notice, however, that these effects are typically rare (*i.e.*, they happen only on some of the views) and spatially local. Hence, a natural way to model image patches of non Lambertian objects is by using an additive model where one of the two factors is sparse and the other is low-rank. If a finite set of possible depth candidates for a patch is



available, one can then verify which hypothesis best fits the low-rank + sparse model. Our strategy is therefore a competition between the different disparity hypotheses. We essentially allow the data to be explained by a simultaneous linear combination of **all** low-rank + sparse models. However, we force coefficients to focus on only a few of the models (where each model corresponds to a single disparity hypothesis) via group-sparsity penalty terms. We expect that coefficients be mostly non zero at the true disparity as this is the case that gives the fit with the sparsest set of outliers. Notice that the individual coefficients of each linear combination are not important, and indeed, typically, infinite solutions might be possible especially at the correct disparity. However, as long as coefficients have most non zero values at only one group, we can still correctly identify the disparity.

While this approach seems straightforward, in practice it faces considerable dimensionality challenges because data is replicated several times due to the patch-based model and the number of disparity hypotheses. This makes operations such as matrix inversion, often encountered in optimization schemes, impossible to carry out. To address these challenges we propose a primal-dual approach that results in simple element-wise thresholding operations and 2 (global) matrix multiplications at each step.

**Contributions:** We propose a framework to address the disparity estimation problem of light fields. In particular, we make the following contributions:

- We present a novel model for light field disparity estimation to represent a light field image patch as a linear combination of other light field patches. This representation satisfies a group sparse model and depends only on a group of light field patches of the same disparity.
- Occlusions are handled uniformly in our framework as a sparse component and this brings more robustness than in traditional matching methods.
- We introduce a robust and globally optimal solution for light field patch matching based on a preconditioned primal-dual algorithm [24], which allows to match a light field patch in all the views to estimate the disparity map.

## 2 Related Work

**Light Field Disparity Estimation:** Light fields can be captured using a camera array [30] or lenslet arrays [23] or as a sequence of images. One of the first approaches to compute light field depth exploits linear structures in light fields through a line fitting algorithm [5]. Other methods use more traditional stereo reconstruction techniques to match the corresponding pixels in light field images, such as block-matching techniques [4] or clustering methods to identify similar pixel matches [3, 11]. Ziegler et al. [36] proposed a Fourier-based technique to compute depth values. To achieve higher global coherence, light field depth estimation methods employ a global cost function to impose smoothness on the estimated depth values [8, 19, 32]. A limitation common to all these methods is that they optimize a global cost function that is not convex. Therefore, the estimated depth map depends on the initial input. Moreover, fine details are lost

because a coarse-to-fine multi-resolution technique is often used to avoid ending in weak local minima. Our approach overcomes these limitations by introducing a convex formulation.

**Multiview Stereo Methods:** Multiview techniques require detecting and handling outliers [2, 16]. The difficulty of outlier modeling is due to the unstructured nature of errors produced by outliers. However, these errors can only influence a small part of the image and are therefore sparse in a canonical basis [2, 33]. An alternative to explicit occlusion modeling is to match only reliable pixels and fill the unmatched correspondences via regularization [18, 27]. However, as explained in [28], these methods are prone to artifacts. Multiview stereo methods employ a large number of images [13, 17] to compute the full geometry of a scene and often yield a smooth geometry. Our light field disparity estimation yields a representation that falls in the middle: it is more complete than in stereo techniques, but less than in multiview stereo.

**Sparse Representation:** The similarity of image structures in a dataset is used in data clustering [9, 22] to determine the low-dimensional subspace of high dimensional data. Many schemes exploit data similarity to represent image correspondences in a dataset [21, 33]. In contrast to these clustering techniques, our proposed disparity estimation scheme looks for the best representation of each patch within a set of clusters. The clusters are generated from a number of disparity hypotheses, such that the members of a cluster are either chosen or discarded together. To achieve this we introduce a coupling term between the coefficients via group sparsity.

In this paper, we estimate disparity from light fields by representing patches of a desired light field view with an overcomplete dictionary. The elements of the dictionary are patches of other views reprojected back onto a reference view for a given set of disparity candidates. If sufficiently many patch samples are available, patches of the reference view can be written as a linear combination of patches from the correct disparity hypothesis. This representation is naturally group sparse, since only a single disparity candidate of the dictionary can be assigned to a given patch. This representation can be recovered efficiently via group sparsity minimization [34].

### 3 Multiple Views and Light Fields

We consider capturing several images of the same static scene by translating a camera on the  $x - y$  plane, where  $z$  is aligned to the camera optical axis, or, equivalently, by employing a camera array, or a plenoptic camera, where all the camera sensors lie on the same plane. More in general, we can describe the captured data as a 4D light field  $L : \Omega \times \Theta \mapsto [0, +\infty)$  where  $\Omega \equiv \mathbb{R}^{N \times M}$  denotes the spatial domain (the pixel coordinates within each image) and  $\Theta$  the angular domain (the camera center coordinates). We consider cameras arranged in a regular lattice and denote with  $\Delta = [\Delta_x \ \Delta_y]^T \in \mathbb{R}^2$  the displacement between a camera and its north-west neighbor. Then, we define  $\Theta = \{[\Delta_x i \ \Delta_y j]^T | i = 1 \dots n, j = 1 \dots m\}$  as the 3D camera center of the  $(i, j)$ -th camera is located at

$[\Delta_x i \ \Delta_y j \ 0]^T$ . For simplicity, we use the notation  $L_{i,j}(x, y)$  to denote  $L(x, y, i, j)$ . A visible plane in the scene, parallel to the images planes of the cameras, will generate images in the light field  $L$  that are related to each other by a shift or *disparity*  $\rho : \Omega \mapsto [0, +\infty)$ , for simplicity we denote  $\rho(x, y)$  by  $\rho$ . In formulas, this can be written as

$$L_{i,j}(x, y) = L_{p,q}(x - \rho\Delta_x(p - i), y - \rho\Delta_y(q - j)) \quad (1)$$

for all  $(x, y)$  that fall within the spatial domain of both light field views and for all  $(i, j)$  and  $(p, q)$  camera pairs.

A common approach to estimating the disparity  $\rho$  is then to pose a variational problem of the form

$$\min_{\rho} \sum_{\substack{i,j,p>i \\ q>j,x,y}} \Phi(L_{i,j}(x, y) - L_{p,q}(x - \rho(p - i)\Delta_x, y - \rho(q - j)\Delta_y)) + \Gamma(\rho), \quad (2)$$

where  $\Phi$  is some robust penalty term for departures from zero and  $\Gamma$  is a regularization term for the unknown disparity  $\rho$  such as total variation. This problem is non convex and therefore finding the global optimum is a very challenging task. While good solutions have been obtained for the above problem, recent efforts have produced convex variational formulations [12, 25] with high-quality disparity reconstructions. Both of these methods work with continuous representations. However, one of the key differences between these two methods is that, while [25] achieves convexity by increasing the problem dimensionality, [12] achieves convexity by fixing the structure tensor with some initial approximate disparity estimate. Our method follows the strategy of the first approach and also results in a high-dimensional representation. However, we do not rely on any initial estimate (although it might considerably speed up the convergence). Moreover, as we describe in the next sections, our convex formulation is entirely in the discrete domain and exploits the quantization of the disparity values.

## 4 A Patch-Based Image Formation Model

Our first step is to rewrite the problem (2) as a patch matching problem. Let us define the *patch operator*  $\mathcal{P}_{x,y}$  as the mapping that extracts the  $W \times W$  patch whose top-left corner lies at  $(x, y)$  of an image  $I$ , *i.e.*,

$$\mathcal{P}_{x,y}(I) = \{I(x + x_0, y + y_0)\}_{x_0,y_0=0,\dots,W-1}. \quad (3)$$

We define the output of the patch operator to be a patch rearranged as a column vector whose  $W^2$  elements have been rearranged in lexicographical order. Consider extracting one patch from each view of a light field, except for the  $(i_0, j_0)$ -th one (for example, this could be the central view), given a disparity  $\rho$  and collecting all the patches in a matrix  $Q_{x,y}^\rho \in \mathbb{R}^{W^2 \times (nm-1)}$ . This operation can be described via

$$Q_{x,y}^\rho = \{\mathcal{P}_{x-\rho\Delta_x(p-i_0), y-\rho\Delta_y(q-j_0)}(L_{p,q}) : \forall(p, q) \neq (i_0, j_0)\}. \quad (4)$$

If  $\rho$  is the true disparity of a fronto-parallel object in space, then all the columns in  $Q_{x,y}^\rho$  will be identical to each other (in the ideal Lambertian case) and identical to the column vector  $\mathcal{P}_{x,y}(L_{i_0,j_0})$ . We also denote the latter vector with the symbol  $Y_{x,y}$ . More in general however, noise, non Lambertianity, shadows, occlusions, inter reflections and so on need to be taken into account. Since we believe that most of the time the Lambertian approximation will hold, we consider all the other image distortions as infrequent and use a sparse representation to model them, *i.e.*,

$$Y_{x,y} = Q_{x,y}^\rho C_{x,y}^\rho + E_{x,y} \quad (5)$$

where  $C_{x,y}^\rho$  is a  $nm - 1$  column vector and  $E_{x,y}$  is a  $W^2$  column vector with few nonzero entries. The coefficients in  $C_{x,y}^\rho$  determine the linear combination of vectors in  $Q_{x,y}^\rho$  that generate  $Y_{x,y}$ . When the disparity  $\rho$  corresponds to the true solution, any  $C_{x,y}^\rho$  such that  $\mathbf{1}^T C_{x,y}^\rho = 1$  will satisfy the above equation. Vice versa, when the disparity is incorrect and the scene has sufficiently rich texture, there should not exist any vector  $C_{x,y}^\rho$  that satisfies (5). Thus, we propose to force the disparity  $\rho$  to take values only from the set  $\{\rho_1, \rho_2, \dots, \rho_D\}$  and extend (5) to

$$Y_{x,y} = [Q_{x,y}^{\rho_1} \ Q_{x,y}^{\rho_2} \ \dots \ Q_{x,y}^{\rho_D}] [C_{x,y}^{\rho_1} \ C_{x,y}^{\rho_2} \ \dots \ C_{x,y}^{\rho_D}]^T + E_{x,y} \doteq Q_{x,y} C_{x,y} + E_{x,y} \quad (6)$$

where the  $W^2 \times (nm - 1)D$  matrix  $Q_{x,y}$  and the  $(nm - 1)D$  vector  $C_{x,y}$  are implicitly defined by the equation to the right.

## 5 Depth Estimation

Based on the model (5), a first formulation for estimating disparity through patch matching is

$$\min_{C,E} \frac{1}{2} \sum_{x,y} \|Y_{x,y} - Q_{x,y} C_{x,y} - E_{x,y}\|_2^2 + \mu \|E_{x,y}\|_1 \quad (7)$$

where  $\mu > 0$  is a constant determining the degree of sparsity of  $E_{x,y}$ ,  $\|E_{x,y}\|_1$  denotes the  $\ell^1$  norm of  $E_{x,y}$ , and  $C$  and  $E$  are the column vectors obtained by stacking vertically all the vectors  $C_{x,y}$  and  $E_{x,y}$  respectively. Since the total number of patches within the image domain is  $\tilde{M}\tilde{N}$ , where  $\tilde{M} = M - W + 1$  and  $\tilde{N} = N - W + 1$ , the  $E$  vector has  $\tilde{M}\tilde{N}W^2$  elements and the  $C$  vector has  $\tilde{M}\tilde{N}(nm - 1)D$  elements.

As explained in the previous section, we aim at concentrating the coefficients of  $C_{x,y}$  on the patches belonging to just one disparity hypothesis. If this is the case, then, given  $C_{x,y}$ , one can estimate the disparity at a pixel  $(x, y)$  by using

$$\hat{\rho} = \operatorname{argmax}_{\rho \in \{\rho_1, \dots, \rho_D\}} \|C_{x,y}^\rho\|. \quad (8)$$

The same problem can be written in the following compact form

$$\min_{C,E} \frac{1}{2} \|Y - QC - E\|_2^2 + \mu \|E\|_1 \tag{9}$$

where the column vector  $Y$  has been obtained by stacking all the  $Y_{x,y}$ , and  $Q$  is a block diagonal matrix whose blocks are the matrices  $Q_{x,y}$ . To encourage the concentration of nonzero entries in a single disparity block of  $C_{x,y}$  we propose to minimize the mixed  $\ell_1/\ell_2$  norm of  $C_{x,y}$ , which is defined as  $\|C\|_{1,2} \doteq \sum_{x,y} \sum_{k=1,\dots,D} \|C_{x,y}^{\rho k}\|_2$ . Finally, since the disparity is a smooth map, we add a vector-valued isotropic total variation (TV) regularization term

$$\|\nabla C\|_{1,2} \doteq \sum_{x,y} \sqrt{\|C_{x,y} - C_{x+1,y}\|_2^2 + \|C_{x,y} - C_{x,y+1}\|_2^2} \tag{10}$$

where  $\nabla$  denotes the finite gradient in the spatial domain (and can be written in matrix form). By minimizing this term we encourage  $C$  coefficients to be similar across the spatial domain. The complete minimization problem can be written as follows

$$\min_{C,E} \frac{1}{2} \|Y - QC - E\|_2^2 + \mu \|E\|_1 + \lambda \|\nabla C\|_{1,2} + \gamma \|C\|_{1,2} \tag{11}$$

where  $\lambda, \gamma > 0$  are two constants. This is a convex problem and therefore it has the desirable property of converging to the same global optimum given any initialization. The minimization of problem (11) presents several challenges due to its high dimensionality, which we address in the next section.

## 6 Primal-Dual Formulation

One immediate issue of a primal solver for problem (11) is that it requires inverting very large matrices that are not easily diagonalized. To avoid such computational difficulties, we consider the primal-dual method, which is a first order algorithm, it does not require matrix inversions and enjoys fast convergence rates [25].

Firstly, we rewrite problem (11) in a more compact way by combining all the unknowns  $C$  and  $E$  into a single variable  $X$ , and by defining 3 new functions  $F_1$ ,  $F_2$ , and  $F_3$  as follows

$$F_1(AX - Y) \doteq \frac{1}{2} \|Y - QC - E\|_2^2 \tag{12}$$

$$F_2(\Pi_E X) \doteq \|E\|_1 \tag{13}$$

$$F_3(BX) \doteq \|\nabla C\|_{1,2} + \frac{\gamma}{\lambda} \|C\|_{1,2} \tag{14}$$

where  $A \doteq [Q I_d]$ , with  $I_d$  the identity matrix,  $\Pi_E X \doteq E$  and  $B \doteq [\nabla^T \frac{\gamma}{\lambda} I_d]^T \Pi_C$ , with  $\Pi_C X \doteq C$ . Notice that all the above functions are convex in the variable  $X$ . Then, our primal formulation becomes

$$\min_X F_1(AX - Y) + \mu F_2(\Pi_E X) + \lambda F_3(BX). \tag{15}$$

To solve the primal problem we can compute the gradients of the cost function and set it to zero. An immediate observation is that the gradient will yield in the best case linear systems with non-diagonal matrices. For example, the first term  $F_1(AX - Y)$  yields

$$\frac{\partial}{\partial X} F_1(AX - Y) = A^T AX - A^T Y \tag{16}$$

which requires dealing with the matrix  $A^T A$ . To avoid that, we use the primal-dual method. This method is based on the Legendre-Fenchel (LF) transform. Given a function  $F$ , the LF transform yields a conjugate function  $F^*$  such that

$$F^*(Z) \doteq \sup_X \langle X, Z \rangle - F(X). \tag{17}$$

The conjugate function  $F^*$  is by construction convex and when  $F$  is also convex, then the LF transform  $F^{**}$  of the conjugate  $F^*$  is again  $F$ . When the conjugate functions  $F_1^*$ ,  $F_2^*$ , and  $F_3^*$  can be computed easily and possibly in closed-form, then it is convenient to consider the primal-dual problem

$$\begin{aligned} \min_X \max_{Z_1, Z_2, Z_3} & \langle AX - Y, Z_1 \rangle - F_1^*(Z_1) + \mu \langle \Pi_E X, Z_2 \rangle - \mu F_2^*(Z_2) \\ & + \lambda \langle BX, Z_3 \rangle - \lambda F_3^*(Z_3). \end{aligned} \tag{18}$$

which we write in more compact form as

$$\min_X \max_Z \langle KX, Z \rangle - \hat{F}(Z) \tag{19}$$

where  $K \doteq [A^T \ \mu \Pi_E^T \ \lambda B^T]^T$ ,  $Z \doteq [Z_1^T \ Z_2^T \ Z_3^T]^T$ , and  $\hat{F}(Z) \doteq F_1^*(Z_1) + \mu F_2^*(Z_2) + \lambda F_3^*(Z_3)$ . To solve the above saddle point problem, we need to define the *proximity operator*, which is our fundamental computational tool to deal with the conjugate functions.

### 6.1 Proximity Operator

A proximity operator  $\text{prox}_{\sigma F}$ , with  $\sigma > 0$ , takes as input a convex and lower semicontinuous function  $F$  and maps it to the following function

$$\text{prox}_{\sigma F}(Z) = \underset{X}{\operatorname{argmin}} \frac{1}{2} \|Z - X\|_2^2 + \sigma F(X), \quad \forall Z, \tag{20}$$

see for more information the review paper [7]. The main result that we will exploit here is Moreau’s identity. Given the conjugate  $F^*$  of  $F$  we have that

$$\text{prox}_{\sigma F^*}(Z) = Z - \sigma \text{prox}_{F/\sigma}(Z/\sigma) \tag{21}$$

and hence we can compute the proximity operator of the conjugate function  $F^*$  directly by using the proximity operator of the function  $F$ .

### 6.2 Primal-Dual Algorithm

The primal-dual algorithm to solve problem (19) is

$$\begin{aligned}
 Z_1^{n+1} &= \text{prox}_{\sigma F_1^*}(Z_1^n + \sigma(A\bar{X}^n - Y)) \\
 Z_2^{n+1} &= \text{prox}_{\sigma\mu F_2^*}(Z_2^n + \sigma\mu\Pi_E\bar{X}^n) \\
 Z_3^{n+1} &= \text{prox}_{\sigma\lambda F_3^*}(Z_3^n + \sigma\lambda B\bar{X}^n) \\
 X^{n+1} &= X^n - \tau K^T Z^{n+1} \\
 \bar{X}^{n+1} &= X^{n+1} + \theta(X^{n+1} - X^n)
 \end{aligned}
 \tag{22}$$

where  $n$  is the iteration index,  $\theta \in (0, 1]$  and  $\tau\sigma\|K\|^2 < 1$ . While the bottom two iterations are straightforward, the first one on the dual variable  $Z$  requires computing the proximity operator of the conjugate functions  $F_1^*$ ,  $F_2^*$ , and  $F_3^*$ . The first two functions are relatively easy to obtain as the conjugate functions can be computed in closed-form

$$F_1^*(Z_1) = \frac{1}{2}\|Z_1\|_2^2, \quad \{F_2^*(Z_2)\}_s = \begin{cases} 0 & \text{if } |\{Z_2\}_s| \leq \mu \\ +\infty & \text{otherwise} \end{cases}
 \tag{23}$$

where  $s = 1, \dots, \tilde{M}\tilde{N}W^2$ . Hence, we can readily obtain the first two steps of the primal-dual algorithm

$$Z_1^{n+1} = \frac{1}{\sigma + 1}(Z_1^n + \sigma(A\bar{X}^n - Y)), \quad \{Z_2^{n+1}\}_s = \mathcal{H}_{\sigma\mu} \left( \left\{ \frac{Z_2^n}{\sigma\mu} + \Pi_E\bar{X}^n \right\}_s \right)
 \tag{24}$$

where  $s = 1, \dots, \tilde{M}\tilde{N}W^2$  and  $\mathcal{H}_{\sigma\mu}$  denotes the element-wise thresholding operator

$$\mathcal{H}_{\sigma\mu}(z) \doteq \min\{\sigma\mu, |z|\} \text{sign}(z).
 \tag{25}$$

The last term  $F_3^*$  is more involved. We compute the update equation by exploiting Moreau’s identity

$$\text{prox}_{\sigma\lambda F_3^*}(Z_3^n + \sigma\lambda B\bar{X}^n) = Z_3^n + \sigma\lambda B\bar{X}^n - \sigma\lambda \text{prox}_{F_3/(\sigma\lambda)}(Z_3^n/(\sigma\lambda) + B\bar{X}^n)
 \tag{26}$$

so that we only need to compute  $\text{prox}_{F_3/(\sigma\lambda)}$ . Notice that  $F_3(Z_3)$  is the  $\ell_1/\ell_2$  norm  $\|Z_3\|_{1,2}$ . Thus, we need to evaluate

$$\text{prox}_{F_3/(\sigma\lambda)}(Z_3^n/(\sigma\lambda) + B\bar{X}^n) = \underset{Z}{\text{argmin}} \frac{1}{2} \left\| \frac{1}{\sigma\lambda} Z_3^n + B\bar{X}^n - Z \right\|_2^2 + \frac{1}{\sigma\lambda} \|Z\|_{1,2}.
 \tag{27}$$

The solution is computed in closed-form and results in a block soft-thresholding

$$\text{prox}_{F_3/(\sigma\lambda)}(Z_3^n/(\sigma\lambda) + B\bar{X}^n) = \mathcal{S}_{1/(\sigma\lambda)}\left(\frac{1}{\sigma\lambda}Z_3^n + B\bar{X}^n\right) \quad (28)$$

with

$$\{\mathcal{S}_{1/(\sigma\lambda)}(Z_3)\}_b = \{Z_3\}_b \max\left\{0, 1 - \frac{1}{\sigma\lambda\|\{Z_3\}_b\|_2}\right\} \quad (29)$$

and where blocks are indexed by  $b = 1, \dots, (3\tilde{M}\tilde{N} - \tilde{M} - \tilde{N})D$ , since  $Z_3$  is a  $(3\tilde{M}\tilde{N} - \tilde{M} - \tilde{N})D(nm - 1)$  dimensional vector.<sup>1</sup> Finally, by plugging the last expression in the proximity operator of  $F_3^*$ , the last update equation becomes

$$\{\text{prox}_{\sigma\lambda F_3^*}(Z_3^n + \sigma\lambda B\bar{X}^n)\}_b = \{Z_3^n + \sigma\lambda B\bar{X}^n\}_b \cdot \left(1 - \max\left\{0, 1 - \frac{1}{\|\{Z_3^n + \sigma\lambda B\bar{X}^n\}_b\|_2}\right\}\right) \quad (30)$$

where  $b = 1, \dots, (3\tilde{M}\tilde{N} - \tilde{M} - \tilde{N})D$ .

In all update equations there are no matrix inversions and calculations are therefore highly parallelizable. The final algorithm is summarized in Table 1.

**Table 1.** Primal-dual algorithm for disparity estimation from light field data. Notice that  $Z_1 \in \mathbb{R}^{\tilde{M}\tilde{N}W^2 \times 1}$ ,  $Z_2 \in \mathbb{R}^{\tilde{M}\tilde{N}W^2 \times 1}$ , and  $Z_3 \in \mathbb{R}^{(3\tilde{M}\tilde{N} - \tilde{M} - \tilde{N})D(nm-1) \times 1}$ .

$Z_1^{n+1} = (Z_1^n + \sigma(A\bar{X}^n - Y))/(\sigma + 1)$ $\{Z_2^{n+1}\}_s = \mathcal{H}_{\sigma\mu}(\{Z_2^n/(\mu\sigma) + \Pi_E\bar{X}^n\}_s)$ $\{Z_3^{n+1}\}_b = \{Z_3^n + \sigma\lambda B\bar{X}^n\}_b \left(1 - \max\left\{0, 1 - \frac{1}{\ \{Z_3^n + \sigma\lambda B\bar{X}^n\}_b\ _2}\right\}\right)$ $X^{n+1} = X^n - \tau K^T Z^{n+1}$ $\bar{X}^{n+1} = X^{n+1} + \theta(X^{n+1} - X^n)$ $s = 1, \dots, \tilde{M}\tilde{N}W^2$ $b = 1, \dots, (3\tilde{M}\tilde{N} - \tilde{M} - \tilde{N})D$
--

### 6.3 Implementation Details

Because of the discretization, the dimensionality of the problem is quite high. One approach to managing such dimensionality is to use block coordinate descent [29], where one works iteratively on different subsets of the variables. In this paper we consider a simple and efficient approximation: we consider restricting the possible disparities  $\rho_1, \dots, \rho_D$  to a small but carefully selected subset and

---

<sup>1</sup> The total variation term introduces 2 blocks for any pixel in  $\Omega$  except for the left hand side column and the bottom row of pixels (total blocks is  $2(\tilde{M} - 1)(\tilde{N} - 1)$ ). These two rows of pixels, except for the bottom right corner, introduce only one block (total blocks  $(\tilde{M} - 1) + (\tilde{N} - 1)$ ). Finally, the block sparsity term introduces  $\tilde{M}\tilde{N}D$  blocks.



always work with that subset. To gain additional freedom, at each pixel  $(x, y)$  we make a different choice of such subset. Our strategy is to evaluate the function

$$g_{x,y}(\rho) = \sum_{i,j} \sum_{p>i,q>j} \Phi(L_{i,j}(x, y) - L_{p,q}(x - \rho\Delta_x(p - i), y - \rho\Delta_y(q - j))) \quad (31)$$

for as many  $\rho$  values as possible. Then, we sort  $g_{x,y}$  in ascending order and take the disparities corresponding to the first 5 values of  $g_{x,y}$ . We then also add 5 more disparity candidates by selecting the disparities of neighboring pixels (in a 4-neighborhood structure) corresponding to the smallest cost. The purpose of this second group of disparity candidates is to allow (spatially) smooth disparity estimates.

## 7 Experimental Results

We study the performance and robustness of our light field disparity estimation framework on different datasets, Buddha [31], Watch [1], Amethyst and Truck from the Stanford light field archive.<sup>2</sup> We compare our results with two light field depth estimation schemes [19,32], and convex formulations [26]. Our parameters are:  $\mu = 0.6$  and  $\gamma = 1$  for all datasets, and  $\lambda = 0.1$  for Amethyst and Truck. We work with  $5 \times 5$  pixels patches ( $W = 5$ ). Our algorithm is also demonstrated in the limit case where there are only two views (stereo). The group sparsity constraint can still work quite successfully. Another important factor is the input image size. We find that the method works better with high resolution images. However, it can also perform reasonably well on low-resolution data. In contrast, [19,32] are challenged with few views and/or low-resolution images. The runtime of our algorithm is higher than [19]. If parallelism is fully exploited the ideal running time is about 1-3 minutes depending on the resolution and number of views. In our experiments we search through 200 disparity candidates to determine the 10 candidates. Figure 1 compares our scheme with simple plane sweep disparity search (independently at each pixel). We observe that our scheme imposes the global smoothness on the estimated disparity while the plane sweep fails to provide a smooth disparity map. As expected, the number of views used in the disparity estimation problem improves the depth estimate considerably. In our approach an increase in the number of views results in more samples per disparity candidate in the  $Q$  matrix, and therefore a better chance of fitting data more reliably. This is clearly noticeable in Fig. 1 and Fig. 2. We compare qualitatively our disparity estimation algorithm with the techniques introduced in [14,32] in Table 2. It is clear that our scheme provides a better reconstruction quality. In Fig. 4 we illustrate how the patch size  $W$  has an immediate effect on the recovered depth map. As is well known, the larger the patch, the less noisy the depth estimate is. However, increases in patch size also affect the performance

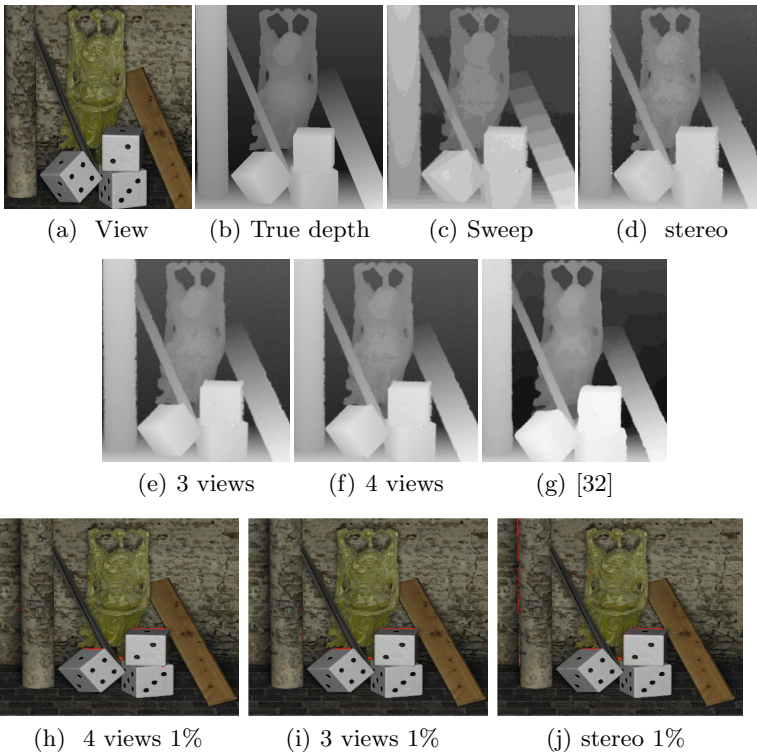
---

<sup>2</sup> See <http://lightfield.stanford.edu>.

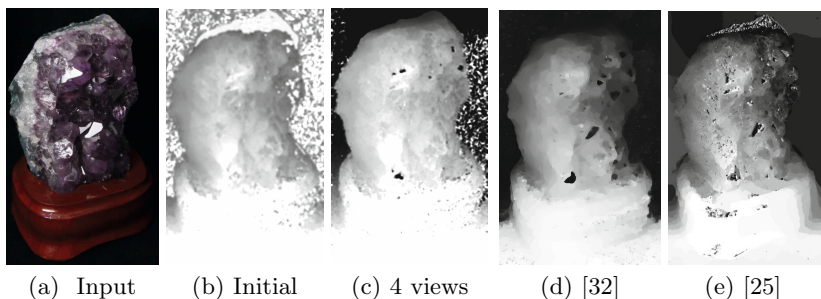
of the algorithm in the recovery of small details. More comparisons are included in [15].

**Table 2.** Qualitative results for Buddha shown in Fig. 1. The table shows the percentage of pixels with relative depth error of more than 0.2%, 0.5% and 1%.

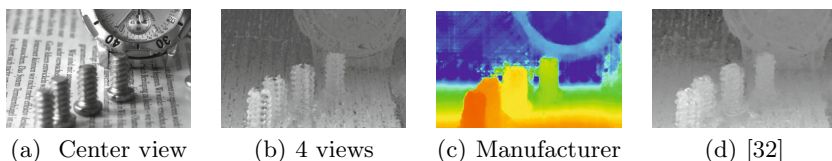
4 views			3 views			stereo			[14]			[32]	
1%	0.5%	0.2%	1%	0.5%	0.2%	1%	0.5%	0.2%	1%	0.5%	0.2%	1%	0.2%
0.13	0.33	1.9	0.139	0.33	1.99	0.42	0.85	3.26	1.15	2.44	15.05	2.9	60.4



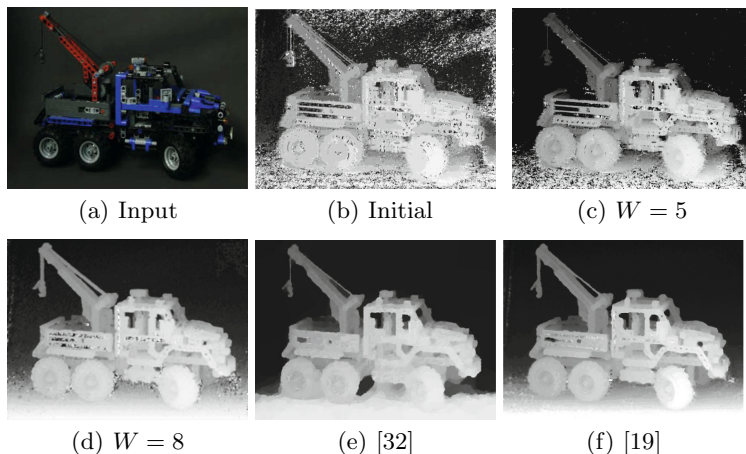
**Fig. 1.** Buddha dataset: Comparison of the depth maps obtained from our method with the ground truth. From left to right, top row shows: the center view, the ground truth, the depth map obtained by plane sweep depth search (independently at each pixel). Middle row: the estimated depth map using different number of views, and the depth map obtained from [32]. Bottom: the estimated disparity in areas with error more than 1% are highlighted in red. We observe that an increase in the number of views improves the reconstruction quality and our scheme provides sharp edges while the depth map estimated using [32] blurs the edges and has staircasing artifacts.



**Fig. 2.** Amethyst dataset. (a) One of the input images. (b) Initial depth estimate (plane sweep depth search) (c) Estimated disparity using our scheme. (d-e) Estimated depth map using [32] and [25]. Notice how we obtain a reasonable estimate of the top part of the stone, while competing methods either fail or obtain a noisier estimate.



**Fig. 3.** Depth estimation with the Raytrix plenoptic camera (handheld light field camera). We compare our algorithm with the reference depth provided by the manufacturer and [32]. Our scheme on a handheld light field camera yields a more detailed depth map.



**Fig. 4.** Truck dataset. We assess the influence of patch size in our scheme. Increasing the patch size results in a less noisy, but also smoother, depth map. In comparison to [19, 32], our algorithm provides sharper edges with a noisier background. This is due to two main reasons: 1) The initial 10 disparity candidates selected among 200 candidates do not contain the true disparity value, which can be improved by working on 200 candidates using block coordinate descent [29]. 2) The selection of the highest coefficients in  $C$  may lead to noisy disparity which can be addressed by imposing smoothness in the final estimation of the disparity from the coefficients of  $C$ .

## 8 Conclusions

We have presented a novel convex formulation to estimate depth from light field data. The method is based on a careful discretization of disparity values and exploits a linear patch-based formulation to represent patches in one view with patches in other views. The proposed model can easily be extended to handle simple departures from the ideal Lambertian model. For example, the current model can already handle contrast changes due to illumination (these changes would be reflected in the magnitude of the coefficients of  $C$ ). The problem of depth estimation is cast as a minimization problem subject to group sparsity constraints and spatial smoothing. To gain computational efficiency we use the primal-dual method. This results in an algorithm where each dual variable update can be computed easily, independently and efficiently. Our experiments show that this method competes well with the state of the art.

## References

1. Raytrix, <http://www.raytrix.de/>
2. Ayvaci, A., Raptis, M., Soatto, S.: Sparse occlusion detection with optical flow. *IJCV* (2012)
3. Basha, T., Avidan, S., Hornung, A., Matusik, W.: Structure and motion from scene registration. In: *CVPR. IEEE* (2012)
4. Bishop, T., Favaro, P.: The light field camera: extended depth of field, aliasing and superresolution. *PAMI* (2012)
5. Bolles, R.C., Baker, H.H., Marimont, D.H.: Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV* (1987)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *PAMI* (2001)
7. Combettes, P.L., Pesquet, J.C.: Proximal Splitting Methods in Signal Processing. In: *Fixed-Point Alg. for Inv. Prob. in Science and Eng.* (2011)
8. Donatsch, D., Bigdeli, S.A., Robert, P., Zwicker, M.: Hand-held 3d light field photography and applications. *The Visual Computer* (2014)
9. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: *CVPR. IEEE* (2009)
10. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE* (2002)
11. Fitzgibbon, A.W., Wexler, Y., Zisserman, A., et al.: Image-based rendering using image-based priors. In: *ICCV*, vol. 3, pp. 1176–1183 (2003)
12. Goldluecke, B., Cremers, D.: An approach to vectorial total variation based on geometric measure theory. In: *CVPR* (2010)
13. Goldluecke, B., Magnor, M.A.: Joint 3d-reconstruction and background separation in multiple views using graph cuts. In: *CVPR. IEEE* (2003)
14. Heber, S., Ranftl, R., Pock, T.: Variational shape from light field. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMMCVPR 2013. LNCS*, vol. 8081, pp. 66–79. Springer, Heidelberg (2013)
15. Hosseini Kamal, M., Favaro, P., Vandergheynst, P.: A Convex Solution to Disparity Estimation from Light Fields via the Primal-Dual Method. [oai:infoscience.epfl.ch/202076](http://oai.infoscience.epfl.ch/202076) (2014)

16. Humayun, A., Mac Aodha, O., Brostow, G.J.: Learning to find occlusion regions. In: CVPR. IEEE (2011)
17. Kang, S.B., Szeliski, R.: Extracting view-dependent depth maps from a collection of images. *IJCV* 58(2), 139–163 (2004)
18. Kang, S.B., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. In: CVPR. IEEE (2001)
19. Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A., Gross, M.: Scene reconstruction from high spatio-angular resolution light fields. In: SIGGRAPH (2013)
20. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. *IJCV* (2000)
21. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: Dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
22. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: ICML (2010)
23. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. CSTR (2005)
24. Pock, T., Chambolle, A.: Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In: ICCV, pp. 1762–1769 (2011)
25. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: Global solutions of variational models with convex regularization. *SIAM J. on Imag. Sciences* (2010)
26. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 792–805. Springer, Heidelberg (2008)
27. Sun, X., Mei, X., Zhou, M., Wang, H., et al.: Stereo matching with reliable disparity propagation. In: 3DIMPVT. IEEE (2011)
28. Szeliski, R., Scharstein, D.: Symmetric sub-pixel stereo matching. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part II. LNCS, vol. 2351, pp. 525–540. Springer, Heidelberg (2002)
29. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* (2001)
30. Vaish, V., Wilburn, B., Joshi, N., Levoy, M.: Using plane+ parallax for calibrating dense camera arrays. In: CVPR. IEEE (2004)
31. Wanner, S., Meister, S., Goldluecke, B.: Datasets and benchmarks for densely sampled 4d light fields. In: Vision, Modelling and Visualization, (VMV) (2013)
32. Wanner, S., Goldluecke, B.: Globally consistent depth labeling of 4d light fields. In: CVPR. IEEE (2012)
33. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. *PAMI* 31(2), 210–227 (2009)
34. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. of the Royal Statist Society: Series B (Stat. Meth.)* (2006)
35. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV –  $\ell_1$  range image integration. In: ICCV, pp. 1–8. IEEE (2007)
36. Ziegler, R., Bucheli, S., Ahrenberg, L., Magnor, M., Gross, M.: A bidirectional light field-hologram transform. In: Computer Graphics Forum., vol. 26, pp. 435–446. Wiley Online Library (2007)

# Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames

Ryan Kennedy and Camillo J. Taylor

Department of Computer and Information Science  
University of Pennsylvania  
{kenry, cjtaylor}@cis.upenn.edu

**Abstract.** Optical flow research has made significant progress in recent years and it can now be computed efficiently and accurately for many images. However, complex motions, large displacements, and difficult imaging conditions are still problematic. In this paper, we present a framework for estimating optical flow which leads to improvements on these difficult cases by 1) estimating occlusions and 2) using additional temporal information. First, we divide the image into discrete triangles and show how this allows for occluded regions to be naturally estimated and directly incorporated into the optimization algorithm. We additionally propose a novel method of dealing with temporal information in image sequences by using “inertial estimates” of the flow. These estimates are combined using a classifier-based fusion scheme, which significantly improves results. These contributions are evaluated on three different optical flow datasets, and we achieve state-of-the-art results on MPI-Sintel.

## 1 Introduction

Optical flow has a long history and many different methods have been used. Several modern methods have their roots in the seminal work of Horn and Schunck [1]. Current variants of this approach employ robust cost functions [2] and modern optimization techniques [3] and can compute optical flow accurately and efficiently for many types of images. This is corroborated by results on the Middlebury dataset [4], for which the top-performing algorithms are nearly error-free.

Despite this success, optical flow is far from solved. The underlying assumption of most models is that matching pixels should have similar intensity values and nearby pixels should have similar motions. However, these assumptions are violated in many situations, especially when motion blur, lighting variation, large motions, and atmospheric effects are involved. While Middlebury does not contain these real-world situations, more recent datasets [5,6] have many of these difficulties and the error rates of the best methods are correspondingly much higher. It remains an open question of how best to deal with these complex motions and imaging conditions.

In this paper, we present a framework for optical flow that leads to an improvement for difficult datasets. Our method is based on a triangulation of the image domain, over which we compute optical flow. We employ a tectonic model where the triangular facets are allowed to move relative to each other and a numerical quadrature scheme is used

to handle the resulting occlusion effects. This allows for occlusions to be directly incorporated into the optimization procedure without the need for arbitrary regularization terms.

Additionally, we describe a novel way to incorporate temporal information over multiple frames. First, we propose “inertial estimates,” which are estimates of the optical flow based on nearby frames. Next, we combine these estimates using a classifier-based fusion. Although fusion-based methods have been used previously [7], our approach is fundamentally different and does not require optimizing an NP-hard quadratic binary program. The approach is also agnostic to the underlying optical flow method and could be used with other algorithms.

In summary, we make the following four contributions:

- We show how occluded regions can be easily detected during optimization by using a triangulation of the image domain.
- We introduce “inertial estimates” which provide multiple motion estimates based on nearby frames.
- We show how fusing inertial estimates using a classifier provides a simple and effective means of incorporating temporal information into optical flow.
- We evaluate our method on multiple datasets, achieving state-of-the-art accuracy on the difficult MPI-Sintel dataset [5].

## 2 Related Work

Occlusions are often modeled as noise and handled through a robust cost function [2], but an explicit model for occlusions is desirable in many situations. One common approach is to compare forward and backward flow estimates, which will not match for occluded pixels [8]. Strecha *et al.* [9] use a probabilistic framework and estimate occlusions as latent variables. Occlusions can also be dealt with by incorporating layers into the model [10]. In [11], layer ordering was determined using the relative data cost between overlapping layers. Another approach is that of Xu *et al.* [12] and Kim *et al.* [13], who label points as occluded when multiple pixels map to a single point. The most similar approach to our own is from Sun *et al.* [14], who jointly estimate motion and occlusions, although their method is significantly more complex than our own and is only used to fine-tune flow fields computed using other methods.

Multi-frame optical flow estimation is often approached by assuming that flow estimates are smooth over time as well as space [15,10,16], rather than using a fusion method as we do here. Fusion methods have usually been proposed in the context of fusing the results of multiple algorithms [17] rather than incorporating temporal information. Lempitsky *et al.* [7] write the fusion problem as a quadratic binary program which they then approximate the solution to, while Jung *et al.* [18] use an algorithm involving random partitions of the flow estimates.

Triangulations of an image were used in optical flow estimation by Glocker, *et al.* [11], although otherwise their approach is quite different from ours; their triangulations are used for incorporating higher-order likelihood terms, while we use triangulations to model the flow and the resulting occlusions. Superpixel-based approaches that are not based on triangles have also been previously applied to optical flow problems [19].



**Fig. 1.** A triangulated section of an image. Blue circles denote edge points and red squares denote points generated on a uniform grid with a spacing of 5 pixels. The Delaunay triangulation given by the green lines tessellates the image into regions which form the basis of our algorithm. In practice, the data cost functions are evaluated at a set of quadrature points within each triangle, shown here as black dots.

### 3 Problem Setup

Let  $I_1, I_2 : (\Omega \subseteq \mathbb{R}^2) \rightarrow \mathbb{R}^d$  be two  $d$ -dimensional images. In this paper, we consider both to be color images in the CIE Lab color space such that  $d = 3$ . Channels are denoted using a superscript, such as  $I_1^{(c)}$ . We attempt to estimate the motion of each point from  $I_1$  to  $I_2$ . The estimated motions in the horizontal and vertical directions are denoted by  $u$  and  $v$  respectively. Let  $\mathbf{x} = (x, y)$  be a point in  $\Omega$ , and let  $f : \Omega \rightarrow \mathbb{R}^2$  be a function such that  $f(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$ , that is  $f$  returns the estimated motion vector associated with every point in the image. In addition, we estimate a function  $m : \Omega \rightarrow \mathbb{R}$  which is a multiplicative factor that measures changes in lightness between frames, as we will define in Section 4. This “generalized brightness constancy” model has been previously used [20], and we found that it improved our results.

A key aspect of our approach is that we consider the image to be a continuous 2D function of the image domain, rather than a set of sampled pixel locations. We do so by extending the sampled pixel values to intermediate locations in the image plane using bicubic interpolation. More specifically, at any continuous-valued location  $(x, y) \in \Omega$ , the value of the image at channel  $c$  is computed using a quadratic form

$$I_1^{(c)}(x, y) = [x^3 \ x^2 \ x \ 1] K_c [y^3 \ y^2 \ y \ 1]^T, \quad (1)$$

where  $K_c$  is the matrix of coefficients based on the channel values of nearby pixels. Note that spatial image derivatives at any point are easily computed using derivatives of this quadratic form. Given this representation of the image as a continuous function, our goal is to compute a corresponding continuous function  $f(\cdot)$  that specifies the motion of each point in  $\Omega$ , along with a multiplicative brightness factor  $m(\cdot)$ .

We discretize the problem by tessellating the image  $I_1$  into discrete triangular regions (Figure 1), and then seek to estimate a constant motion vector  $f(\cdot)$  and brightness offset  $m(\cdot)$  for each triangle. Because we assume the motion to be constant within each triangle, the triangles should be made to conform to the content of the image in order



to find an accurate solution. This approach is similar to that of [11], where a triangulation of the image domain was also used. We use the following procedure. First, we extract edges from the image  $I_1$  by using the method of [21] and threshold the given ultrametric contour map at 0.2. Each edge pixel in the image is then used as a vertex in our triangulation. In addition to these points, we also use a set of grid points that are evenly spaced throughout the 2D image, which serve to limit the maximum dimension of the resulting triangles. The grid points and edge pixels are combined and a Delaunay triangulation is constructed. An example of a tessellated image is shown in Figure 1.

### 4 Cost Function

Our cost function consists of data terms and smoothness terms. The data terms penalize incorrectly-matched pixels based on image data, while the smoothness terms encourage solutions that are smooth over the image domain. Our cost function takes the form

$$\mathcal{E}(f, m) = \mathcal{D}(f, m) + \tau_0 \mathcal{F}(f) + \tau_1 \mathcal{S}_1(f) + \tau_2 \mathcal{S}_2(f) + \tau_3 \mathcal{S}_3(m) , \tag{2}$$

where  $\mathcal{D}(\cdot)$  is a data cost term based on image data,  $\mathcal{F}(\cdot)$  is a feature matching term, and  $\mathcal{S}_1(\cdot), \mathcal{S}_2(\cdot)$  and  $\mathcal{S}_3(\cdot)$  are smoothness terms. The parameters  $\tau_0, \tau_1, \tau_2$  and  $\tau_3$  control the tradeoff between these terms.

The cost function will be defined as an integral over the entire continuous image domain. We approximate this continuous integral by considering a discrete set of *quadrature points* within each triangle using the scheme described in [22]. The integral is then approximated by forming a weighted sum of the cost function evaluated at these points. We used 3 quadrature points per triangle, as shown in Figure 1.

#### 4.1 Data Term

Our data term is given by the equation

$$\mathcal{D}(f, m) = \int_{\Omega} \Phi_{\gamma} \left( I_2(\mathbf{x} + f(\mathbf{x})) - \begin{bmatrix} m(\mathbf{x}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} I_1(\mathbf{x}) \right) dx , \tag{3}$$

where  $\Phi_{\gamma}(\cdot)$  is a robust error function with parameter vector  $\gamma$ . Because of the large amount of data made available in the MPI-Sintel dataset, we chose our robust cost function through a fitting procedure. In particular, the difference values,  $I_2^{(c)}(\mathbf{x} + f(\mathbf{x})) - I_1^{(c)}(\mathbf{x})$ , are well-modeled by a Cauchy distribution, as has been previously observed [23]. The robust function  $\Phi_{\gamma}(\cdot)$  is then the negative log-likelihood of the Cauchy density function, summed over all channels:

$$\Phi_{\gamma}(\delta) = \sum_{c=1}^d \log [\pi(\delta_c^2 + \gamma_c^2)/\gamma_c] . \tag{4}$$

A separate distribution was fit to the lightness and to the combined color channels, giving values of  $\gamma_1 = 0.3044$  for lightness and  $\gamma_2 = \gamma_3 = 0.2012$  for the color channels.

### 4.2 Feature Matching Term

Feature matching has been shown to be effective at improving optical flow results, especially for large motions [24,12]. We use HOG features [25], computed *densely* at every pixel. These descriptors are then matched to their nearest neighbor in the opposite image using the approximate nearest neighbors library FLANN [26]. The matches from  $I_1$  to  $I_2$  generate motion estimates for each of the pixels, which we denote as  $f_{HOG} : \Omega \rightarrow \mathbb{R}^2$ .

If the HOG match is correct, then it is desirable to have  $f(\mathbf{x})$  be close to  $f_{HOG}(\mathbf{x})$ . Thus, our feature matching term is given by

$$\mathcal{F}(f) = \int_{\Omega} s(\mathbf{x})\Psi_{\alpha} (\|f(\mathbf{x}) - f_{HOG}(\mathbf{x})\|_2) d\mathbf{x} , \tag{5}$$

where

$$\Psi_{\alpha}(\delta) = (\delta^2 + \epsilon)^{\alpha} \tag{6}$$

is a robust cost function with parameter  $\alpha$  and small constant epsilon (i.e.,  $\epsilon = 0.001$ ) [3]. For  $\alpha = 1$ , this is a pseudo- $\ell_2$  penalty. As  $\alpha$  decreases, it becomes less convex with it becoming a pseudo- $\ell_1$  penalty for  $\alpha = 0.5$ . For our feature matching term, we set  $\alpha = 0.5$ .

The function  $s : \Omega \rightarrow \mathbb{R}$  is a weighting function which measures the confidence in each HOG match, and is defined as follows. First, we enforce forward-backward consistency by setting  $s(\mathbf{x}) = 0$  if a match is not a mutual nearest-neighbor. Otherwise, we let  $s(\mathbf{x}) = ((d_2 - d_1)/d_1)^{0.2}$ , where  $d_i$  is the  $\ell_1$  distance between the HOG feature vector in  $I_1$  at location  $\mathbf{x}$  and its  $i^{\text{th}}$ -closest match in  $I_2$ . This is similar to the weight used in [24] and provides a measure of confidence for each HOG match.

When evaluating this term on a triangulation, each triangle is assigned a HOG flow estimate by taking the mean of all flow values within the triangle  $t$  weighted by their confidence scores,  $\sum_{\mathbf{x} \in t} \frac{s(\mathbf{x})}{\sum_{\mathbf{x} \in t} s(\mathbf{x})} f_{HOG}(\mathbf{x})$ . The confidence of each triangle is similarly set to the average of its confidence values. These flow values are then used for all quadrature points within each triangle when evaluating the cost function.

While we used HOG features due to their speed and simplicity, more complex feature matching could be used here as well, such as [27] or [28].

### 4.3 Smoothness Terms

We use two different smoothness terms in our cost function: a first-order term that penalizes non-constant flow fields, and a second-order term that penalizes non-affine flow fields.

**First-Order Smoothness.** A first-order smoothness term penalizes non-constant motion estimates. In our cost function, all pairs of neighboring triangles are considered. The cost is defined as

$$\mathcal{S}_1(f) = \sum_{t_i, t_j \in N} |t_i||t_j| \Psi_{\alpha} \left( \frac{\|f(t_i) - f(t_j)\|_2}{\|\bar{t}_i - \bar{t}_j\|_2} \right) , \tag{7}$$

where  $N \subseteq T \times T$  is the set of all neighboring triangles  $T$  in the tessellation,  $\bar{t}_i$  is the centroid of triangle  $t_i \in T$ , and  $|t_i|$  is its area. The function  $\Psi_\alpha(\cdot)$  is a robust cost function, which was defined in Equation (6).

This cost function penalizes differences in the flows between neighboring triangles, modulated by the distance between their centroids. Note that we also multiply by the area of the two triangles (rather than by the edge length), which effectively connects all points within one triangle to all points in the other triangle. Now, recall that our triangulation is constructed using both edges points and a set of uniform grid points (Figure 1). The triangles along edges will therefore tend to have a smaller area, resulting in a weaker smoothness constraint. In this way, our triangulation naturally allows for a non-local smoothness cost [29].

We also apply this same smoothness cost to the multiplicative term  $m$  to encourage only locally-consistent changes in image brightness. This is denoted as the function  $\mathcal{S}_3(m)$ , and for this we use  $\alpha = 0.5$ .

**Second-Order Smoothness.** While a first-order smoothness term penalizes non-constant flows, a second-order smoothness term penalizes non-planar flows. This allows for motion fields with a constant gradient, which is important for datasets where such motions are common, such as KITTI (Section 8.3).

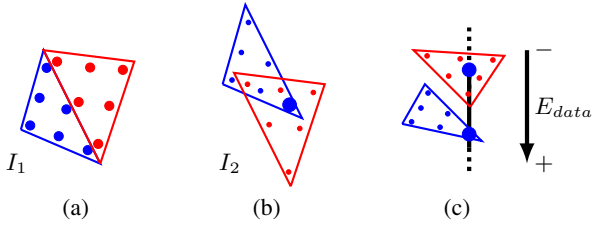
Intuitively, our second order smoothness term says that the flow of each triangle is encouraged to be near the plane that is formed from the flow values of its three neighbors. Formally, the cost function is written as a sum of costs over all triangles  $t \in T$ :

$$\mathcal{S}_2(f) = \sum_{t \in T} |t_i| |t_j| |t_k| \Psi_\alpha \left( \frac{\|f(t) - [\lambda_i f(t_i) + \lambda_j f(t_j) + \lambda_k f(t_k)]\|_2}{|\Delta_{ijk}|} \right). \quad (8)$$

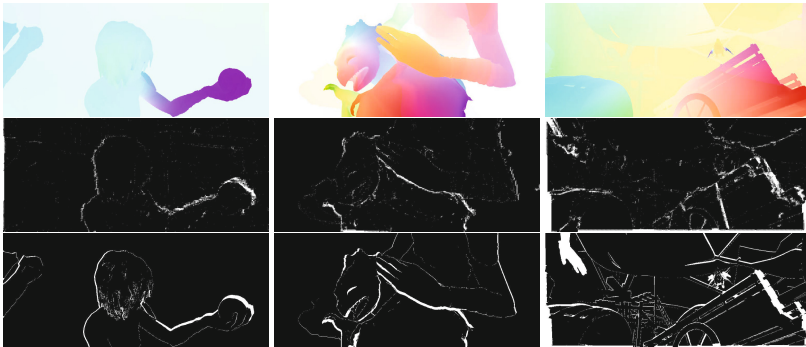
Here,  $t_i, t_j$  and  $t_k$  are the three neighboring triangles to  $t$ . The values  $\lambda_i, \lambda_j$  and  $\lambda_k$  are the barycentric coordinates of the centroid of  $t$  with respect to the centroids of  $t_i, t_j$  and  $t_k$ . In other words, the numerator is exactly zero when the flow vector associated with the triangle  $t$  can be linearly interpolated from the values associated with the neighboring triangles. This discrepancy is then normalized by  $|\Delta_{ijk}|$ , the area of the triangle formed by connecting the centroids of  $t_i, t_j$  and  $t_k$ , making the cost akin to a finite-difference approximation of the Laplacian. Similar to the first-order smoothness term, the function  $\Psi_\alpha(\cdot)$  is a robust cost function and each term is multiplied by the areas of the three neighboring triangles to impart a non-local character to the cost.

## 5 Occlusion Reasoning

Since we model an image as a set of triangular pieces that can move independently, we can directly reason about occlusions. A depiction of this process is shown in Figure 2. At each iteration of our algorithm, for each quadrature point in each triangle of  $I_1$ , we compute where it appears in the other image  $I_2$ . We then determine whether any other triangles overlap it in  $I_2$ . For each of these overlapping triangles, we determine whether that triangle offers a better explanation for that location as measured by the data



**Fig. 2.** Depiction of our occlusion term. **(a)** Two triangles and their quadrature points in the tessellation of  $I_1$ . **(b)** The triangles are moved to their estimated locations in  $I_2$ , where they now overlap. Each quadrature point is processed separately and we have highlighted one quadrature point as an example. **(c)** The data cost is compared for all overlapping triangles at the quadrature point. The quadrature point here has a lower data cost at the same location in the red triangle, and so we mark the quadrature point as occluded.



**Fig. 3.** Examples of our occlusion estimation on MPI-Sintel. During optimization, the occlusion status of each quadrature point in each triangle is directly estimated. For visualization, we label each triangle a value in  $[0, 1]$  as the proportion of its quadrature points that are labeled occluded, and then each pixel is labeled based on the triangles that it overlaps. **Top:** Groundtruth flow. **Middle:** Estimated occlusions. **Bottom:** Groundtruth occlusions.

cost (Equation (3)). If a better solution exists, then the quadrature point in question is labeled as occluded. The occluded quadrature points are not included in the evaluation of the data cost. In this way, the data cost function only includes points which are estimated to be unoccluded. Note that these occlusion estimates are generated *directly* from the geometry and from the data cost term; no additional regularization parameters are needed to avoid the trivial solution of labeling all points occluded. An example of our occlusion estimation is shown in Figure 3.

Occlusions can be calculated efficiently by rasterizing the triangles to find which pixels they overlap in  $I_2$ . When evaluating the occlusion term for a quadrature point, only triangles rasterized to the same pixel need to be considered as potential occluders.

## 6 Optimization

As is standard, local optimization is carried out within a coarse-to-fine image pyramid [30]. We begin with a zero-valued flow at the coarsest level and iteratively perform local optimization until a local minimum is reached. During this process, image values and gradients are calculated using bicubic interpolation. The resulting solution is then propagated to the next pyramid level where it is used as an initialization and the local optimization is repeated. At each level, a new triangulation is calculated as described in Section 3.

Rather than linearizing the Euler-Lagrange equations [30], we use Newton's method, a second-order optimization algorithm. Newton's method provides flexibility to our framework since any suitably-differentiable function can be substituted for our cost function without changing the optimization scheme. To find the Newton step at each iteration, a sparse linear system must be solved. This is commonly done with an iterative method, such as Successive Over-Relaxation (SOR). Instead, we decompose the Hessian matrix into its Cholesky factorization, after which the linear system can be solved directly. Cholesky factorization is often avoided since it has the potential to use a significant amount of memory, but we have found that the use of a triangulation makes it possible to reduce these memory requirements. First, there are often fewer triangles than pixels, resulting in a smaller linear system. Also, the memory requirement for Cholesky factorization is dependent on the adjacency structure of the matrix, which gives triangulations an advantage since each triangle has only three neighbors rather than four or eight. We have found that the resulting Hessian matrices can be efficiently reordered and factorized using algorithms such as [31].

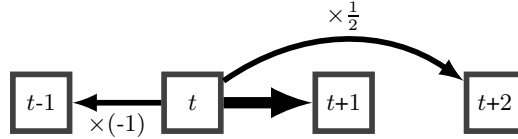
## 7 Multi-frame Fusion of Inertial Estimates

A significant challenge for modern optical flow algorithms is when objects move large distances. This is especially true when objects move either into or out of frame, for which there are no matches. In this case, it is often not possible to estimate the motion of these pixels from two-frame optical flow. In this section, we address this by proposing a simple method of incorporating temporal information from adjacent frames.

### 7.1 Inertial Estimates

Let  $[t \rightarrow (t + 1)]$  denote the estimated flow between frames  $t$  and  $t + 1$ , and suppose that we also have access to frames  $t - 1$  and  $t + 2$ . If it is assumed that objects move at a constant velocity (i.e., they are carried by inertia) and move parallel to the image plane, then an estimate of the motion from  $[t \rightarrow (t + 1)]$  is given by  $-[t \rightarrow (t - 1)]$ , which is found by computing the flow from  $t$  to  $t - 1$  and negating it, as shown in Figure 4. Similarly, another estimate can be found using frame  $t + 2$  as  $\frac{1}{2}[t \rightarrow (t + 2)]$ . We call these "inertial estimates" since they provide an estimate of the flow by assuming that inertia moves all objects at a constant velocity. All three inertial estimates are computed independently, using the same optical flow algorithm.

Of course, these estimates will, on average, be inferior to using  $[t \rightarrow (t + 1)]$  directly. However, if an object is visible in frame  $t$  and moves out of frame in  $t + 1$ , then it may



**Fig. 4.** Inertial flow estimates used in multi-frame fusion. In addition to using the two-frame estimate  $[t \rightarrow (t + 1)]$  directly, we also estimate the flow from  $[t \rightarrow (t - 1)]$  and from  $[t \rightarrow (t + 2)]$ . These two estimates are then multiplied by the factors  $-1$  and  $\frac{1}{2}$ , respectively, to give an estimate of the desired flow  $[t \rightarrow (t + 1)]$ . All three flow estimates are then fused using a classifier.

still be visible in  $t - 1$  and so  $-[t \rightarrow (t - 1)]$  will likely give a better estimate for that part of the image. Similarly, using the estimate  $\frac{1}{2}[t \rightarrow (t + 2)]$  will provide an additional source of information.

## 7.2 Classifier-Based Fusion

The three inertial estimates  $[t \rightarrow (t + 1)]$ ,  $\frac{1}{2}[t \rightarrow (t + 2)]$  and  $-[t \rightarrow (t - 1)]$  must be fused. We do so by training a random forest classifier whose output tells us which estimate for each pixel is predicted to have the lowest error. We use the following features:

- The tail probability for the Cauchy distribution used in the match cost  $\mathcal{D}(\cdot)$ . This value varies from 0 to 1 with larger values indicating a better match. The index of the flow estimate with the best score, and its associated score, are also used.
- Each pixel in frame  $t$  is projected forward via the flow estimate and then projected back using the backward flow. The Euclidean norm of this discrepancy vector is used as a feature for each flow estimate. The index of the flow estimate with the smallest discrepancy and its corresponding value are also included as features.
- The flow estimates  $u$  and  $v$ , and the magnitudes  $\sqrt{u^2 + v^2}$ .
- The multiplicative offset  $m(\mathbf{x})$  at each pixel.
- For every pixel, an indicator of whether the pixel is estimated to be occluded.
- The  $(x, y)$  location of each pixel.

This results in a total of 27 features. For each dataset that we evaluated our methods on, we sampled a number of points uniformly at random from the associated training images such that the resulting dataset had  $\sim 10^6$  observations.

In this classification problem, not all data points should be counted equally. In particular, a misclassification is more costly when the three inertial estimates have very different errors. To take this into account, each data point was weighted by the difference between the lowest endpoint error of all three flow estimates and the mean of the other two. This weighting indicates how important each datapoint is. We then trained a random forest classifier with 500 trees using MATLAB's `TreeBagger` class. An example of our fusion is shown in Figure 5 on the MPI-Sintel dataset [5]. As a final step in our procedure, a median filter of size  $15 \times 15$  was applied.



**Fig. 5.** Examples of our multi-frame fusion from the MPI-Sintel Final training set. **Top row:** Frame at time  $t$ . **Rows 2-4:** Inertial estimates of the flow  $[t \rightarrow t + 1]$ ,  $-[t \rightarrow t - 1]$ , and  $\frac{1}{2}[t \rightarrow t + 2]$ . **Row 5:** Fusion classification for each pixel. Color indicates the estimate used at each pixel. Colors correspond to the border colors of the inertial estimates. **Row 6:** Fused flow estimate. **Bottom row:** Groundtruth flow. For all flow estimates, the endpoint error is printed in the image.

## 8 Experiments

We evaluate our algorithm on three datasets. The free parameters of our method are  $\tau_0, \tau_1, \tau_2, \tau_3$ , the value of  $\alpha$  used in the smoothness terms  $\mathcal{S}_1(\cdot)$  and  $\mathcal{S}_2(\cdot)$ , and the spacing of the uniform grid used in the triangulation. For  $\mathcal{S}_3(\cdot)$ , we set  $\alpha = 0.5$  for all experiments. Parameters were chosen for each dataset using a small-scale grid search on the training data.

We use a scale of 0.95 between pyramid levels – resulting in around 60 levels – and used 10 Cholesky-based iterations of Newton’s method at each level.

Our basic method is denoted as TF (TriFlow), and when occlusion estimation, multi-frame fusion and median filtering are used, they are denoted as “O” (occlusion), “F” (fusion) and “M” (median filtering), respectively. Our final method with all components is thereby denoted as TF+OFM.

## 8.1 Middlebury

We begin with the Middlebury dataset [4] since it is a standard benchmark for optical flow, although it has only small and simple motions. For this dataset, the parameters were set to  $\tau_0 = 0$ ,  $\tau_1 = 3.5$ ,  $\tau_2 = 0$ ,  $\tau_3 = 25$ ,  $\alpha = 0.36$ , and a small grid spacing of 2 pixels was used in order to capture the small details in this dataset.

Results on the test dataset are given in Table 1. Note that we did not evaluate our multi-frame fusion since the dataset was too small for a reliable classifier to be trained. Our results are comparable to other similar coarse-to-fine methods such as DeepFlow [27]. Our occlusion estimation provides little benefit in this case, since the dataset has very small occlusion regions.

**Table 1.** Endpoint error on the Middlebury test dataset. Our results are comparable with similar coarse-to-fine methods

	Army	Mequon	Scheff.	Wooden	Grove	Urban	Yosemite	Teddy	mean
TF+OM	0.10	0.22	0.36	0.20	0.98	0.56	0.16	0.76	0.42
Layers++ [10]	0.08	0.19	0.20	0.13	0.48	0.47	0.15	0.46	0.27
MDP-Flow2 [12]	0.09	0.19	0.24	0.16	0.74	0.46	0.12	0.78	0.35
DeepFlow [27]	0.12	0.28	0.44	0.26	0.81	0.38	0.11	0.93	0.42
LDOF [24]	0.12	0.32	0.43	0.45	1.01	0.10	0.12	0.94	0.56

## 8.2 MPI-Sintel

The MPI-Sintel dataset [5] is a large, difficult dataset that includes large displacements, significant occlusions and atmospheric effects. Parameters were set to  $\tau_0 = 0.5$ ,  $\tau_1 = 2.0$ ,  $\tau_2 = 0$ ,  $\tau_3 = 100$ ,  $\alpha = 0.6$ , and the grid spacing was set to 5 pixels.

Results on the MPI-Sintel test dataset are given in Table 2. As of this writing, our method is ranked 2<sup>nd</sup> among all submissions on the Final dataset and it outperforms all other published methods in terms of endpoint error. The results are especially good for unmatched pixels which are helped by our occlusion term and multi-frame fusion. In particular, on the Final dataset the occlusion term improves the error on unmatched pixels by 6.4% and the fusion improves it by an additional 7.2%.

Several examples of results from our multi-frame fusion for the Final version of the training dataset are shown in Figure 5. As we would expect, the inertial estimates that the classifier selects are spatially localized around the edges of objects where occlusions occur. In all cases, the multi-frame fusion significantly reduces the endpoint error.

Figure 3 shows several examples of the occlusion estimates on images from MPI-Sintel. During optimization, the occlusion status of each quadrature point in each triangle is estimated. For visualization, we label each triangle a value in  $[0, 1]$  as the proportion of its quadrature points that are labeled as occluded, and then each pixel is labeled based on the triangles that it overlaps. The occlusion term is able to estimate the occlusions accurately, which results in reduced error.



**Table 2.** Results on the MPI-Sintel test set. Our algorithm outperforms all other published results in terms of endpoint error (EPE) on the Final dataset. The largest change is on unmatched pixels due to our occlusion estimation and multi-frame fusion.

	Final			Clean		
	EPE	matched	unmatched	EPE	matched	unmatched
TF+OFM	<b>6.727</b>	3.388	<b>33.929</b>	4.917	1.874	29.735
TF+OF	6.780	3.436	34.029	4.986	1.937	29.857
TF+O	7.164	3.547	36.657	5.357	2.033	32.474
TF	7.493	3.609	39.170	5.723	2.077	35.471
DeepFlow [27]	7.212	<b>3.336</b>	38.781	5.377	1.771	34.751
AggregFlow [32]	7.329	3.696	36.929	<b>4.754</b>	<b>1.694</b>	<b>29.685</b>
FC-2Layers-FF [16]	8.137	4.261	39.723	6.781	3.053	37.144
MDP-Flow2 [12]	8.445	4.150	43.430	5.837	1.869	38.158
LDOF [24]	9.116	5.037	42.344	7.563	3.432	41.170

### 8.3 KITTI

The KITTI dataset [6] consists of grayscale images taken from a moving vehicle. We used the parameter settings  $\tau_0 = 0.05$ ,  $\tau_1 = 0.02$ ,  $\tau_2 = 7$ ,  $\tau_3 = 125$ ,  $\alpha = 0.6$ , and the grid spacing was set to 5 pixels.

On the KITTI test dataset, error is measured as the percentage of pixels with an endpoint error greater than 3, in addition to the standard endpoint error. Our results on this dataset are given in Table 3. This dataset is quite different than MPI-Sintel: the images are grayscale and have low contrast and the motions are often dominated by that of the camera. Top-performing methods on this dataset take advantage of these properties by using better features such as census transforms and more information such as stereo and epipolar information [33]. However, our results are comparable to similar coarse-to-fine approaches such as DeepFlow [27], especially for endpoint error (which the fusion classifier was trained to minimize).

We also evaluate the effect of our occlusion and fusion terms on a validation set from the training images. For this, 100 training images were used to train a fusion classifier and evaluation was done on remaining 94 images. Results are shown in Table 4. Both the occlusion and multi-frame fusion terms significantly improve results, as measured by either endpoint error or the percentage of pixels with an endpoint error more than 3.

### 8.4 Timing

Timing was evaluated on a laptop with a 1.80 GHz Intel Core i5 processor and 4 GB of RAM. The typical time taken for two-frame flow estimation on a  $1024 \times 436$  image from MPI-Sintel (including all setup and feature matching, but excluding multi-frame fusion) was 500 seconds. About half of this time is spent evaluating the cost function within Newton’s method, and another 20% is spent solving linear systems. Much of our approach can be sped up through parallelization. For example, the cost function evaluation, running the algorithm on all three inertial estimates, and the random forest fusion are all trivially-parallelizable.

**Table 3.** Results on the KITTI test set. We show both the endpoint error (EPE) and the percentage of pixels with an EPE more than 3, for all pixels as well as non-occluded pixels.

	EPE (all)	EPE (not occ.)	% > 3 (all)	% > 3 (not occ.)
TF+OFM	5.0	2.0	18.46%	10.22%
PCBP-Flow [33]	2.2	0.9	8.28%	3.64%
DeepFlow [27]	5.8	1.5	17.79%	7.22%
LDOF [24]	12.4	5.6	31.39%	21.93%
DB-TV-L1 [34]	14.6	7.9	39.25%	30.87%

**Table 4.** Results on a validation set from the KITTI training dataset. The occlusion estimation term and multi-frame fusion significantly improve results.

	EPE	% > 3
TF+OFM	4.23	16.43%
TF+OF	4.32	16.62%
TF+O	5.29	16.91%
TF	6.89	19.96%

## 9 Conclusion

This paper presents a novel framework for estimating optical flow based on a triangulation of the image which improves results in difficult regions due to occlusions and large motions. We use a geometric model that allows us to directly account for occlusion effects. We also present a method that exploits temporal information from adjacent frames by acquiring several flow estimates and fusing them via a classifier. Together, these contributions result in state-of-the-art performance on the MPI-Sintel dataset. Our approach was evaluated on a range of datasets and the results demonstrate that the proposed enhancements have a significant impact on the quality of the resulting flow.

## References

1. Horn, B.K., Schunck, B.G.: Determining optical flow. *Artificial Intelligence* 17(1), 185–203 (1981)
2. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU* 63(1), 75–104 (1996)
3. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: *CVPR* (2010)
4. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *IJCV* 92(1), 1–31 (2011)
5. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part VI*. LNCS, vol. 7577, pp. 611–625. Springer, Heidelberg (2012)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: *CVPR* (2012)
7. Lempitsky, V., Roth, S., Rother, C.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In: *CVPR*, pp. 1–8. IEEE (2008)
8. Xu, L., Chen, J., Jia, J.: A segmentation based variational model for accurate optical flow estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 671–684. Springer, Heidelberg (2008)
9. Strecha, C., Fransens, R., Van Gool, L.: A probabilistic approach to large displacement optical flow and occlusion detection. In: Comaniciu, D., Mester, R., Kanatani, K., Suter, D. (eds.) *SMVP 2004*. LNCS, vol. 3247, pp. 71–82. Springer, Heidelberg (2004)
10. Sun, D., Sudderth, E.B., Black, M.J.: Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In: *NIPS*, pp. 2226–2234 (2010)
11. Glocker, B., Heibel, T.H., Navab, N., Kohli, P., Rother, C.: TriangleFlow: Optical flow with triangulation-based higher-order likelihoods. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part III*. LNCS, vol. 6313, pp. 272–285. Springer, Heidelberg (2010)

12. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. *PAMI* 34(9), 1744–1757 (2012)
13. Kim, T.H., Lee, H.S., Lee, K.M.: Optical flow via locally adaptive fusion of complementary data costs. In: *ICCV* (2013)
14. Sun, D., Liu, C., Pfister, H.: Local layering for joint motion estimation and occlusion detection. In: *CVPR* (2014)
15. Volz, S., Bruhn, A., Valgaerts, L., Zimmer, H.: Modeling temporal coherence for optical flow. In: *ICCV*, pp. 1116–1123. *IEEE* (2011)
16. Sun, D., Wulff, J., Sudderth, E.B., Pfister, H., Black, M.J.: A fully-connected layered model of foreground and background flow. In: *CVPR* (2013)
17. Mac Aodha, O., Humayun, A., Pollefeys, M., Brostow, G.J.: Learning a confidence measure for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(5), 1107–1120 (2013)
18. Jung, H.Y., Lee, K.M., Lee, S.U.: Toward global minimum through combined local minima. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV. LNCS*, vol. 5305, pp. 298–311. Springer, Heidelberg (2008)
19. Chang, H.S., Wang, Y.C.F.: Superpixel-based large displacement optical flow. In: *ICIP*, pp. 3835–3839 (2013)
20. Negahdaripour, S.: Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *PAMI* 20(9), 961–979 (1998)
21. Donoser, M., Schmalstieg, D.: Discrete-continuous gradient orientation estimation for faster image segmentation. In: *CVPR* (2014)
22. Cowper, G.: Gaussian quadrature formulas for triangles. *International Journal for Numerical Methods in Engineering* 7(3), 405–408 (1973)
23. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning optical flow. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III. LNCS*, vol. 5304, pp. 83–97. Springer, Heidelberg (2008)
24. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI* 33(3), 500–513 (2011)
25. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1, pp. 886–893. *IEEE* (2005)
26. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: *VISAPP*, pp. 331–340 (2009)
27. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: *ICCV* (2013)
28. Byrne, J., Shi, J.: Nested shape descriptors. In: *ICCV*, pp. 1201–1208. *IEEE* (2013)
29. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: *CVPR*, pp. 2464–2471. *IEEE* (2010)
30. Brox, T., Bruhn, A., Papenbergh, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004. LNCS*, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
31. Amestoy, P.R., Davis, T.A., Duff, I.S.: Algorithm 837: Amd, an approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.* 30(3), 381–388 (2004)
32. Fortun, D., Boutheimy, P., Kervrann, C.: Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow. *arXiv preprint arXiv:1407.5759v1*
33. Yamaguchi, K., McAllester, D., Urtaasun, R.: Robust monocular epipolar flow estimation. In: *CVPR*, pp. 1862–1869. *IEEE* (2013)
34. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV- $L^1$  optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) *DAGM 2007. LNCS*, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)

# Adaptive Dictionary-Based Spatio-temporal Flow Estimation for Echo PIV

Ecaterina Bodnariuc<sup>1,\*</sup>, Arati Gurung<sup>2</sup>, Stefania Petra<sup>1,\*\*</sup>,  
and Christoph Schnörr<sup>1,\*</sup>

<sup>1</sup> Image and Pattern Analysis Group, University of Heidelberg, Germany  
ecaterina.bodnariuc@iwr.uni-heidelberg.de,  
{petra,schnoerr}@math.uni-heidelberg.de

<sup>2</sup> Laboratory for Aero and Hydrodynamics (3ME-P&E), Delft University of  
Technology, The Netherlands  
a.gurung@tudelft.nl

**Abstract.** We present a novel approach to detect the trajectories of particles by combining (a) adaptive dictionaries that model physically consistent spatio-temporal events, and (b) convex programming for sparse matching and trajectory detection in image sequence data. The mutual parametrization of these two components are mathematically designed so as to achieve provable convergence of the overall scheme to a fixed point. While this work is motivated by the task of estimating instantaneous vessel blood flow velocity using ultrasound image velocimetry, our contribution from the optimization point of view may be of interest also to related pattern and image analysis tasks in different application fields.

**Keywords:** motion estimation, fixed point algorithm, adaptive dictionaries, sparse representation, sparse error correction, Echo PIV.

## 1 Introduction

**Overview.** *Ultrasound Image Velocimetry (Echo PIV)* has evolved into an active research interest primarily due to its ability to measure instantaneous flow velocity and wall shear stress in a non-intrusive manner [1,2] with a wide range of applications (e.g. from arterial wall shear stress measurements for atherosclerosis-related studies to two-phase flow quantification for industrial studies such as dredging).

Currently available sensors, however, severely limit the spatial and temporal resolution of measurements. Computational cross-correlation techniques, adopted from the traditional laser-based optical PIV and used in different fields of experimental fluid mechanics [3], suffer from poor signal to noise in the reconstructed

---

\* EB and CS appreciate financial support by the German Research Foundation (DFG).

\*\* SP acknowledges financial support by the Ministry of Science, Research and Arts, Baden-Württemberg, within the Margarete von Wrangell postdoctoral lecture qualification program.

image sequences. Moreover, the established cross-correlation methods make it difficult to mathematically quantify motion information over an entire image sequence in a consistent frame-by-frame analysis of the spatio-temporal flow characteristics. As such, it becomes important, but yet challenging, to incorporate the physical principles governing the imaged fluid flow.

In this paper we present a novel approach that directly addresses these shortcomings in terms of adaptive spatio-temporal dictionaries of particle trajectories. These dictionaries are based on a basic physical model of vessel blood flow and are integrated into a standard sparse convex programming framework.

**Related Work, Contribution.** Research in connection with Echo PIV concerns (i) sensor design image reconstruction and (ii) image analysis. Since research on sensor design is rapidly evolving [4,5], we ignore this inverse modelling aspect and focus on (ii) with context to PIV wherein we derive a *mathematical abstraction of “particles”, to be understood as coefficients of a basis expansion, that discretises a realistic imaging operator in our future work.*

Echo PIV employs the standard cross-correlation technique for motion estimation [1,2]. In this paper, we propose a novel approach radically different from this standard protocol with the following objectives:

1. Any imaging operator model discretized by suitable basis functions can be incorporated later on.
2. Particle trajectories are detected by a comprehensive spatio-temporal analysis of entire image sequences in terms of dictionaries of trajectories. This copes better with noise in comparison to techniques that merely analyse subsequent image pairs. Furthermore, physical models of vessel blood flow [6,7] can be directly exploited.
3. The computational costs for the aforementioned spatio-temporal analysis are subdivided by adapting a smaller collection of dictionaries until convergence.

While the novelty of our approach is obvious from the viewpoint of Echo PIV, our main contribution from the optimization point of view concerns the *consistent integration of adaptive dictionaries* into a standard sparse convex programming framework. This is accomplished by carefully modelling the mutual interaction of dictionary parametrization and sparse convex particle matching so as to obtain a provably converging fixed point scheme. These mathematical aspects of our approach might be of interest also to related computational image and pattern analysis tasks in different application fields.

**Organization.** The application and the corresponding imaging techniques are sketched in Section 2. Section 3 details the model-based definition of dictionaries together with the variational approach for motion estimation through particle trajectory detection. Section 4 provides a convergence analysis of the adaptive variational approach. Properties of our approach are validated experimentally in Section 5.

**Basic Notation.** We set  $[n] = \{1, 2, \dots, n\}$  for  $n \in \mathbb{N}$ . Vectors are column vectors and indexed by superscripts.  $\langle x, z \rangle$  denotes the standard scalar product

in  $\mathbb{R}^n$ , and  $\|x\|_1 = \sum_{i=1}^n |x_i|$  and  $\|x\| := \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ .  $\mathbb{1} = (1, 1, \dots, 1)^\top$  denotes the one-vector whose dimension will always be clear from the context.  $\Delta_d = \{x \in \mathbb{R}_+^d : \langle \mathbb{1}, x \rangle = 1\}$  denotes the probability simplex in  $\mathbb{R}^d$ .

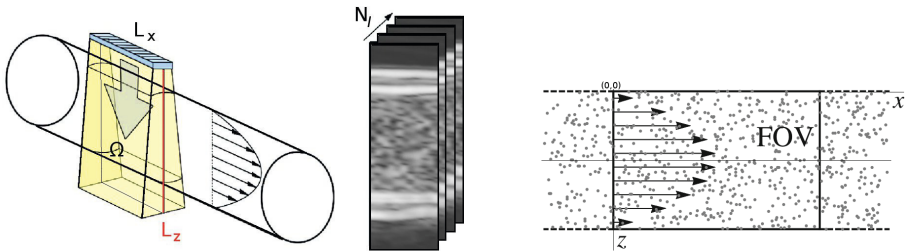
## 2 Ultrasound Imaging and Echo PIV

We briefly sketch the state-of-the-art in imaging and motion analysis in Echo PIV to highlight the novelty of our own methodological approach compared to the established computational PIV techniques.

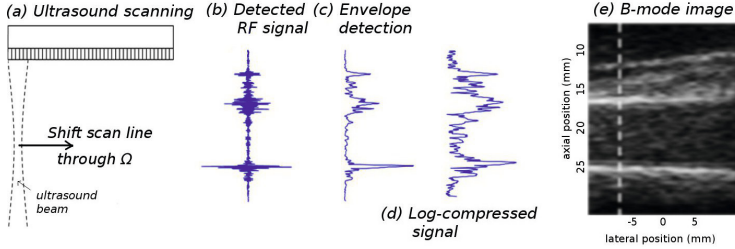
**Particle Image Velocimetry (PIV).** *PIV* is an optical method for measuring fluid flows. For the purpose of imaging, the fluid is seeded with particles that follow the flow dynamics. The region of interest is illuminated with a laser sheet and a high-speed camera takes successive images. In a subsequent step, a cross correlation technique is applied to every pair of two subsequent images and returns an estimate of the instantaneous velocity field. For a recent overview of the history of PIV techniques, we refer to [8].

**Ultrasound Microbubble Imaging.** *Echo PIV*, first introduced in [1], is a technique based on the same PIV principles. Instead of the high-speed cameras used in optical PIV, an *ultrasound transducer* is used in Echo PIV to capture tracer images with the ability to image opaque media. Another major difference to optical 2D PIV is the generation of so-called *B-mode* images, as sketched in Figures 1 and 2. These 2D images are acquired via the conventional pulse-echo technique that concatenates a series of scan lines within the field of view (FOV), as depicted in Fig. 2. This severely limits the spatio-temporal resolution of flow measurements.

One way to overcome this problem is to replace multiple line measurements by a single *plane wave* illumination of the medium [4]. *Plane wave imaging* was very recently applied to Echo PIV [5] and allows for measuring higher velocities, since the frame rate is only limited by the propagation time of the waves, rather



**Fig. 1.** A schematic representation of an Echo PIV setup. The left image, adapted from [2], overviews geometry and orientation of the transducer. Velocity is estimated from a sequence of B-mode images (middle). Flow motion is estimated from the motion of tracer particles injected in the medium (right), which follow the flow dynamics – here, a steady laminar flow.



**Fig. 2.** B-mode imaging in Echo PIV: images are not recorded as snapshots, but are usually constructed line-by-line, due to the shifting of the ultrasound beam (a). The data – RF signals (b) – can be converted (offline) to so-called B-mode images (e) by means of envelope detection (c) and log compression (d). This scanning procedure results in a blurred, smeared image due to moving particles between consecutive measurements.

than by the number of consecutive measurements necessary to obtain a single B-mode image. *This motivates us to ignore inter-line delay in our present work.*

**Motion Estimation.** Standard Echo PIV setups estimate the velocity field by matching image patterns across consecutive image pairs within the acquired image sequence, as in conventional PIV [8,9]. Such PIV methods fail to

- (i) exploit the entire spatio-temporal context of a corresponding volume of image sequence data, and
- (ii) take into account the physical prior knowledge in a mathematically more principled way.

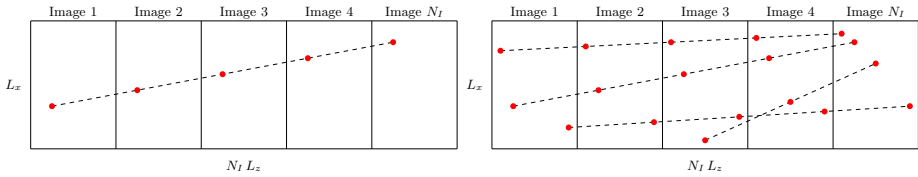
Our present work addresses both aspects for the specific setting of Echo PIV as summarized in Section 1.

### 3 Spatio-temporal Motion Model and Estimation

#### 3.1 Dictionary of Moving Particles

As mentioned in Section 2, ultrasound images of the seeded flow for Echo PIV are composed of vertical scan lines within the FOV acquired at different time steps. This scheme limits the frame rate and consequently the maximum resolvable velocity. In the present work, we propose a different acquisition protocol motivated by current research on image acquisition [4,5] in which the whole image/frame is recorded at the same point in time.

With index  $n$  we label the image of the FOV recorded at time  $\tau_n = (n - 1) \Delta t$ ,  $n \in [N_I]$ , where  $N_I$  is the total number of frames. All images have size  $L_x \times L_z$  in length units or  $l_x \times l_z$  in pixels. We introduce a 2D rectangular grid with lattice spacing  $\Delta x = L_x/l_x$ ,  $\Delta z = L_z/l_z$  in  $x$  and  $z$  respectively in the plane of FOV, induced by discrete pixel representation of images.



**Fig. 3.** Each column of the dictionary  $D$  is an image of an undersampled discrete line, and describes a possible trajectory in the  $N_I$  acquired images concatenated along the tube axis (left). Each such column depends on the discretization of  $\Omega$ , acquisition process and flow model. Here the Poiseuille flow model leads to straight lines. The input data (right) is given by all  $N_I$  frames concatenated along the tube axis. The problem is to *sparsely* match *imaged* particles to trajectories in  $D$  parametrized by the unknown maximal velocity  $v_m$ .

Below we describe how to build a flow dictionary corresponding to steady laminar flow with maximal velocity along the cylinder axis equal to  $v_m$ .

**Dictionary of a Single Velocity Profile.** The dictionary of trajectories  $D$  is a sparse matrix with binary entries  $\{0, 1\}$  and it describes the position of particles at time  $\tau_n$ ,  $n \in [N_I]$  relative to the FOV. Each column in  $D$  is associated to the trajectory of a single particle  $j$ ,  $j \in [N_P]$ , where  $N_P$  denotes the number of particles. The number of columns in  $D$  equals the number of possible trajectories. Due to the discretization, in the limit when a particle is located at all grid points, there is an upper bound for  $N_P < l_x l_z + (N_I - 1) \Delta t v_m l_x L_x / l_z$ . The number of rows in  $D$  is independent of  $v_m$  and equals  $N_I l_x l_z$ .

According to the adopted model sketched in Figure 1 (right panel), the motion of particle  $j$  with initial coordinates  $(x_1^j, z_1^j)$  at time  $\tau_1$  is governed by

$$\begin{cases} x_n^j = x_1^j + (n - 1) \Delta t v_m \left( 1 - \left( \frac{r^j}{R} \right)^2 \right), \\ z_n^j = z_1^j = \text{const.} \end{cases} \quad (1)$$

where  $r^j = |z_1^j - R|$ ,  $z_1^j \in [0, 2R]$  is the distance from the axis and  $R$  the inner radius  $R$  of the cylinder.

If at time  $\tau_n$  particle  $j$  is present in the FOV, i.e.  $x_n^j \in (0, L_x]$ , then its pixel coordinates in image  $n$  is  $(m_{x_n}^j, m_{z_n}^j)$ , where  $m_{x_n}^j = \lceil \frac{x_n^j}{\Delta x} \rceil$ ,  $m_{x_n}^j \in [l_x]$  ( $\lceil a \rceil$  is the smallest integer larger than  $a$ ) and since coordinates  $z$  remain unchanged over time we set  $z_1^j, \forall j \in [N_P]$ , to have the form

$$z_1^j = z_n^j = (m_{z_n}^j - \frac{1}{2}) \Delta z, \quad (2)$$

$m_{z_n}^j \in [l_z]$ . Further, we select the row index

$$i_n^j = (n - 1) l_x l_z + m_{z_n}^j l_x - m_{x_n}^j + 1 \quad (3)$$

and define the entries in the  $j$  column of the dictionary as

$$D_{ij} = D_{ij}(v_m) = \begin{cases} 1, & \text{if } i = i_n^j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$



We stress the fact that, with all discretization parameters fixed, *a dictionary*  $D$  of particle trajectories corresponding to a single velocity profile (1) is parametrized by the single scalar maximal velocity  $v_m$ .

The above definition implies that the number of non vanishing entries in any column  $j$  does not exceed the number of images  $N_I$ . This is consistent with the physical picture that a particle appears only once in a measured image, or it does not appear at all. We note that two columns  $D_{\bullet,j}, D_{\bullet,j'}$  will be equal if and only if the initial coordinates for two different particles are equal, i.e.  $(x_1^j, z_1^j) = (x_1^{j'}, z_1^{j'})$ . Consequently  $D$  will not contain redundant (equal) columns. Another consequence is the orthogonality of the columns of  $D$ , as formally stated next.

**Proposition 1.** *For any two columns  $D_{\bullet,j}$  and  $D_{\bullet,j'}$  in  $D$  corresponding to particles with initial coordinates  $(x_1^j, z_1^j)$  and  $(x_1^{j'}, z_1^{j'})$  we have*

$$\langle D_{\bullet,j}, D_{\bullet,j'} \rangle = 0 \iff (x_1^j, z_1^j) \neq (x_1^{j'}, z_1^{j'}). \tag{5}$$

**Proof.** We show  $\langle D_{\bullet,j}, D_{\bullet,j'} \rangle \neq 0 \iff (x_1^j, z_1^j) = (x_1^{j'}, z_1^{j'})$ .

" $\Leftarrow$ " Clear, in view of (1) and the construction of  $D$ .

" $\Rightarrow$ " Assume  $\langle D_{\bullet,j}, D_{\bullet,j'} \rangle \neq 0$ . We show that this implies  $(x_1^j, z_1^j) = (x_1^{j'}, z_1^{j'})$ . The assumption implies that there exists an index  $i_n = i_{n'}$  such that  $D_{i_n, j} = D_{i_n, j'} = 1$ , i.e. by (3)

$$n l_x l_z + m_{z_n}^j l_x - m_{x_n}^j = n' l_x l_z + m_{z_{n'}}^{j'} l_x - m_{x_{n'}}^{j'}. \tag{6}$$

From  $m_{z_n}^j = \{1, \dots, l_z\}$  and  $m_{x_n}^j = \{1, \dots, l_x\}$ , we have  $0 \leq m_{z_n}^j l_x - m_{x_n}^j \leq l_x l_z - 1$ , and similarly for  $j'$ , i.e.  $0 \leq m_{z_{n'}}^{j'} l_x - m_{x_{n'}}^{j'} \leq l_x l_z - 1$ . Dividing (6) through  $l_x l_z$ , we get

$$\underbrace{n}_{\in \mathbb{N}} + \underbrace{\frac{m_{z_n}^j l_x - m_{x_n}^j}{l_x l_z}}_{\in [0,1) \cap \mathbb{Q}} = \underbrace{n'}_{\in \mathbb{N}} + \underbrace{\frac{m_{z_{n'}}^{j'} l_x - m_{x_{n'}}^{j'}}{l_x l_z}}_{\in [0,1) \cap \mathbb{Q}} \tag{7}$$

from which we conclude  $n = n'$  and  $m_{z_n}^j l_x - m_{x_n}^j = m_{z_{n'}}^{j'} l_x - m_{x_{n'}}^{j'}$ . Rewriting the latter expression as

$$m_{z_n}^j = m_{z_{n'}}^{j'} + (m_{x_n}^j - m_{x_{n'}}^{j'})/l_x, \tag{8}$$

we infer  $m_{x_n}^j - m_{x_{n'}}^{j'} = 0$  as follows: The relation  $|m_{x_n}^j - m_{x_{n'}}^{j'}| \leq l_x - 1$ ,  $m_{z_n}^j, m_{z_{n'}}^{j'} \in \mathbb{N}$  and  $n = n'$  implies  $m_{x_n}^j = \lceil \frac{x_n^j}{\Delta x} \rceil$ . Since this equality must hold for any  $\Delta x$ , we conclude  $x_n^j = x_{n'}^{j'}$ .

As a consequence, (8) implies  $m_{z_n}^j = m_{z_{n'}}^{j'}$  and hence  $z_1^j = z_1^{j'}$  by (2). This together with (1) and  $x_n^j = x_{n'}^{j'}$  finally implies  $x_1^j = x_1^{j'}$ .  $\square$

### 3.2 Variational Motion Estimation

Given noisy measurements  $F$  of particles  $\{(x_n^j, z_n^j)\}_{j \in [N_P], n \in [N_I]}$  for a collection of  $N_I$  subsequent frames at points of time  $\tau_n = (n - 1)\Delta t$ ,  $n \in [N_I]$ , we set up an adaptive variational approach for localizing these particles in  $F$ .

To this end, we exploit the motion model (1) that describes particles' trajectories parametrized by the *unknown maximal velocity*  $v_m$  and *unknown initial coordinates*  $(x_1^j, z_1^j)$ . Aggregating potential local detections over time in this way is our approach (i) to suppress noise, (ii) to discriminate particles from each other, and (iii) to estimate the unknown velocity  $v_m$  that is the ultimate goal from the viewpoint of the application area.

We make the reasonable assumption of knowing an interval

$$v_m \in [v_{\min}, v_{\max}], \quad v_{\min} > 0 \tag{9}$$

that contains the unknown parameter  $v_m$ . Every velocity value  $v'_m \in [0, v_{\max}]$  defines a dictionary  $D(v'_m)$  by (4) that exhaustively enumerates trajectories generated by (1) with  $v_m = v'_m$ , that could have been observed in the image sequence. If we knew the true velocity  $v_m$ , we could detect trajectories in the data  $F$  by sparsely matching  $D(v_m)u$  to  $F$ , where  $u$  corresponds to a sparse indicator vector selecting active trajectories in  $D(v_m)$ .

Since  $v_m$  is not given, we have to estimate it from the data  $F$  as well. Since a single dictionary  $D(v'_m)$  is quite large, setting up a collection of dictionaries

$$D(v) := (D(v_1), D(v_2), \dots, D(v_d)), \quad 0 < v_1 < v_2 < \dots < v_d < v_{\max} \tag{10}$$

with closely spaced values  $\{v_i\}_{i \in [d]}$  is computationally infeasible. We therefore limit  $d$  to a reasonable value (see Section 5 for the setup) and *estimate  $v_m$  by an adaptive sequence of dictionaries* defined by a sequence of velocity vectors

$$D_{(k)} := D(v^{(k)}), \quad v^{(k)} = (v_1^{(k)}, \dots, v_d^{(k)})^\top \in [v_{\min}, v_{\max}]^d, \quad k \in \mathbb{N} \tag{11}$$

that localizes  $v_m \in [v_1^{(k)}, v_d^{(k)}]$  in intervals of shrinking sizes:  $|v_d^{(k)} - v_1^{(k)}| < |v_d^{(k-1)} - v_1^{(k-1)}|$ . At each iterative step  $k$ , we match trajectories and data by solving

$$u^{(k)} := \underset{u \in [0,1]^N}{\operatorname{argmin}} \|D_{(k)}u - F\|_1 + \frac{\alpha}{2} \|u\|^2 + \frac{1}{2\lambda} \|u - u^{(k-1)}\|^2, \quad \alpha > 0, \lambda > 0. \tag{12}$$

We stress that nonnegativity constraints enforce *sparse recovery* without explicit sparse regularization [10]. In order to additionally cope with *sparse outliers* we decided to use an  $\ell_1$ -based data/linear model discrepancy term, since minimizing  $\|D_{(k)}u - F\|_1$  is better suited for sparse error recovery, see [11]. Subsequently, we subdivide  $u^{(k)}$  into subvectors conforming to the structure (10) of  $D_{(k)}$ ,

$$u^{(k)} = (u^{1,(k)}, \dots, u^{d,(k)}), \tag{13}$$

and estimate  $v_m$  as convex combination of the velocity values  $v^{(k)}$  defining the current dictionary  $D_{(k)}$ ,

$$v_m^{(k)} := \sum_{i \in [d]} w_i^{(k)} v_i^{(k)} = \langle w^{(k)}, v^{(k)} \rangle, \quad w_i^{(k)} := \frac{1}{\|u^{(k)}\|_1} \|u^{i,(k)}\|_1, \quad i \in [d]. \tag{14}$$

Iteration step  $k$  is completed by updating the velocity vector

$$v^{(k+1)} = V_\tau(u^{(k)}, v^{(k)}), \quad v_i^{(k+1)} := v_m^{(k)} + \tau(v_i^{(k)} - v_m^{(k)}), \quad i \in [d], \quad (15)$$

with  $\tau \in (0, 1)$ . In the next section, it is shown that for any choice of the parameters  $\lambda > 0$  and  $\tau \in (0, 1)$ , the sequence of *non-stationary* mappings (i.e. depending on  $k$ )

$$v^{(k)} \xrightarrow{\text{Eqn. (12)}} u^{(k)} \xrightarrow{\text{Eqn. (15)}} v^{(k+1)} \quad (16)$$

is a fixed point iteration that converges to a constant vector  $v^{(\infty)} = v_m \mathbb{1}$ , that constitutes the estimate of  $v_m$ . The quality of this estimate from the applied viewpoint as outlined in Section 2, will be assessed in Section 5.

### 4 Convergence Analysis

We next show the convergence of the scheme (16) under mild conditions. The proof reveals how the scheme can be modified from the viewpoint of the intended application without compromising convergence. We describe a promising variant in the next paragraph.

**Convergence.** We write for the proximal mapping  $u^{(k-1)} \rightarrow u^{(k)}$  defined by (12)

$$u^{(k)} = P_\lambda f(u^{(k-1)}, v^{(k)}) := \operatorname{argmin}_u f(u, v^{(k)}) + \frac{1}{2\lambda} \|u - u^{(k-1)}\|^2, \quad (17a)$$

$$f(u, v^{(k)}) := \|D_{(k)}u - F\|_1 + \frac{\alpha}{2} \|u\|^2 + \delta_C(u), \quad C = [0, 1]^N, \quad (17b)$$

$$e_\lambda f(u, v^{(k)}) := \inf_w f(w, v^{(k)}) + \frac{1}{2\lambda} \|w - u\|^2, \quad (17c)$$

in order to exhibit the parametrization by  $v^{(k)}$  defining the dictionary (11). Eq. (17c) additionally introduces the Moreau envelope  $e_\lambda f$  of  $f$  [12, Def. 1.22], that we need in the proof of Prop. 2 below.

Likewise, we regard the mapping  $v^{(k)} \mapsto v^{(k+1)}$  defined by (15) as parametrized by  $u^{(k)}$ . These mutual dependencies of the sequences  $(u^{(k)})_{k \in \mathbb{N}}$  and  $(v^{(k)})_{k \in \mathbb{N}}$  and their convergence are addressed next.

**Proposition 2.** *Let the sequences  $(u^{(k)})_{k \in \mathbb{N}}, (v^{(k)})_{k \in \mathbb{N}}$  be given by (12) and (15), respectively. Suppose the mapping  $v \mapsto D(v)$  is continuous. Then, for any initializations  $v^{(0)} \in [v_{\min}, v_{\max}]^d \subset \mathbb{R}_{++}^d$  and  $u^{(0)} \in C$ , the sequence  $v^{(k)} \xrightarrow{k \rightarrow \infty} v^{(\infty)} = v_m^{(\infty)} \mathbb{1}$  converges to a constant vector as fixed point, and the sequence  $u^{(k)} \xrightarrow{k \rightarrow \infty} u^{(\infty)} = \operatorname{argmin} f(u, v^{(\infty)})$  converges to the corresponding minimizer of  $f$ .*

*Proof.* The mapping (15) reads in view of (14)

$$V_\tau(u, v) = \tau v + (1 - \tau)v_m \mathbb{1} = (\tau I + (1 - \tau)\mathbb{1}v^\top(u))v =: V_\tau(u)v. \quad (18)$$

We observe for every fixed  $u \in C$ :

- (i)  $w(u) \in \Delta_d$  and hence constant vectors  $c\mathbb{1}$ ,  $c > 0$ , constitute fixed points:  $V_\tau(u)(c\mathbb{1}) = \tau c\mathbb{1} + (1 - \tau)\langle w(u), c\mathbb{1} \rangle \mathbb{1} = c\mathbb{1}$ .
- (ii) The matrix  $V_\tau(u)$  has eigenvalues  $\tau \in (0, 1)$  with multiplicity  $d - 1$  and 1, where the constant vectors are the eigenvectors corresponding to the largest eigenvalue 1.

As a consequence,  $V_\tau$  constitutes a contraction for any non-constant vector  $v$ ,  $\|V_\tau(u, v') - V_\tau(u, v)\| < \|v' - v\|$ , independent of  $u$ . Conversely, if we fix any feasible  $v$  and consider any sequence  $u^{(k)} \rightarrow u$ , then we have  $V_\tau(u^{(k)}, v) \rightarrow V_\tau(u, v)$  due to the continuity of  $V_\tau(\cdot, v)$ .

As a consequence of these properties, a variant of Banach’s fixed point theorem [13, Prop. 1.2] asserts that the equation  $v_u = V_\tau(u, v_u)$  has exactly one positive solution in the unit sphere  $(S^{d-1} \cap [v_{\min}, v_{\max}]^d) \subset \mathbb{R}_{++}^d$  and that  $v_{u^{(k)}} \rightarrow v_u$ .

Next, we consider the mapping  $u^{(k-1)} \mapsto u^{(k)}$ , given by the proximal mapping (17), that is parametrized by  $v^{(k)}$ . We have to show convergence of the sequence of minima (17a), which is best covered by the epi(graphical)-convergence [12, Def. 7.1] of the sequence (17b) of functions  $f^{(k)} := f(\cdot, v^{(k)})$ , whose analysis simplifies due to  $f$  being proper, lower semicontinuous and (strongly) convex as follows.

By [12, Thm. 7.37], pointwise convergence  $e_\lambda f^{(k)}(u) \rightarrow e_\lambda f^{(\infty)}(u)$  of the Moreau envelopes (17c) for some  $\lambda > 0$ , which holds due to the continuity of  $v \mapsto D(v)$  by assumption, already yields epi-convergence of the sequence  $f^{(k)}$  to  $f^{(\infty)}$ . This in turn assures by [12, Thm. 7.33] convergence of the unique minima  $u^{(k)} \rightarrow u^{(\infty)}$ , where uniqueness is due to the strict convexity of the objective function of (17a), and finally  $u^{(\infty)} = \operatorname{argmin} f^{(\infty)}$ . □

As a result, the sequence  $v^{(k)}$  converges to a constant vector  $v^{(\infty)} = v_m \mathbb{1}$  in connection with the convergence of minima  $u^{(k)} \mapsto u^{(\infty)}$  that finally determines the constant  $v_m$  which is the estimate we are primarily interested in, by matching the dictionary  $D(v^{(\infty)})$  to the given data  $F$  through minimizing  $\|D(v^{(\infty)})u - F\|_1$ .

*Remark 1.* The assumption of continuity of the mapping  $v \mapsto D(v)$ , made in Prop. 2, does not strictly hold true for our current implementation described in Section 3.1, but only “up to (small) discretization effects”. Our experiments show however that this does not compromise convergence. A more refined discretization using smooth compactly supported basis functions will remove this (minor) deficiency in our future work.

**Variants of the Estimation Scheme.** The proof of Proposition 2 shows that the assertion holds for any smooth mapping

$$u^{(k)} \mapsto w^{(k)} = w(u^{(k)}) \in \Delta_d. \tag{19}$$

As a consequence, we can investigate alternatives to the mapping (14). Attractive candidates are mappings that are more sensitive to the subvector  $u^{i,(k)}$  in (13)

with maximal support  $\max_{i \in [d]} \|u^{i,(k)}\|_1$ . A natural candidate for such a smooth mapping is

$$w_i^{(k)} := \frac{1}{\sum_{j \in [d]} e^{s_j/\varepsilon}} e^{s_i/\varepsilon}, \quad s_i := \|u^{i,(k)}\|_1, \quad \varepsilon > 0, \quad i = 1, 2, \dots, d. \quad (20)$$

This results in a strictly positive vector  $w^{(k)} \in \Delta_d$  that, for  $\varepsilon \rightarrow 0$ , concentrates its mass at the component  $i \in [d]$  corresponding to  $\max_{i \in [d]} \|u^{i,(k)}\|_1$ .

We summarize the performance of this variant in numerical experiments in Section 5.

## 5 Numerical Experiments

In this section, we illustrate the performance of our approach (see Section 3 and Alg. 1 below, for a compact summary), in noisy and non-noisy environments.

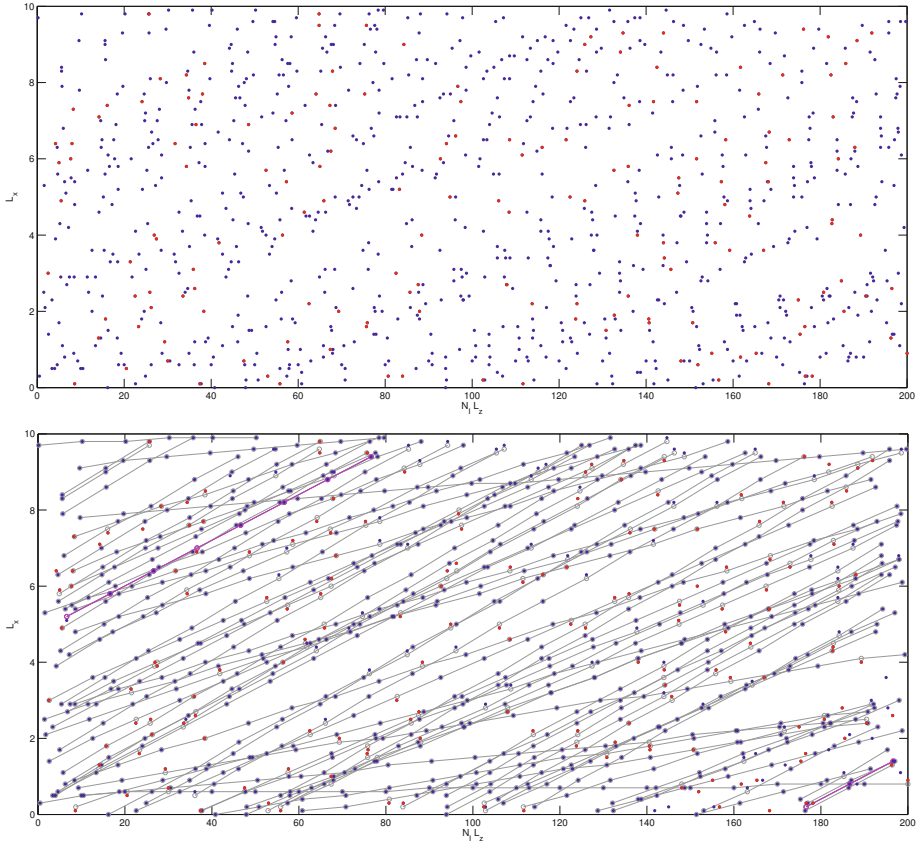
**Experimental Setup.** The experimental verification was done using data simulated as follows.

- (a) first, randomly distribute a fixed number of microbubbles in the cross section of a tube with length  $L$  (100cm) and radius  $R$  (5cm);
- (b) select an arbitrary value for  $v_m^*$  between  $v_{min} = 0.001$  and  $v_{max} = 5$ ;
- (c) calculate the position of every microbubble according to Eq. (1) at each time step  $\tau_n = (n - 1)\Delta t$ ,  $\Delta t = 0.2s$ ;
- (d) scan simultaneously the field of view  $\Omega = [0, L_x] \times [0, L_z]$  at each time  $\tau_n$  and store  $N_I = 20$  binary 2D images of size  $l_x \times l_z$  (in pixels) and microbubbles position therein.  $L_x = L_z = 10$  cm and  $l_x = l_z = 100$ ;
- (e) sort all  $N_I$  images and form the larger image  $F_{ideal} =: F$  of size  $l_x \times N_I l_z$  (see Figure 4);
- (f) add noise to mimic ghost particles or error in the position of particles in the form of outliers or perturbing positions in a random direction of random particles. The amount of noise is given by

$$\# \text{ fraction of corrupted entries} = \frac{\|F_{ideal} - F_{noise}\|_1}{2\|F_{ideal}\|_1}.$$

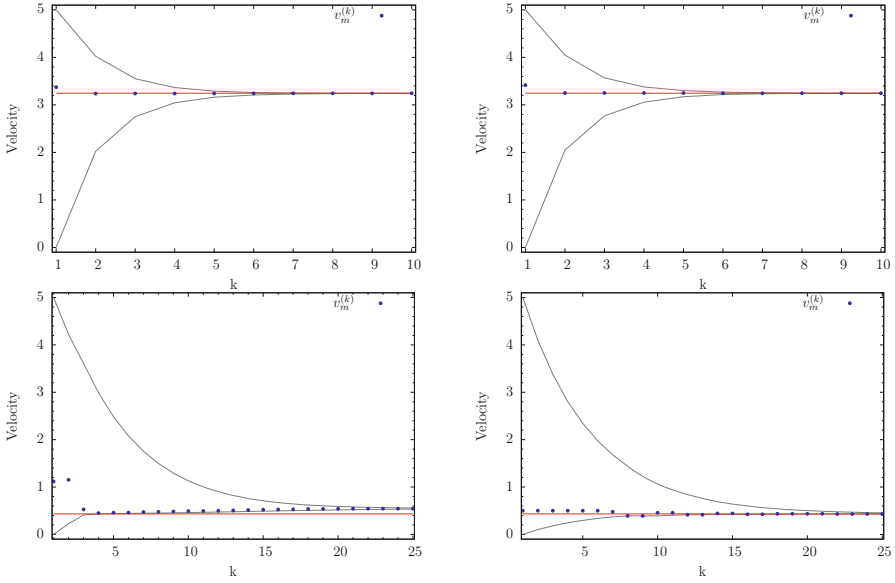
We set the particle density to 10 particles/cm. For practical reasons we precompute and store in advance dictionary blocks corresponding to a single velocity profile for all velocity values in  $[v_{min}, v_{max}]$  in steps of  $\Delta v = 0.001$ . The velocity resolution on this particular grid is of the order of  $\Delta v$ . Thus dictionary blocks  $D(v_1)$  and  $D(v_2)$  corresponding to  $v_1$  and  $v_2$  coincide if  $|v_1 - v_2| < \Delta v$ .

**Optimization.** For the two proposed variants mapping velocities (according to (14) or (20)), we run Alg. 1 below until the accuracy  $\Delta v$  was reached. The large-scale optimization task of Alg. (1) is the application of the proximal mapping and solving (12) at each iteration. To perform this task we currently use the CVX package for *disciplined convex programming* [14]. The average runtime for solving (12) is 5 minutes. Currently each  $D$  is a *highly sparse*  $(2 \cdot 10^5) \times (N_P(v_i^k) \cdot d) \approx 2 \cdot$

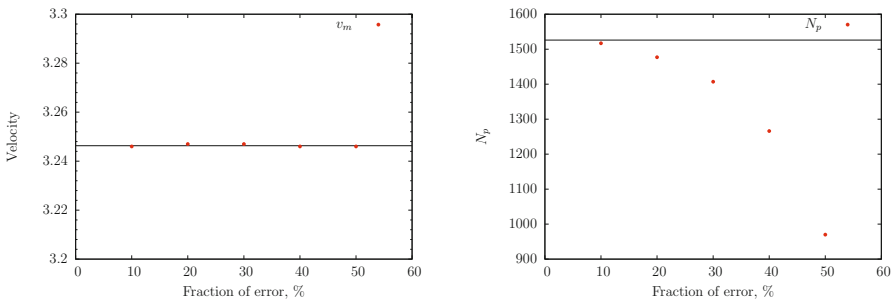


**Fig. 4.** Typical input (top) and output (bottom) of Alg. 1, but here using only 1% of the actual particle density for the purpose of visualization (better viewed in color). 20% (red dots) of input data are corrupted. All points should ideally belong to 84 unknown trajectories. Our proposed algorithm assigns microbubbles in the input frames to particle trajectories from a sparsifying dictionary. Correctly matched trajectories are displayed by thin black lines, wrong ones with magenta. The slopes of matched trajectories yield the velocity of each particle. Quantitative performance statistics for the full data sets are listed in Table 1.

$10^5 \times 10^6$  matrix, with  $d = 11$  and  $i \in [d]$ . Each  $N_P$  depends on each velocity value  $v_i^k$  and  $N_P(v_i^k) < l_x l_z + (N_I - 1) \Delta t v_i^k l_x L_x / l_z = 10^5 + 38 v_i^k$ . For processing real data a dedicated numerical optimization algorithm is necessary as CVX cannot handle much larger problem sizes. We emphasize that by ignoring the quadratic terms in (12) the problem can be recast as a linear program. Thus (12) can be seen as a perturbed linear program. Our future work from the algorithmic point of view will exploit this fact along with the structure and sparsity of  $D$  consisting of  $d$  building blocks having each orthogonal columns due to Proposition 1.



**Fig. 5.** Convergence performance of the fixed point Alg. 1 and its two variants for 20% noise, for *large* ( $v_m^* = 3.2463$ , top row) and *small* true (unknown) velocity ( $v_m^* = 0.4321$ , bottom row). Both variants of the algorithm for estimating  $v_m^*$  converged in 10 (top) and 25 (bottom) iterations. However, computing the weights  $w_i$  according to (20) based on the softmax function – *softmax-weights* – (right) leads to a more accurate estimate of  $v_m^*$  than computing weights according to (14) –  $\ell_1$ -weights – (left). Further numerical values are given in Table 1 based on averaged results over 20 runs.



**Fig. 6.** Estimating the velocity  $v_m^*$  via Alg. 1 is robust (left) to corrupting a large fraction of the input data, although the fraction of correctly detected trajectories decreases (right). This fraction suffices to define a “correct” dictionary  $D(v^{(k)})$  due to the convergence of  $v^{(k)}$  to a uniform vector  $v_m \mathbf{1}$ . Results are consistent for different values of  $\tau \in [0.4, 0.8]$ ,  $\tau \in [0.2, 0.4]$  and  $\varepsilon \in \{50, 100, 150, 200\}$ .

---

**Algorithm 1.** Fixed Point Algorithm with two variants of mapping velocities according to (14) or (20).

---

**Data:** concatenated frames  $F$ ,  $d \in \mathbb{N}$  initial estimates for velocity profiles  
 $v^{(1)} = (v_1^{(1)}, \dots, v_d^{(1)})$ , parameters  $\Delta v > 0$ ,  $\lambda > 0$ ,  $\alpha > 0$ ,  $\varepsilon > 0$ ,  $\tau \in (0, 1)$

**Result:**  $v_m, N_p$

$k = 1$  ;

**while**  $|v_d^{(k)} - v_1^{(k)}| < \Delta v$  **do**

$D_{(k)} = (D(v_1^{(k)}), D(v_1^{(k)}), \dots, D(v_d^{(k)}))$ ;

$u^{(k)} = \arg \min_{u \in [0,1]} \|D_{(k)} u - F\|_1 + \frac{\alpha}{2} \|u\|^2 + \frac{1}{2\lambda} \|u - u^{(k-1)}\|^2$  ;

Compute weights from (14) / (20) ;

$\forall j \in [d]: w_j^{(k)} = \frac{s_j}{\|u^{(k)}\|_1}$ ,  $s_j := \|u^{j,(k)}\|_1$  /  $w_j^{(k)} := \frac{1}{\sum_{\ell \in [d]} e^{s_\ell/\varepsilon}} e^{s_j/\varepsilon}$ ;

$v_m^{(k)} = \sum_{i \in [d]} w_i^{(k)} v_i^{(k)}$ ;

$\forall j \in [d]: v_j^{(k+1)} = v_m^{(k)} + \tau(v_j^{(k)} - v_m^{(k)})$ ;

$k = k + 1$ ;

$v_m = v_m^{(k)}$ ,  $N_p = \|u^{(k)}\|_0$ ;

---

**Results and Discussion.** Fig. 4 illustrates the detection and particle trajectories after convergence to the fixed point according to Prop. 2. The convergence behavior is depicted by Fig. 5 along with a discussion in the caption. Finally Fig. 6 demonstrates a remarkable robustness of our approach against data noise over a wide range of values of the parameters  $\tau \in (0, 1)$ ,  $\lambda > 0$  and  $\varepsilon$  in (20), due to the aggregation of all information over the entire spatio-temporal volume.

**Table 1.** Estimated velocity and number of particles for ideal and noise data. The velocity value to be estimated is  $v_m^*$ . The number of true trajectories is  $N_p^*$ . We averaged results over 20 runs. Velocity estimates are stable against noise, and the results reveal better estimates for the softmax-weights in the case of small velocities.

$v_m^* = 3.2463$ ; $N_p^* = 1526$ ; $\tau = 0.4$							
		0 %		10 %		20 %	
		$v_m$	$N_p$	$v_m$	$N_p$	$v_m$	$N_p$
$\ell_1$ -weights		$3.2437 \pm 0.003$	1526	$3.2438 \pm 0.0003$	$1513 \pm 3$	$3.2437 \pm 0.005$	$1478 \pm 8$
softmax-weights		$3.2450 \pm 0.006$	1526	$3.2456 \pm 0.007$	$1519 \pm 3$	$3.2460 \pm 0.0006$	$1493 \pm 5$
$v_m^* = 0.4321$ ; $N_p^* = 1035$ ; $\tau = 0.8$							
		0 %		10 %		20 %	
		$v_m$	$N_p$	$v_m$	$N_p$	$v_m$	$N_p$
$\ell_1$ -weights		$0.4416 \pm 0.016$	1031	$0.4688 \pm 0.0037$	$754 \pm 11$	$0.5291 \pm 0.0227$	$360 \pm 64$
softmax-weights		$0.4300 \pm 0.020$	1035	$0.4296 \pm 0.0007$	$1032 \pm 2$	$0.4299 \pm 0.0008$	$731 \pm 24$

## 6 Conclusion

We have reformulated the velocity estimation problem for a steady laminar flow via Echo PIV as a sparse and global spatio-temporal estimation problem, using



a physical flow model. The input data was the whole image sequence assumed to be well approximated by the sum of few elements from a flow dictionary. Since the dictionary was parametrized by the unknown velocity profile, we updated the dictionary in each iteration, thereby refining the unknown quantity. We showed convergence to a fixed point of the overall scheme under weak assumptions to a sparsifying dictionary that robustly estimated velocity even in the presence of high levels of noise. Numerical examples demonstrated this robustness, convergence and estimation accuracy of our approach.

Further work will concentrate on adapting the dictionary using more general physical fluid flow models, and incorporating models of the real imaging sensor with proper discretization.

## References

1. Kim, H., Hertzberg, J., Shandas, R.: Development and Validation of Echo PIV. *Exp. Fluids* 36(3), 455–462 (2004)
2. Poelma, C., van der Mijle, R.M.E., Mari, J.M., Tang, M.X., Weinberg, P.D., Westerweel, J.: Ultrasound Imaging Velocimetry: Toward Reliable Wall Shear Stress Measurements. *European Journal of Mechanics - B/Fluids* 35, 70–75 (2012)
3. Raffel, M., Willert, C., Wereley, S., Kompenhans, J.: *Particle Image Velocimetry – A Practical Guide*. Springer (2007)
4. Schiffner, M.F., Schmitz, G.: Fast Image Acquisition in Pulse-Echo Ultrasound Imaging using Compressed Sensing. In: 2012 IEEE International Ultrasonics Symposium (IUS), pp. 1944–1947. IEEE (2012)
5. Rodriguez, S., Jacob, X., Gibiat, V.: Plane Wave Echo Particle Image Velocimetry. *Proceedings of Meetings of Acoustics, POMA 19* (2013)
6. Womersley, J.R.: Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *J. Physiol.* 127, 553–563 (1955)
7. Suter, S., Skalak, R.: The History of Poiseuille’s Law. *Ann. Rev. Fluid Mech.* 25, 1–19 (1993)
8. Adrian, R.J.: Twenty Years of Particle Image Velocimetry. *Experiments in Fluids* 39(2), 159–169 (2005)
9. Westerweel, J.: Fundamentals of Digital Particle Image Velocimetry. *Measurement Science and Technology* 8(12), 1379–1392 (1997)
10. Slawski, M., Hein, M.: Sparse Recovery by Thresholded Non-Negative Least Squares. In: *Proc. NIPS*, pp. 1926–1934 (2011)
11. Candès, E.J., Tao, T.: Decoding by Linear Programming. *IEEE Transactions on Information Theory* 51(12), 4203–4215 (2005)
12. Rockafellar, R., Wets, R.J.B.: *Variational Analysis*, 2nd edn. Springer (2009)
13. Zeidler, E.: *Nonlinear Functional Analysis and its Applications: Fixed Point Theorems*, vol. I. Springer (1993)
14. Grant, M., Boyd, S.: *CVX: Matlab Software for Disciplined Convex Programming*, version 2.1. (March 2014), <http://cvxr.com/cvx>

# Point Sets Matching by Feature-Aware Mixture Point Matching Algorithm

Kun Sun<sup>1</sup>, Peiran Li<sup>1</sup>, Wenbing Tao<sup>1,\*</sup>, and Liman Liu<sup>2</sup>

<sup>1</sup> National Key Laboratory of Science and Technology on Multi-spectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

wenbingtao@hust.edu.cn

<sup>2</sup> School of Biomedical Engineering, South-Central University for Nationalities, Wuhan 430074, China

**Abstract.** In this article we propose a new method to find matches between two images, which is based on a framework similar to the Mixture Point Matching (MPM) algorithm. The main contribution is that both feature and spatial information are considered. We treat one point set as the centroid of the Gaussian Mixture Model (GMM) and the other point set as the data. Different from traditional methods, we propose to assign each GMM component a different weight according to the feature matching score. In this way the feature information is introduced as a reasonable prior to guide the matching, and the spatial transformation offers a global constraint so that local ambiguity can be alleviated. Experiments on real data show that the proposed method is not only robust to outliers, deformation and rotation, but also can acquire the most matches while preserving high precision.

**Keywords:** image matching, Gaussian Mixture Model, feature information, spatial arrangement.

## 1 Introduction

Finding corresponding points between two images is one of the fundamental problems in computer vision and is a key ingredient in a wide range of applications. However, due to the differences between images such as illumination, view point, occlusion, and scaling, the matching results are either too sparse or with too many mismatches. In this paper we focus on finding as more correct matches as possible while preserving satisfactory precision.

### 1.1 Literature Review and Problems

Among all kinds of matching methods, three classes deserve to be mentioned: the *feature descriptor* based methods, the *spatial arrangement* based methods and the methods considering *both* of them.

---

\* Corresponding author.

The feature descriptor based methods build a high dimensional descriptor for each detected feature point in its local neighborhood. Lowe[12] proposed a scale invariant feature transform(SIFT), which combines a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions. Bay et al. [2] proposed a much faster descriptor “SURF” by relying on integral images and Hessian-matrix based detector. The BRIEF descriptor [3] is a n-dimensional binary bitstring computed from pairwise intensity comparison and based on it a new oriented descriptor called ORB is defined by Rublee et al [15]. Hauagge and Snavely [8] designed a specific descriptor for matching symmetric images. Mikolajczyk and Schmid [13] recently evaluated a variety of approaches and concluded the SIFT based features perform best. However, although a lot of work has been done to improve the performance of descriptors, the results are either too sparse or with unavoidable outliers.

The second class of methods solve the matching problem by point sets registration, in which two best aligned points denote a match. Recently a popular view treats the alignment of two point sets as a MAP problem of the Gaussian Mixture Model(GMM). The Mixture Point Matching (MPM) algorithm [9] and the Robust Point Matching (RPM) algorithm [6] are two early algorithms that use the GMM representation explicitly and implicitly. The CPD algorithm [14] directly modeled one point set by the GMM and the other set as the data generated by this model. Jian and Vemuri [10] proposed to model both point sets by the GMM and then minimize their discrepancy. In spite of the success in point sets registration, the methods mentioned above are seldom used in image matching. This is because: 1) an abundant of outliers exist in the initial features, 2) the transformation between two image feature point sets are complex due to the projection of the scene at different depth to the image plane, especially in wide baseline cases.

The third class of methods simultaneously take both feature and spatial information into consideration. Among them graph matching is a hot topic. Leordeanu and Hebert [11] build an adjacent matrix whose nodes represent possible correspondences and edges denote pairwise agreement between them. The correct correspondences were recovered according to the principal eigenvector of the adjacent matrix. Cho and Lee [5] proposed a novel progressive framework which combines probabilistic progression of graphs with matching of graphs. Based on the current graph matching result, the algorithm explores the space of graphs beyond the current graphs. However, the application of graph matching methods is limited due to high computational complexity. Other works utilize both kinds of information by embedding the image coordinates and feature descriptors into a unified subspace. A pairwise matching algorithm PW [17] was proposed in which a spectral decomposition for the affinity matrix in the subspace is adopted to find matches. This work was then improved by Hamid et al. [7], in which random projection is used to approximate subspace learning and matches with high confidence are used to guide the procedure for dense matching. However, the matching in the learned subspace is to find the nearest neighbour, which may also produce mismatches.

## 1.2 Basic Idea for the Solution

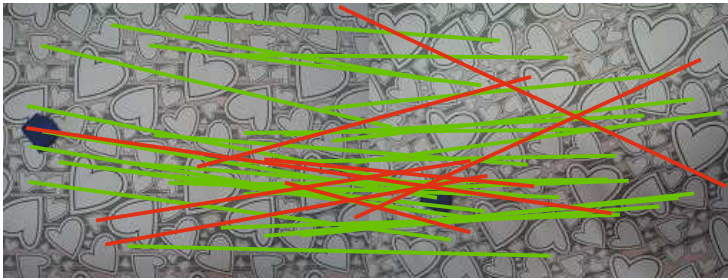
One drawback of the traditional GMM based method is that all the GMM components are assigned the same weights, which means that one data point could be matched to each of the model points with equal chance. It suffers from the following cases when matching feature points extracted from images: 1) When the point set contains a high ratio of outliers. In the context of image matching, the initial features contain a large portion of outliers, which can easily degrade the performance. This is especially severe for wide baseline cases. 2) It cannot handle large geometry changing such as large rotation, which is common in multi-view images. This is because the transformation corresponding to a large rotation will result in a large smoothness penalization. As a result, it will return a seemingly smooth transformation but the correspondence is totally wrong. 3) The shape of the point set is flat or symmetric. The difficulty for aligning this kind of point set is the uncertainty of the transformation. Since several transformations could spatially align the point sets, it prefers a simpler and smoother transformation, which may not be the most appropriate.

In this paper, we propose a new method for image feature points matching. Our main contribution is to take both feature similarity and spatial arrangement into consideration. To achieve this, we model one point set by the GMM but assign each GMM component a different weight. Specifically, for a given data point, we compute each GMM component a weight according to the feature matching score between the data point and the model point corresponding to the component. By doing so we are using the feature information to guide the spatial alignment between the point sets. The motivations for doing this are:

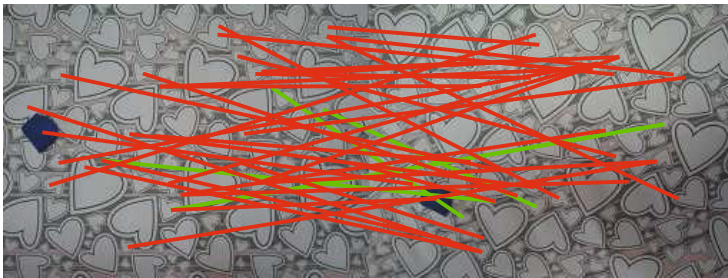
1. If two points are similar in the feature space, they are more likely to form a correct match. So the GMM component should be given a larger weight if its corresponding model point has higher feature matching score with the data point.
2. Feature similarity constraint, which provides a reasonable prior for the alignment, together with the spatial smoothness constraint are encoded in a unified model. As a result, the performance is enhanced.

We will develop our method in the framework similar to the MPM algorithm [9], in which the Deterministic Annealing technique and the EM algorithm are used. The main difference is that [9] considers only the spatial arrangement and uses the same weights for all the GMM components, while our method integrates the feature information into this framework by assigning different weights to the GMM. The Thin Plate Spline(TPS) is used as the transformation model. This is because that even though the scene is rigid, the transformation between its projections on two image planes will be modeled by a non-rigid formulation when depth discontinuity exists.

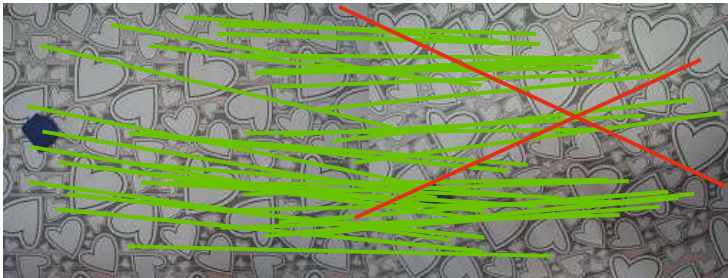
Fig. 1 is a simple example of our method. We consider two images of a scene with repeated patterns. SIFT key points and their descriptors are extracted. We then match them using (a) SIFT [18], (b) RPM [6] and (c) the proposed algorithm. As the recurring patterns produce many local similar regions, the local feature



TP: 27 TN: 8  
(a) SIFT [18]



TP: 4 TN: 70  
(b) RPM [6]



TP: 50 TN: 2  
(c) Ours

**Fig. 1.** A simple example of our method. From top to bottom: matching using SIFT [18], RPM [6] and our method. True Positives(TP) are in green and True Negatives(TN) are in red.

descriptor based matching method such as SIFT will not work well. RPM is unable to find correct matches due to large geometry changes and outliers in the original feature points. Our method not only finds the most correct matches, but also acquires satisfactory precision. This shows that our thinking of using feature similarity to guide the matching procedure while imposing spatial arrangement constraint is feasible, and can enhance the result.

The remainder of this paper is organized as follows: in Sect. 2 we give our proposed Gaussian Mixture Model and formulate the matching task as a MAP problem. In Sect. 3 we solve the problem using the framework similar to [9], in which Deterministic Annealing and the EM algorithm are used. After this, we summarize the algorithm and give the detailed parameters setting in Sect. 4. Section 5 is the experimental part. Following the robustness test and the experimental analysis of parameters is the evaluation on real images. Finally we conclude in Sect. 6.

## 2 The Probabilistic Formulation of the Matching Task

Suppose we have two point sets  $\mathbf{X} = \{(\mathbf{x}_i, \mathbf{g}_i) | \mathbf{x}_i \in \mathbb{R}^2, \mathbf{g}_i \in \mathbb{R}^D\}_{i=1}^M$  and  $\mathbf{Y} = \{(\mathbf{y}_j, \mathbf{h}_j) | \mathbf{y}_j \in \mathbb{R}^2, \mathbf{h}_j \in \mathbb{R}^D\}_{j=1}^N$ . The former in the two-tuples is the 2-Dimension image coordinate and the latter is its corresponding N-Dimension feature descriptor.  $M$  and  $N$  are the number of points in each point set, respectively. Our goal is to learn a transformation  $\mathbf{f}$  that can best align  $\mathbf{f}(\mathbf{X})$  and  $\mathbf{Y}$  and then to find matches. To achieve this, we model points from  $\mathbf{X}$  by the Gaussian Mixture Model so that the probability of a data point from  $\mathbf{Y}$  is

$$p(\mathbf{y}_j) = \sum_{i=1}^M \mathbf{v}_{ij} \mathcal{N}(\mathbf{y}_j; \mathbf{x}_i, \sigma^2, \mathbf{f}), \quad (1)$$

where

$$\mathcal{N}(\mathbf{y}_j; \mathbf{x}_i, \sigma^2, \mathbf{f}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{y}_j - \mathbf{f}(\mathbf{x}_i)\|^2}{2\sigma^2}\right) \quad (2)$$

is the component of the GMM and  $\mathbf{v}_{ij}$  is the weight of each component. In order to account for outliers in the point set, we introduce another term

$$\mathcal{N}(\mathbf{y}_j; \mathbf{x}_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{\|\mathbf{y}_j - \mathbf{x}_0\|^2}{2\sigma_0^2}\right) \quad (3)$$

to (1). Here  $\mathbf{x}_0 = \frac{1}{N} \sum_{j=1}^N \mathbf{y}_j$  and  $\sigma_0^2 = \max \|\mathbf{y}_a - \mathbf{y}_b\|^2$  are the center and covariance of the outlier component, respectively. Then (1) is extended to the following form

$$p(\mathbf{y}_j) = (1 - \theta) \sum_{i=1}^M \mathbf{v}_{ij} \mathcal{N}(\mathbf{y}_j; \mathbf{x}_i, \sigma^2, \mathbf{f}) + \theta \mathcal{N}(\mathbf{y}_j; \mathbf{x}_0, \sigma_0^2), \quad (4)$$

in which  $\theta$  is the ratio of outliers that we assume the point set may contain.

We compute  $\mathbf{v}_{ij}$  according to the feature matching score between  $\mathbf{x}_i$  and  $\mathbf{y}_j$ . The intuition is that if a model point and a data point are similar in the feature space, they are more likely to form a correct match. So the component represented by this model point should be assigned a larger weight. Denote  $\delta(\cdot, \cdot)$  as a metric that measures the pairing score between two features, then  $\mathbf{v}_{ij}$  has the following form:

$$\mathbf{v}_{ij} = \delta(\mathbf{g}_i, \mathbf{h}_j). \quad (5)$$

Suppose  $\mathbf{V}$  is a  $M \times N$  matrix with elements  $\mathbf{v}_{ij}$ . Then the  $j^{\text{th}}$  column of  $\mathbf{V}$  is the weight vector of the GMM describing  $\mathbf{y}_j$ . There are many candidates for the choice of  $\delta$ . In our paper, we leverage the SLH algorithm [16] to compute  $\mathbf{V}$ . Specifically, we compute a matrix  $\mathbf{G}$  with elements  $\mathbf{G}_{ij} = \exp(-\frac{\|\mathbf{g}_i - \mathbf{h}_j\|^2}{2\beta^2})$  and then get its singular value decomposition  $\mathbf{G} = \mathbf{T}\mathbf{D}\mathbf{U}$ , where  $\mathbf{D}$  is a non-negative diagonal matrix. We convert  $\mathbf{D}$  into another matrix  $\mathbf{E}$  by replacing its diagonal elements by one. The matrix  $\mathbf{V}$  is then computed from  $\mathbf{V} = \mathbf{T}\mathbf{E}\mathbf{U}$ . However,  $\mathbf{V}$  can not be directly used as the weight in our application since it contains negative elements. To tackle this we simply set all the negative values in  $\mathbf{V}$  to zero and normalize each column. Note that  $\mathbf{G}$  can be directly used as  $\delta$  in our method, but the performance may be dropped. This is because that considering only the ‘‘proximity’’ principle in  $\mathbf{G}$  may lead to false matches. The SLH algorithm can refine the matching result and provide more reliable guidance by taking both ‘‘proximity’’ principle and ‘‘exclusion’’ principle into consideration. So in this paper we use the latter instead.

Suppose the data points are independent identically distributed, then the joint probability distribution of the whole data set  $\mathbf{Y}$  is

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{f}, \sigma^2) = \prod_{j=1}^N p(\mathbf{y}_j). \quad (6)$$

Eq. (6) is also known as the likelihood function. Denoting the regularization over the transformation  $\mathbf{f}$  as  $\|L\mathbf{f}\|^2$ , where  $L$  is an operator that extracts the high frequency part of the function, we can take the prior with the form

$$p(\mathbf{f}) = \exp(-\frac{\lambda}{2}\|L\mathbf{f}\|^2). \quad (7)$$

Thus according to the Bayes rule the posterior probability is

$$p(\mathbf{f}|\mathbf{Y}, \mathbf{X}, \sigma^2) \propto p(\mathbf{f})p(\mathbf{Y}|\mathbf{X}, \mathbf{f}, \sigma^2). \quad (8)$$

### 3 The Solution Based on EM and Deterministic Annealing

As is known to all, maximizing (8) is equivalent to minimizing the following negative logarithm energy function

$$E_1(\mathbf{f}, \sigma^2) = -\log p(\mathbf{Y}|\mathbf{X}, \mathbf{f}, \sigma^2) - \log p(\mathbf{f}). \quad (9)$$

However, (9) is difficult to solve because it does not offer a closed form solution for the parameters. The EM algorithm is an elegant algorithm to solve the problem in (9). The EM algorithm alternates between two steps: the E-step and the M-step. In the E-step, the correspondences are estimated based on current parameters, and in the M-step, the parameters are updated according to current correspondences. On the other hand, as Chui pointed in [9], including extra variables as free parameters can result in more local minimum and make the optimization harder. So we leverage the Deterministic Annealing to make our method robust and insensitive to initialization. Specifically, we replace the parameter  $\sigma^2$  with a newly introduced parameter  $T$ , and gradually reduce it in the matching process. As a result, we aim to minimize the following energy function:

$$E_2(\mathbf{P}, \mathbf{f}) = \lambda T \|\mathbf{L}\mathbf{f}\|^2 + \sum_{j=1}^N \sum_{i=1}^M p_{ij} \|\mathbf{y}_j - \mathbf{f}(\mathbf{x}_i)\|^2 + T \log T \sum_{j=1}^N \sum_{i=1}^M p_{ij} + T \sum_{j=1}^N \sum_{i=1}^M p_{ij} \log p_{ij}, \tag{10}$$

where  $\mathbf{P}$  is a matrix with elements  $p_{ij}$  and  $T$ , also called “temperature”, is a newly introduced parameter in place of  $\sigma^2$ .

**E-step:** in the E-step, a probability matrix  $\mathbf{P}$  is estimated. Each of its element is the posterior of the GMM component computed from

$$p_{ij} = \frac{\mathbf{v}_{ij} \mathcal{N}(\mathbf{y}_j; \mathbf{x}_i, T, \mathbf{f})}{\sum_{k=1}^M \mathbf{v}_{kj} \mathcal{N}(\mathbf{y}_j; \mathbf{x}_k, T, \mathbf{f}) + c_0}, \tag{11}$$

where  $c_0 = \frac{\theta}{1-\theta} \mathcal{N}(\mathbf{y}_j; \mathbf{x}_0, \sigma_0^2)$  is a constant.  $p_{ij}$  in (11) indicates to what extent a data point  $\mathbf{y}_j$  corresponds to a model point  $\mathbf{x}_i$ . Then each column of  $\mathbf{P}$  is the matching score vector of a certain data point to all the GMM components. A property of  $\mathbf{P}$  is that it implicitly tells the correspondence. Note that  $p_{ij}$  in (8) is modulated by the introduced  $\mathbf{v}_{ij}$ .  $p_{ij}$  will take a large value only when  $\mathbf{x}_i$  and  $\mathbf{y}_j$  are not only spatially close but also similar in the feature space.

**M-step:** in the M-step, the transformation function  $\mathbf{f}$  is updated based on  $\mathbf{P}$  estimated in the E-step by minimizing

$$E_3(\mathbf{f}) = \lambda T \|\mathbf{L}\mathbf{f}\|^2 + \sum_{i=1}^M \|\mathbf{z}_i - \mathbf{f}(\mathbf{x}_i)\|^2, \tag{12}$$

where  $\mathbf{z}_i = \sum_{j=1}^N p_{ij} \mathbf{y}_j$  can be seen as the new estimated position of the data.

We use the Thin Plate Spline(TPS) to parameterize the non-rigid transformation  $\mathbf{f}$  as

$$\mathbf{f}(\mathbf{x}_i; d, \omega) = \mathbf{x}_i d + \phi(\mathbf{x}_i) \omega, \tag{13}$$

where  $d$  and  $\omega$  are the affine and non-affine transformation matrix, respectively. Each model point  $\mathbf{x}_i$  is represented by its homogenous coordinate.  $\phi(\mathbf{x}_i)$  is a vector with elements computed from the kernel  $\phi_a(\mathbf{x}_i) = \|\mathbf{x}_a - \mathbf{x}_i\|^2 \log \|\mathbf{x}_a - \mathbf{x}_i\|$ .



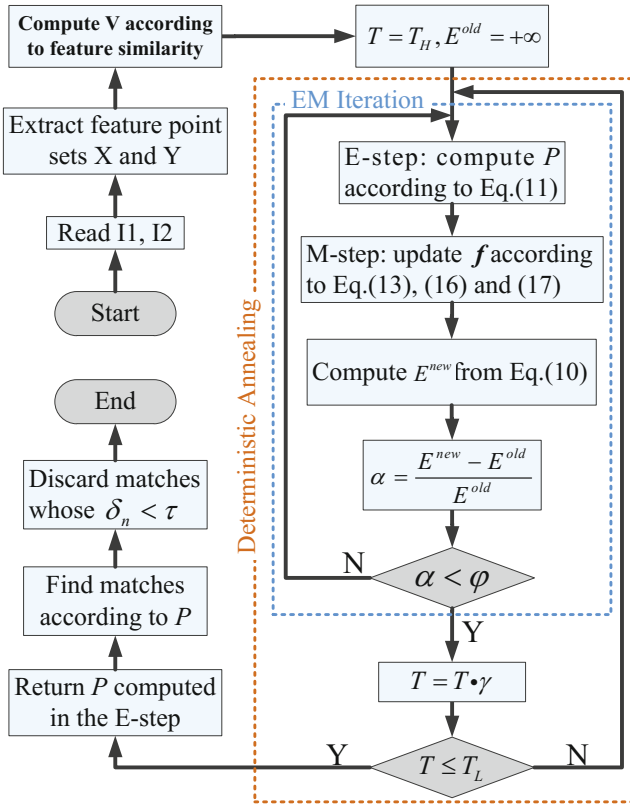


Fig. 2. The flowchart of our algorithm

Thus (13) becomes

$$E_4(\mathbf{f}) = \lambda \text{trace}(\omega^T \Phi \omega) + \|\mathbf{Z} - \mathbf{X}d - \Phi \omega\|^2, \tag{14}$$

where  $\mathbf{Z}$ ,  $\mathbf{X}$  and  $\Phi$  are the concatenated matrix form of  $\mathbf{z}_i$ ,  $\mathbf{x}_i$  and  $\phi(\mathbf{x}_i)$ , respectively. To solve  $\omega$  and  $d$ , we first apply the QR decomposition to  $\mathbf{X}$

$$\mathbf{X} = [Q_1 Q_2] \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{15}$$

and substitute it to (14). Then the optimal solutions of  $\omega$  and  $d$  are

$$\omega = Q_2(Q_2^T \Phi Q_2 + \lambda \mathbf{I}_{M-3})^{-1} Q_2^T \mathbf{Z} \tag{16}$$

and

$$d = R^{-1}(Q_1^T \mathbf{X} - \Phi \omega). \tag{17}$$

We finish the above matching process and get the probability matrix  $\mathbf{P}$  after convergence. Suppose the maximum for each column of  $\mathbf{P}$  is stored in a vector  $\{\delta_n | \delta_n = p_{mn}, n = 1 \dots N, m \in [1, M]\}$ , which means that the maximum of the

$n^{th}$  column is its  $m^{th}$  element. Then we denote  $\mathbf{x}_m$  and  $\mathbf{y}_n$  as a match. To this end,  $\delta_n$  can be seen as the matching confidence, which can be obtained at the same time we denote a match. Besides, we can also tell a match is “strong” if its matching confidence is high and “weak” otherwise. To achieve more robust results, we introduce a threshold parameter  $\tau$  on the matching confidence  $\delta_n$  and discard the matches with confidence lower than  $\tau$ .

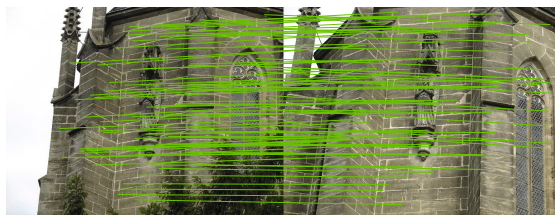
## 4 Algorithm and Implementation Details

### 4.1 Summary of Our Proposed Algorithm

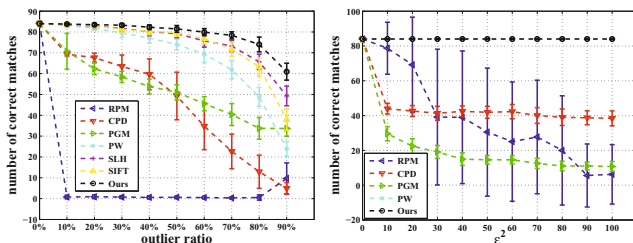
The flowchart of our method is shown in Fig. 2. At the beginning of our algorithm, two images  $I_1$  and  $I_2$  are loaded from the disk. Feature points are then extracted, including their image coordinates and feature descriptors. Next the matrix  $\mathbf{V}$  is computed using the SLH algorithm. Each column of  $\mathbf{V}$  is the weight vector of the GMM describing a certain data point. We set the starting temperature to a relatively high value  $T_H$  and initialize the energy of EM as infinity. The main part of our algorithm consists of two nested loops. The inner loop is the EM iteration, during which the temperature  $T$  is constant. In the E-step, the probability matrix  $\mathbf{P}$  is estimated from (11) and in the M-step the transformation model  $\mathbf{f}$  is updated according to (13), (16), and (17). After each iteration we compute the new EM energy  $E^{new}$ . The change between  $E^{new}$  and  $E^{old}$  is indicated by  $\alpha$ . If  $\alpha$  is smaller than a threshold  $\varphi$ , the energy tends to be stable and the EM algorithm converges. Otherwise the E- and M-steps are repeated. The outer loop is the deterministic annealing iteration. Different from the inner loop with a constant temperature, the outer loop gradually decreases  $T$  by a rate  $\gamma$  after EM converges. The annealing procedure will end if  $T$  reaches a relatively lower temperature  $T_L$ . Then our algorithm returns the probability matrix  $\mathbf{P}$  computed in the E-step and correspondences are built by finding the maximum for each column of  $\mathbf{P}$ . Finally, we discard matches whose matching confidence  $\delta_n$  is smaller than a threshold  $\tau$ .

### 4.2 Analysis and Setting of the Parameters

We then explain our parameter setting. We follow the setting of  $\beta$  in [16]. In (4) the parameter  $\theta$  is the ratio of outliers that we assume the point set would contain. Since this ratio may be quite different for different instances and is difficult to know in advance, we set it to 0.5, which assumes an equal split of inliers and outliers. The starting temperature  $T_H$ , the ending temperature  $T_L$  and the annealing rate  $\gamma$  are set as described in [9]. We set  $T_H = \sigma_0^2 = \max \|\mathbf{y}_a - \mathbf{y}_b\|^2$ , where  $a, b \in [1, \dots, N]$  and  $T_L = \frac{1}{N} \sum_{s=1}^N \min \|\mathbf{y}_s - \mathbf{y}_t\|^2$ , where  $t \in [1, \dots, N]$  and  $t \neq s$ . This means at the beginning we allow a data point to match to any model point with even probability, and at the final stage a data point should precisely match to only one model point. The annealing rate  $\gamma$  is set to 0.93.  $\lambda$  is a very important parameter that controls the smoothness of  $\mathbf{f}$ . A large  $\lambda$  will be



(a) The “Zleby4” pair and matches selected.



(b) Outliers (c) deformation

**Fig. 3.** Robustness to outliers and deformation tested on the image pair “Zleby4” [4]. Outliers are randomly selected points and deformation perturbations are generated by  $\mathcal{N}(0, \epsilon^2)$ . The results show that our method has strong ability to recover matches with high ratio of outliers or large deformation.

less tolerant to the perturbation of the transformation while a small  $\lambda$  may lead to a disordered motion. We set  $\lambda = 0.1 * N$  in our experiments. Another important parameter is  $\tau$ , which has close relation with the precision and amount of matches. We set its value to 0.5 in our experiments. In our implementation, we do not use  $\varphi$  to control the convergence of EM. Instead, we fix the iteration number of EM to 5 for each temperature. Since we search for correspondences in a global to local way, so this simplification will not lead to significant performance reduction.

## 5 Experiment Results

### 5.1 Robustness Test of Our Algorithm

We first carry out experiments to show that our algorithm is robust to outliers and deformation. Given the image pair “Zleby4” [4] in Fig. 3(a), we manually mark 83 points on each image so that they compose 83 one-to-one true positive matches, which are treated as the ground truth. After computing the SIFT feature descriptor for each point, we re-find the correspondences between the two point sets, and compare the result with the ground truth. Outliers are added to both point sets by randomly selecting pixels with their SIFT descriptors computed. We gradually increase the ratio of outliers from 0 to 90%. On the other

hand, we regard deformation as the Gaussian White noise added to the position for each point. Specifically, we add each point a perturbation generated by a Gaussian  $\mathcal{N}(0, \varepsilon^2)$  with zero mean and  $\varepsilon$  standard. We also gradually increase  $\varepsilon$  so that the original point sets are more deformed. Since the selection of outliers and the generation of Gaussian White noise are stochastic, we repeat 50 trials for both outlier and deformation tests and then plot the mean and variance. For both outlier and deformation test, we compare our method with some state-of-the-art methods such as RPM [6], CPD [14], PGM [5] and PW [17]. Another two methods SLH [16] and SIFT [18] are not compared in the deformation test because only the position of the points are changed but their feature descriptors are constant. From Fig. 3(b) we can see that with the ratio of outlier increases, the performance of our method drops a little, but is still the best when compared with others. In Fig. 3(c) our method can always find all the correct matches. This shows that our method has strong ability to recover correct matches with high outlier ratio and large deformation.

Then we adopt the NewYork sequence [1] to test the robustness to rotation. This sequence contains 35 images, with rotation angle increases from 0 to  $360^\circ$ . We match the first image to all the other 34 images and evaluate the results using the ground truth provided. From Fig. 4 we can see that our method can find the most correct matches for all the image pairs. The PW [17] algorithm is slightly worse than ours around  $180^\circ$ . The SLH [16] algorithm can always find a certain number of matches but is quite unstable. Large rotation angle leads to collapse for RPM [6], CPD [14] and PGM [5].

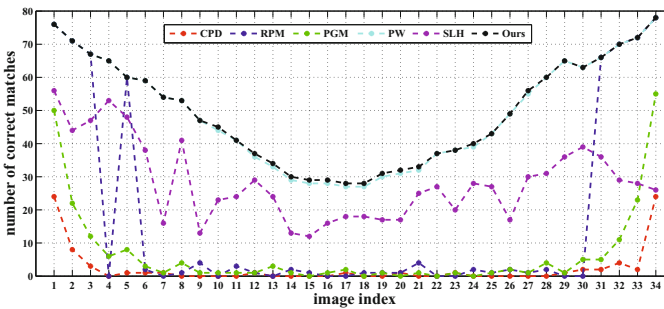
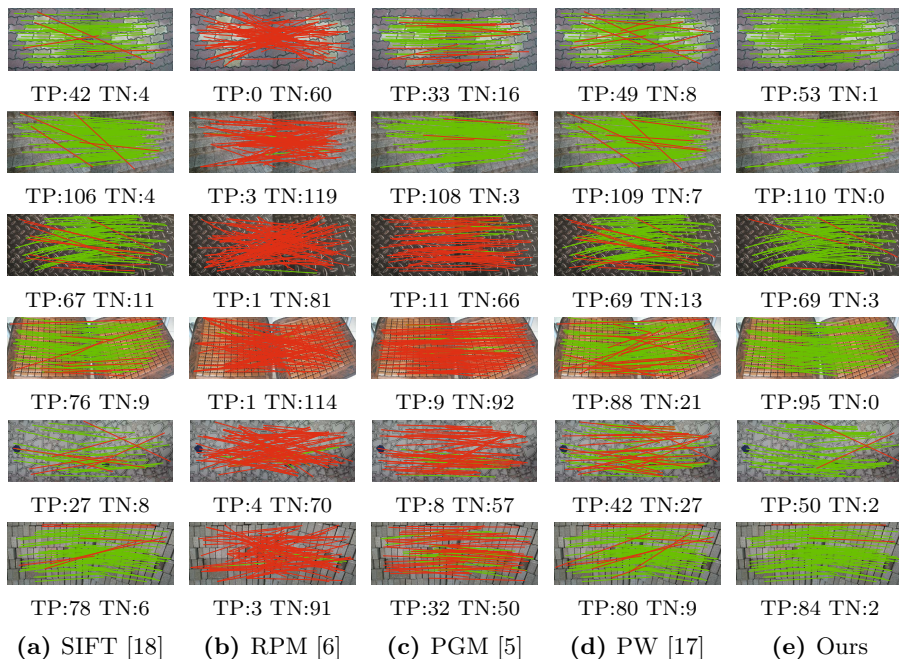


Fig. 4. Robustness to rotation tested on the NewYork sequence [1]

## 5.2 Results on Real Images

More examples are displayed to visually show the advantage of our method in Fig. 5. The images were taken by a digital camera. Each of them contain several repeated patterns so that feature descriptor based methods may produce many false matches. On the other hand, there exists apparent geometry differences as well as outliers between a pair of images which may also defect the spatial



**Fig. 5.** The matching results of 6 image pairs with several repeated patterns as well as apparent geometry differences. True Positives (TP) are in green while True Negatives (TN) are in red. Our method performs the best when considering both the number of correct matches and the precision.

arrangement based methods. True positives and true negatives are manually labeled. From the results we can see that most mismatches of SIFT [18] relate two parts that have similar local appearance but differ from each other globally. RPM [6] also collapses due to outliers and geometry differences. PGM [5] and PW [17] are two other state-of-the-art methods which considers both spatial and feature constraints, but their results are still not as good as ours. Our method can find the most correct matches and at the same time the least mismatches are included. We want to use this experiment to show that our idea of dual constraints can enhance the result when using only feature descriptor or spatial arrangement is insufficient.

## 6 Conclusion

In this paper we propose a new method for image matching. Our main contribution is to use feature information to guide the spatial movement. Specifically, we assign each GMM component a different weight according to feature matching score between the data point and each of the model point. This is achieved by

simply decompose a distance matrix in the feature space. By doing so both the feature and spatial information are considered to enhance the result. Comparison results with state-of-the-art methods on real data show that our method can find the most correct matches while preserving high precision.

**Acknowledgement.** We would like to thank the editors and reviewers for their valuable comments and time spending. This work is supported by National Natural Science Foundation of China(Grants 61371140, 61273279 and 61305044).

## References

1. <http://lear.inrialpes.fr/people/Mikolajczyk/Database/rotation.html>
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Computer Vision and Image Understanding* 110(3), 346 (2008)
3. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part IV*. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
4. Cech, J., Matas, J., Perdoch, M.: Efficient sequential correspondence selection by cosegmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9), 1568–1581 (2010)
5. Cho, M., Lee, K.M.: Progressive graph matching: Making a move of graphs via probabilistic voting. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 398–405 (2012)
6. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* 89(2-3), 114–141 (2003)
7. Hamid, R., Decoste, D., Lin, C.J.: Dense non-rigid point-matching using random projections. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2914–2921 (2013)
8. Hauage, D., Snaveley, N.: Image matching using local symmetry features. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–213 (2012)
9. Chui, H., Rangarajan, A.: A feature registration framework using mixture models. In: *Proc. IEEE Workshop on Math. Methods in Biomedical Image Analysis*, pp. 190–197.
10. Jian, B.C.B., Vemuri: Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8), 1633–1645 (2011)
11. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: *IEEE International Conference on Computer Vision*, vol. 2, pp. 1482–1489 (2005)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
13. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(10), 1615–1630 (2005)
14. A., Myronenko, X.S.: Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12), 2262–2275 (2010)
15. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: *IEEE International Conference on Computer Vision*, pp. 2564–2571 (November 2011)

16. Scott, G.L., Longuet-Higgins, H.C.: An algorithm for associating the features of two images. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 244(1309), 21–26 (1991)
17. Torki, M., Elgammal, A.: One-shot multi-set non-rigid feature-spatial matching. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3058–3065 (2010)
18. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), <http://www.vlfeat.org/>

# Multi-utility Learning: Structured-Output Learning with Multiple Annotation-Specific Loss Functions

Roman Shapovalov<sup>1</sup>, Dmitry Vetrov<sup>1</sup>, Anton Osokin<sup>1,2</sup>, and Pushmeet Kohli<sup>3</sup>

<sup>1</sup> Lomonosov Moscow State University

<sup>2</sup> INRIA — SIERRA Project Team, Paris

<sup>3</sup> Microsoft Research Cambridge

**Abstract.** Structured-output learning is a challenging problem; particularly so because of the difficulty in obtaining large datasets of fully labelled instances for training. In this paper we try to overcome this difficulty by presenting a multi-utility learning framework for structured prediction that can learn from training instances with different forms of supervision. We propose a unified technique for inferring the loss functions most suitable for quantifying the consistency of solutions with the given weak annotation. We demonstrate the effectiveness of our framework on the challenging semantic image segmentation problem for which a wide variety of annotations can be used. For instance, the popular training datasets for semantic segmentation are composed of images with hard-to-generate full pixel labellings, as well as images with easy-to-obtain weak annotations, such as bounding boxes around objects, or image-level labels that specify which object categories are present in an image. Experimental evaluation shows that the use of annotation-specific loss functions dramatically improves segmentation accuracy compared to the baseline system where only one type of weak annotation is used.

**Keywords:** semantic image segmentation, structured-output learning, weakly-supervised learning, loss functions.

## 1 Introduction

Training structured-output classifiers is a challenging problem; not only because of the associated computational burden, but also due to difficulties in obtaining the ground-truth labelling for training data: in problems like semantic image segmentation the structured label may comprise thousands of scalars, so annotation of large datasets requires a lot of human effort. In contrast, it is much easier to obtain a weak annotation of an image, i.e. some statistic of the image labelling. This may take various forms: an image-level label that indicates presence or counts the number of pixels of a particular object category like ‘sky’ or ‘water’, a set of objects’ bounding boxes—rectangles that tightly bound object instances’ segmentations, or a set of seeds—the pixels that have to take the specified labels (Fig. 1). More broadly, weakly-supervised learning may be useful in many training problems where the input is obtained by crowdsourcing.

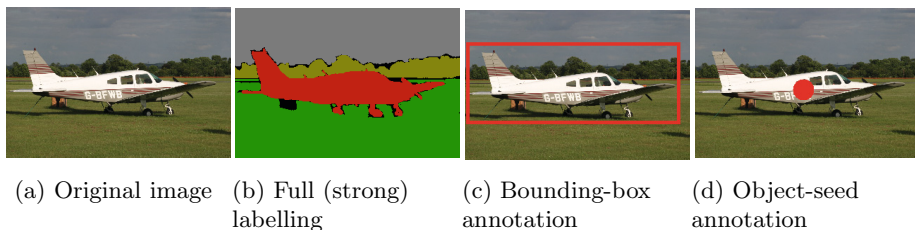


Usage of different annotation types help not only to overcome logistic difficulties, but also to characterize certain categories better. For example, many object categories (i.e. ‘things’ in terms of Heitz and Koller [3]) are better described by bounding-box annotations, while the background categories (i.e. ‘stuff’ [3])—which tend to fill significant parts of an image—by image-level labels.

A number of researchers have recognized the importance of weak annotations for learning semantic segmentation, but most of the methods only use image-level labels. For example, Vezhnevets et al. [20, 21] use a multi-image probabilistic graphical model to propagate image-level annotations across different training images. In this paper, we present a framework for learning structured classification from the mixture of fully and weakly annotated instances. Our framework can employ different types of weak annotations, even for a single instance.

Some papers approach weakly-supervised structural learning by introducing latent variables. Training is performed by latent-variable structural support vector machine (LV-SSVM) [23, 5], hidden conditional random field (HCRF) [10], or their amalgamation [11]. Our work introduces to LV-SSVM *annotation-specific* loss functions, which measure the inconsistency of some labelling predicted by the algorithm with the ground-truth weak annotation. We define those loss functions by describing parametric families and then setting parameters such that their expected value would equal Hamming loss. Due to this definition, the loss functions specific to different annotation types have the same scale. Our framework thus requires only one coefficient, which balances the relative impact of the loss functions for fully labelled and weakly annotated data, since the latter are typically less informative. We empirically show that balancing between these two kinds of loss functions can improve labelling performance.

A number of key technical challenges arise while learning an LV-SSVM model with multiple annotation-specific loss functions. These include solution of the *loss-augmented* and *annotation-consistent* inference problems. The former involves finding the labelling that satisfies the current model and deviates from the annotation the most, while the latter involves finding the best labelling that is consistent with the weak annotation. We show how to solve these optimization problems for various loss functions using efficient optimization algorithms.



**Fig. 1.** Types of annotation for an image from the MSRC dataset [13]

**Relation to Previous Work.** Our work is most closely related to the work of Kumar et al. [5], who use a sequential method to learn semantic segmentation from different types of annotations. Their method starts by training LV-SSVM with a loss function defined on partial labellings; it performs loss-augmented inference using carefully initialized iterated conditional modes (ICM). Once this model is trained, they infer the partial labellings for weakly-annotated images that are consistent with their bounding-box or image-level annotations. The model is then re-trained by considering those solutions as the true partial labellings for the training instances. Unlike Kumar et al. [5], at the training stage we minimize our annotation-specific loss functions simultaneously. In this regard, our framework does not require neither fully nor partially labelled images, which are essential for the first stage of their algorithm. Furthermore, our loss functions allow us to use powerful graph cut based algorithms for solving the loss-augmented and annotation-consistent inference problems, instead of using an ICM-based inference. Finally, we use different types of weak annotations.

For some of the loss functions we use, the loss-augmented inference problems cannot be decomposed to the individual variables. This relates us to the recent work on supervised learning with non-decomposable loss functions [9, 14]. Pletscher and Kohli [9] use a higher-order loss function that penalizes the difference in the area of the target category between binary segmentations. They show how to use graph cuts for efficient exact loss-augmented inference. Tarlow and Zemel [14] use message-passing inference in SSVM training with three different higher-order loss functions: PASCAL VOC loss, bounding box fullness loss, and local border convexity loss.

The line of our work that employs bounding-box weak annotations is related to papers that leverage object detection to perform segmentation, which is extremely helpful for recognizing categories underrepresented in a training set. Ladický et al. [6] describe a CRF model that employs object detection hypotheses and allows for efficient graph-cut inference. Yao et al. [22] use a multi-layer graphical model including the indicators for bounding boxes and image-level categories to perform joint inference. Tighe and Lazebnik [17] perform segmentation and object detection independently, then transfer training set segmentations for the detected objects and combine the resulting segmentation maps on the late stage. In contrast to those methods, we employ the bounding-box annotations within a single structured learning framework using specific loss function.

## Our Contributions

- we propose an LV-SSVM based multi-utility learning framework, which simultaneously minimizes different annotation-specific loss functions, and a unified technique for establishing loss functions for weak annotation of different types;
- we apply our framework to define the loss functions for training semantic segmentation that are specific to the following weak annotation types and their combinations: image-level labels, bounding boxes, and objects’ seeds;
- we propose efficient inference algorithms required for LV-SSVM training with these loss functions.

## 2 Latent-Variable SSVM

### 2.1 Structured-Output Learning

Structured-output learning attempts to learn a mapping  $H$  from the space of features  $\mathcal{X}$  to the space of all possible labellings  $\mathcal{Y}$ . In what follows, we consider only the mappings that can be expressed as maximization of a discriminant function  $F$  that depends linearly on its parameters  $\mathbf{w}$ :

$$H(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}), \quad (1)$$

where vector function  $\Psi(\mathbf{x}, \mathbf{y})$  denotes so-called generalized features of instance  $\mathbf{x} \in \mathcal{X}$  and labelling  $\mathbf{y}$ .  $\Psi(\mathbf{x}, \mathbf{y})$  is defined in a problem-specific way, while the weights  $\mathbf{w}$  are learned from the training data. We address a wide class of so-called *labelling problems*, where the structured label is a vector of discrete variables:  $\mathcal{Y} = \mathcal{K}^V$ , where  $\mathcal{K} = \{1, \dots, K\}$ . Its length  $V$  may vary for individual instances.

The goal of supervised structured-output learning is to obtain the most appropriate weights  $\mathbf{w}$  given the set of features and ground-truth labels of training instances:  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ ,  $\mathbf{y}_n \in \mathcal{Y}_n$ . Here  $\mathcal{Y}_n$  is a set of possible labellings compatible with the  $n$ -th instance. In this paper we follow the max-margin formulation of structured-output learning (also called structural support vector machine, SSVM) [15, 19, 4]:

$$\min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \frac{C}{N} \sum_{n=1}^N \xi_n, \quad (2)$$

$$\text{s.t. } F(\mathbf{x}_n, \mathbf{y}_n; \mathbf{w}) \geq \max_{\bar{\mathbf{y}} \in \mathcal{Y}_n} (F(\mathbf{x}_n, \bar{\mathbf{y}}; \mathbf{w}) + \Delta(\bar{\mathbf{y}}, \mathbf{y}_n)) - \xi_n, \quad \forall n, \quad (3)$$

where  $\Delta(\bar{\mathbf{y}}, \mathbf{y}_n)$  is the loss of some labelling  $\bar{\mathbf{y}} = \{\bar{y}_i\}_{i=1}^V$  with respect to the ground truth labelling  $\mathbf{y}_n = \{y_i^n\}_{i=1}^V$ . Let  $c_i^n$  be some cost associated with the  $i$ -th variable in the labelling of the  $n$ -th instance. The commonly used loss function is the weighted Hamming distance:

$$\Delta(\bar{\mathbf{y}}, \mathbf{y}_n) = \sum_{i \in \mathcal{V}_n} c_i^n [\bar{y}_i \neq y_i^n],^1 \quad (4)$$

This loss function is decomposable w.r.t. the individual variables. It often implies that loss-augmented inference, i.e. maximization in (3), is no more difficult than the maximization of discriminant function  $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ . In some cases it is possible to use higher-order loss functions that cannot be decomposed w.r.t. the individual variables [9, 14, 2].

Problem (2)–(3) is convex and can be solved by the cutting-plane method [19, 4]. This method replaces the constraint (3) with a bunch of linear constraints and then iteratively approximates the feasible polytope by adding the most violated constraint. Such constraint is determined in each iteration by running the loss-augmented inference in (3).

<sup>1</sup> We use the Iverson bracket notation:  $[e] = 1$  if the logical expression  $e$  is true, and  $[e] = 0$  otherwise.

## 2.2 Learning with Weak Annotations

Consider the case when in addition to  $N$  fully-labelled objects, train set contains  $M$  weakly-annotated ones:  $\{(\mathbf{x}_m, \mathbf{z}_m)\}_{m=N+1}^{N+M}$ . From now on, we assume that the weak annotation  $\mathbf{z}_m$  defines a subset of full labellings  $\mathcal{L}(\mathbf{z}_m) \subset \mathcal{Y}$  that are consistent with it, and thus  $\mathbf{z}_m$  is less informative than an individual full labelling  $\mathbf{y}_m$ . Examples of such weak annotations for the image segmentation problem are (1) bounding boxes of the segments of a given label; (2) a value of some global statistic (area, average intensity, number of connected components etc.) for the segments of a given label; (3) subsets of superpixels that belong to a given label (seeds).

We now generalize the standard SSVM formulation to make it handle both fully and weakly annotated data simultaneously. Our multi-utility SSVM is formally defined as follows:

$$\min_{\mathbf{w}, \xi \geq 0, \eta \geq 0} \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \frac{C}{N+M} \left( \sum_{n=1}^N \xi_n + \alpha \sum_{m=1}^M \eta_m \right), \quad (5)$$

$$\text{s.t.} \quad F(\mathbf{x}_n, \mathbf{y}_n; \mathbf{w}) \geq \max_{\bar{\mathbf{y}} \in \mathcal{Y}_n} (F(\mathbf{x}_n, \bar{\mathbf{y}}; \mathbf{w}) + \Delta(\bar{\mathbf{y}}, \mathbf{y}_n)) - \xi_n, \quad \forall n, \quad (6)$$

$$\max_{\mathbf{y} \in \mathcal{L}(\mathbf{z}_m)} F(\mathbf{x}_m, \mathbf{y}; \mathbf{w}) \geq \max_{\bar{\mathbf{y}} \in \mathcal{Y}_m} (F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + K(\bar{\mathbf{y}}, \mathbf{z}_m)) - \eta_m, \quad \forall m. \quad (7)$$

Note that for  $M = 0$  the above formulation degenerates to the standard SSVM formulation, while for  $N = 0$  it reduces to the latent-variable SSVM [23]. Note also that the full labelling  $\mathbf{y}_n$  can be seen as a degenerate weak annotation, where  $\mathcal{L}(\mathbf{z}_m) = \{\mathbf{y}_n\}$ . Therefore, Problem (5)–(7) is almost equivalent to LV-SSVM, but it contains the slack balancing coefficient  $\alpha$ . Ignoring this coefficient may hurt the performance of multi-utility learning, as we show in Section 4.2. In order to perform the optimization, in addition to the loss-augmented inference in (6), we should also be able to perform the weak-loss augmented inference in (7), as well as the *annotation-consistent inference* in the left-hand side of (7).

Optimization problem (5)–(7) is not convex and thus hard. We follow Yu and Joachims [23] and use the concave-convex procedure (CCCP) [24] to solve it approximately.

## 3 Weak Annotation for Semantic Image Segmentation

Semantic image segmentation aims to assign category labels to image pixels. We assume that an image is represented as a set of *superpixels*, i.e. groups of co-located pixels similar by appearance. Consider a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Its nodes  $\mathcal{V}$  correspond to superpixels of the image. The set of edges  $\mathcal{E}$  represents a neighborhood system on  $\mathcal{V}$  that includes the pairs of nodes that correspond to all adjacent superpixels. Let  $\mathbf{x}_i \in \mathbb{R}^d$  be a vector of superpixel features associated with some node  $i \in \mathcal{V}$ ,  $\mathbf{x}_{ij} \in \mathbb{R}^e$  be a vector of superpixel interaction features for the edge connecting nodes  $i$  and  $j$ , and  $\mathbf{x} = \bigoplus_{i \in \mathcal{V}} \mathbf{x}_i \oplus \bigoplus_{(i,j) \in \mathcal{E}} \mathbf{x}_{ij}$  be their

concatenation. The value of each variable  $y_i$  corresponds to the label of the  $i$ -th superpixel. We use the following discriminant function  $F$ :

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) = \sum_{i \in \mathcal{V}} \sum_{k=1}^K [y_i = k] (\mathbf{x}_i^\top \mathbf{w}_k^u) + \sum_{(i,j) \in \mathcal{E}} [y_i = y_j] (\mathbf{x}_{ij}^\top \mathbf{w}^p), \quad (8)$$

where  $\mathbf{w} = \bigoplus_{k=1}^K \mathbf{w}_k^u \oplus \mathbf{w}^p$  is a vector of the model parameters, and  $\mathbf{w}_k^u \in \mathbb{R}^d$ ,  $\mathbf{w}^p \in \mathbb{R}^e$ . The summands in the first and the second terms are called unary and pairwise potentials, respectively. We restrict pairwise weights  $\mathbf{w}^p$  and pairwise features  $\mathbf{x}_{ij}$  to be nonnegative and thus obtain an associative discriminative function (with only attractive pairwise potentials) [15]. Maximizing  $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$  w.r.t.  $\mathbf{y}$  is known to be NP-hard, but efficient approximate algorithms exist, e.g.  $\alpha$ -expansion [1].

We use the weighted Hamming loss (4) for fully-labelled images, where  $c_i$  is the number of pixels in the corresponding superpixel, so the loss function estimates the number of mislabelled image pixels.<sup>2</sup> To use some type of weak annotations for training, we need to define the annotation-specific loss function that allows loss-augmented inference and annotation-consistent inference. The former should be efficient, since it is performed in the inner loop of training and thus is typically a bottleneck. We show how to define and combine them for the annotations of the following types: image-level labels, bounding boxes around objects, and objects' seeds.

### 3.1 Image-Level Labels

We start by defining loss functions  $K(\mathbf{y}, \mathbf{z})$  for some arbitrary labelling  $\mathbf{y}$  and ground-truth weak annotation  $\mathbf{z}$ . In this subsection we assume that  $\mathbf{z}$  is a set of labels used in the ground-truth image labelling (for the image in Fig. 1,  $\mathbf{z} = \{\text{'sky'}, \text{'tree'}, \text{'plain'}, \text{'grass'}\}$ ). We cannot compute the Hamming loss (4) if the full labelling is unknown for one of its arguments. Let's instead define a proxy loss function, that is symmetric and does not compare labels of any superpixels directly:

$$\Delta_{\text{il}}(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i \in \mathcal{V}} c_i [\nexists j \in \mathcal{V} : y_j = \bar{y}_i \vee \nexists j \in \mathcal{V} : \bar{y}_j = y_i]. \quad (9)$$

It penalizes all the superpixels that have been given any label that is absent in the annotation  $\bar{\mathbf{y}}$ , as well as superpixels which have ground truth labels that is absent in  $\mathbf{y}$ . Unfortunately, the ground-truth labelling  $\bar{\mathbf{y}}$  is unknown. If we knew the areas  $S_k$  of each label  $k \in \mathbf{z}$ , we could derive the following upper bound on (9):

$$K_{\text{il}}(\mathbf{y}, \mathbf{z}; \{S_k\}_{k \in \mathbf{z}}) = \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathbf{z}} S_k \prod_{i \in \mathcal{V}} [y_i \neq k]. \quad (10)$$

<sup>2</sup> In practice, ground-truth labelling of a superpixel may contain several labels; in this case the number of incorrectly inferred pixels is added to the loss. We ignore this case to ease the notation, but all the algorithms still work in that case.

This upper bound is tight up to a factor of 2. The first term penalizes the pixels labelled with wrong labels, while the second term penalizes ignoring the labels from  $\mathbf{z}$ .

Since we do not know the areas  $S_k$ , the best we can do is to assume  $K(\mathbf{y}, \mathbf{z})$  to be the expectation of (10) taken over all full labellings consistent with  $\mathbf{z}$ . If there are enough fully-labelled images, the areas  $S_k$  can be estimated. Otherwise we assume the uniform distribution over the feasible full labellings  $\mathbf{y} \in \mathbf{z}$  and get

$$K_{\text{il}}(\mathbf{y}, \mathbf{z}) = \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathbf{z}} \frac{\sum_{i \in \mathcal{V}} c_i}{|\mathbf{z}|} \prod_{i \in \mathcal{V}} [y_i \neq k]. \quad (11)$$

Having defined the loss function  $K_{\text{il}}$ , we need to provide algorithms for inference problems in (7). For annotation-consistent inference  $\max_{\mathbf{y} \in \mathbf{z}_m} F(\mathbf{x}_m, \mathbf{y}; \mathbf{w})$  we use  $\alpha$ -expansion over the labels from  $\mathbf{z}_m$  only. Note that this may result in an inconsistent labelling: some labels from  $\mathbf{z}_m$  may miss in  $\mathbf{y}$ . We have tried an heuristic algorithm for making it strictly consistent with  $\mathbf{z}$  by changing one node per missing label, but observed no significant difference in practice.

The loss-augmented inference is now not decomposable to unary and pairwise factors. To work this around, we derive:

$$\begin{aligned} & \max_{\bar{\mathbf{y}} \in \mathcal{Y}_m} (F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + K_{\text{il}}(\bar{\mathbf{y}}, \mathbf{z}_m)) = \\ & \max_{\bar{\mathbf{y}} \in \mathcal{Y}_m} \left( F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [\bar{y}_i = k] - \sum_{k \in \mathbf{z}} \frac{\sum_{i \in \mathcal{V}} c_i}{|\mathbf{z}|} [\exists i : \bar{y}_i = k] \right) + \text{const.} \end{aligned} \quad (12)$$

The last maximization is the standard MRF inference problem with label costs. We use the efficient modification of  $\alpha$ -expansion for accounting label costs [2].

### 3.2 Bounding Boxes

It is convenient to annotate instances in an image with tight bounding boxes (Fig. 1c). On the other hand, they do not give much information for background categories. Therefore, we consider the annotation that consists of both bounding boxes and image-level labels. For example, annotation of an image may contain the bounding-box locations of cars and pedestrians, and additionally state that there are buildings, road, and sky in the image. We assume that within a certain image each category can be defined either with image-level labels, or with bounding boxes, though the type of annotation for a category may vary from image to image (see Section 4.3 for an example where it can be useful).

We model weak annotation  $\mathbf{z}$  of an image as a pair  $(\mathbf{z}^{\text{il}}, \mathbf{z}^{\text{bb}})$  of image-level and bounding box annotations. The latter is a set of bounding boxes with associated category labels:  $\mathbf{z}^{\text{bb}} = \{z_i\}$ , with the following functions defined:  $\text{label}(z_i)$ , which defines the associated category label, and  $\text{box}(z_i) = [\text{left}(z_i), \text{right}(z_i)] \times [\text{top}(z_i), \text{bottom}(z_i)]$  that defines the extent of the bounding box. The set of labels  $\mathcal{K}$  is partitioned into three subsets w.r.t. the weak annotation  $\mathbf{z}$ : the labels

that are defined with bounding boxes ( $\mathcal{K}_b = \bigcup_{z \in \mathbf{z}^{\text{bb}}} \text{label}(z)$ ), those that are present somewhere else in the image ( $\mathcal{K}_p = \mathbf{z}^{\text{il}}$ ), and those that are absent ( $\mathcal{K}_a = \mathcal{K} \setminus (\mathcal{K}_b \cup \mathcal{K}_p)$ ). Nodes  $\mathcal{V}$  are also partitioned:  $\mathcal{V}_k = \bigcup_{z \in \mathbf{z}^{\text{bb}}: \text{label}(z)=k} \text{box}(z)$  is the union of pixel indices in the bounding boxes corresponding to the label  $k \in \mathcal{K}_b$ , and  $\mathcal{V}_0 = \mathcal{V} \setminus \bigcup_{k \in \mathcal{K}_b} \mathcal{V}_k$ . We now define the combined loss function as:

$$\begin{aligned}
 K_{\text{il-bb}}(\mathbf{y}, \mathbf{z}) = & \sum_{k \in \mathcal{K}_a} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathcal{K}_p} \sigma_k \prod_{i \in \mathcal{V}} [y_i \neq k] + \\
 & \beta \sum_{z \in \mathbf{z}^{\text{bb}}} \left( \sum_{p=\text{top}(z)}^{\text{bottom}(z)} \nu_p^z \prod_{q=\text{left}(z)}^{\text{right}(z)} V((p, q); \mathbf{y}, \text{label}(z)) + \sum_{q=\text{left}(z)}^{\text{right}(z)} \omega_q^z \prod_{p=\text{top}(z)}^{\text{bottom}(z)} V((p, q); \mathbf{y}, \text{label}(z)) \right) \\
 & + \sum_{k \in \mathcal{K}_b} \sum_{i \in \mathcal{V}_0} c_i [y_i = k]. \quad (13)
 \end{aligned}$$

The first two terms are almost the same as in (11), but the estimate of the category area in the second term does not include the pixels within the bounding boxes:  $\sigma_k = (\sum_{i \in \mathcal{V}_0} c_i) / |\mathbf{z}^{\text{il}}|$ . The third term penalizes ‘empty’ rows and columns in the bounding boxes, i.e. those rows and columns that do not contain pixels of a target category at all. The violation function  $V$  is defined as:

$$V(\mathbf{p}; \mathbf{y}, k) = \begin{cases} 1, & \text{if } \text{map}(\mathbf{y})_{\mathbf{p}} \neq k, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Here  $\text{map}(\mathbf{y})$  is the classification map induced by the superpixel labelling  $\mathbf{y}$ . Coefficients  $\nu_p^z$  and  $\omega_q^z$  allow us to assign the penalty for the corresponding row or column being empty, depending on its position in the bounding box. One can learn the category-specific profiles of  $\nu^z$  and  $\omega^z$  when the full labelling is abundant enough, but we use uniform profiles assuming that half of a bounding box is occupied by the object on average:  $\nu_p^z = (\text{right}(z) - \text{left}(z))/2$ ,  $\omega_q^z = (\text{bottom}(z) - \text{top}(z))/2$ . Note that this makes the loss an estimate on the number of mislabelled pixels (similar to the image-level label loss (11)), so the value coefficient  $\beta = 1$  should work well (we show in Section 4.3 that it really does). We have also tried linearly decreasing loss used by Kumar et al. [5], but it did not affect the performance significantly. The last term penalizes the bounding-box labels outside of bounding boxes.

We have shown in the previous section how to account for the two initial terms in the loss-augmented inference. The last term is decomposable w.r.t. superpixels. The third term is a sum over the higher-order cliques of the following form. For each bounding box  $z$ , each row and each column generates a clique of nodes corresponding to the superpixels that intersect that row/column. We treat them the same way as the image-level loss: we modify  $\alpha$ -expansion with label costs [2] to penalize each clique of superpixels, which contains at least one superpixel labelled with  $\text{label}(z)$ . There is a technical difficulty with the superpixels that cross the bounding box border: it is unclear if their labelling with  $\text{label}(z)$  should be penalized. We adopted the following strategy: shrink the bounding box to allow some margin, and treat all superpixels that intersect the shrunk bounding

box (and only them) as insiders. We set the margin width equal to 6% of the corresponding bounding box dimension.

During the annotation-consistent inference, we need to infer a labelling that has objects only in bounding boxes of the corresponding category labels, and they should fill those bounding boxes tightly, i.e. touch upon all four sides of the bounding box shrunk to allow a 6% margin (Lempitsky et al. [7] showed that this corresponds to the tightness in a typical labelling produced by a human). The first condition is easy to satisfy: we can suppress certain labels outside of bounding boxes by using infinite unary potentials. To provide tightness, we use a variation of the pinpointing algorithm [7], adapted for the multi-class segmentation. First, segmentation is performed without the tightness constraints. Then, until those constraints are satisfied, one of the superpixels changes its unary potential, and expansion move is performed. In our implementation, we select the superpixel with the highest relative potential for  $label(z)$  that has not been assigned this label yet, and assign it the infinite potential for  $label(z)$  to guarantee that it will change its label. This procedure is finite because at each iteration at least one superpixel within  $box(z)$  switches to  $label(z)$ . In contrast to Lempitsky et al. [7], we do not perform further dilation, since it is unclear, which label we should use for expansion move(s); neither of the heuristics we tried improved the result significantly. We also found that initialization of the latent variables in LV-SSVM matters: we obtained the best results when initially all superpixels within  $box(z)$  were initialized with  $label(z)$ . Note that Kumar et al. [5] used a different criterion during the annotation-consistent inference: they penalize the empty rows and columns within bounding boxes (the opposite to what we do in loss-augmented inference). Note that their heuristic does not guarantee the tightness of the resulting segmentation.

### 3.3 Objects' Seeds

Another form of a weak annotation natural for the object categories is the seed annotation (Fig. 1d). In general, for a segment of some category, a seed is a subset of its pixels. We consider a particular case, where only one pixel, presumably close to the segment center, is labelled. During the annotation-consistent inference, we require the superpixel where this point is located to have the fixed seed label.

We now model the weak annotation  $\mathbf{z}$  as a pair  $(\mathbf{z}^{il}, \mathbf{z}^{os})$ , where  $\mathbf{z}^{os}$  is a set of 2D points with the corresponding labels:  $(\mathbf{p}, k)$ . The seed centrality assumption allows us to set the Gaussian penalty for inferring any non-seed label in the neighbourhood of each seed, which brings us to the following loss function:

$$K_{il-os}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathbf{k}_a} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathbf{k}_p} \sigma_k \prod_{i \in \mathcal{V}} [y_i \neq k] + \beta \sum_{\substack{(\mathbf{p}', k') \\ \in \mathbf{z}^{os}}} \sum_{\mathbf{p} \in I} V(\mathbf{p}; \mathbf{y}, k') \exp\left(-\frac{\pi \|\mathbf{p} - \mathbf{p}'\|^2}{\tau_{k'}}\right). \quad (15)$$



Here the first two terms are the same as in the image-level label loss. The inner sum in the third term is taken over all image pixels  $I$ . The form of the Gaussian is defined in such a way that the penalty for misclassification of the central pixel  $\mathbf{p}'$  is 1, and whenever no superpixels of the label  $k'$  are found, the penalty is equal to the estimated area of the label  $k'$  w.r.t. all labellings consistent with the weak annotation; specifically,

$$\tau_{k'} = \frac{\sum_{i \in \mathcal{V}} c_i}{(|\mathbf{z}^{\text{il}}| + \#\text{Lab}(\mathbf{z}^{\text{os}})) \cdot \#\text{Obj}(\mathbf{z}^{\text{os}}, k')}. \quad (16)$$

Here  $\#\text{Lab}(\mathbf{z}^{\text{os}})$  is the number of different labels in  $\mathbf{z}^{\text{os}}$ , and  $\#\text{Obj}(\mathbf{z}^{\text{os}}, k')$  is the number of seeds of the label  $k'$  in  $\mathbf{z}^{\text{os}}$ . Loss (15) is decomposable to factors, so the loss-augmented inference is trivial.

## 4 Experiments

### 4.1 Datasets and Metrics

We test the proposed framework on two datasets: MSRCv2<sup>3</sup> [13, 20] and SIFT-flow<sup>4</sup> [8, 16, 21]. MSRC contains 276 training and 256 test images that are fully labelled using 23 category labels; significant part of pixels remains unlabelled. SIFT-flow is a more challenging dataset: it is a subset of the LabelMe database [18], which contains 2488 training and 200 test images; they are labelled to 33 categories using crowd-sourcing. See Appendix A for details on the used features.

*Quality Measures.* We use two standard measures of segmentation quality: accuracy and per-class recall. The accuracy is defined as the rate of correctly labelled pixels of the test set. The per-class recall is the number of correctly labelled pixels of each category divided by the true total area of that category, averaged over categories. Following the previous work [20, 12], we exclude the pixels of rare categories (‘horse’ and ‘mountain’) from recall computation for MSRC, but include the ‘other’ label, see Section 4.2. Similarly, we exclude rare categories (‘cow’, ‘desert’, ‘moon’, and ‘sun’) from SIFT-flow recall computation.

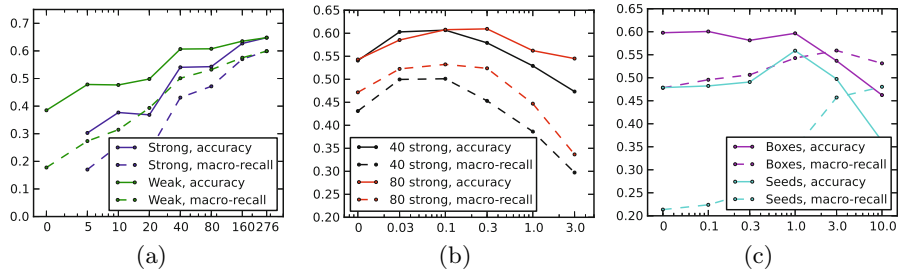
### 4.2 Image-Level Labels

*Varying the Full-Labeling Rate.* In our basic setting we have a (possibly empty) part of the training set fully labelled, while the rest of the images have only image-level labels. We generate those subsets using the Metropolis–Hastings sampling, trying to make the distribution of their label counts approximate that of the whole training set. Fig. 2a shows the accuracy and per-class recall of the test set segmentation for various full labelling rates in comparison to the fully-supervised setting.<sup>5</sup> In the most common scenario—when less than 20% of the training set is fully labelled—the weakly-annotated subset provides a stable 10–15% improvement both in terms of the accuracy and mean per-class recall.

<sup>3</sup> <http://research.microsoft.com/en-us/projects/objectclassrecognition/>

<sup>4</sup> <http://people.csail.mit.edu/ceiliu/LabelTransfer/code.html>

<sup>5</sup> <http://shapovalov.ro/data/MSRC-weak-train-masks.zip>



**Fig. 2.** (a)–(c) Accuracy (solid lines) and per-class recall (dashed lines) subject to different parameters on the MSRC dataset. (a) Varying the number of fully-labelled images. Blue line show test set segmentation quality when only fully-labelled images are available; green line—when the complementary part of the train set has image-level labels. (b) Varying the coefficient of the weak-loss coefficient  $\alpha$ . Black line show test set segmentation quality when 40 images are fully labelled, red line—when 80 images; the complementary part of the train set has image-level labels. (c) Varying the coefficient of the bounding box (magenta line) or object seed (cyan line) loss  $\beta$ . All 276 training images have image-level labels, all objects have tight bounding box or seed annotations, respectively.

*Balancing the Loss Functions.* When the training set consists of both weak annotations and full labellings, the coefficient  $\alpha$  from (5) needs to be set. We discovered that its optimal value was lower than 1 (Fig. 2b shows the dependency of performance on  $\alpha$ ). We speculate that this is because we are more certain about the strong loss, so it should contribute to the objective more. Thus, for all the other experiments we set  $\alpha = 0.1$ .

*SIFT-Flow Results.* On the SIFT-flow dataset, we compare fully-supervised learning with weakly-supervised at one point, i.e. when only 256 training images are fully labelled, and the rest 2232 images have only image-level labels (Table 1). This weakly-learned model loses to the fully-supervised one only 2% in the accuracy and 4% in the per-class recall. Note that our model is *on par* with Vezhnevets et al. [21], who also reached 21% on that dataset with the same superpixels and features. The difference is they used only image-level annotation, while we used about 10% fully labelled images. However, their model is substantially more complicated: they use extremely-randomized hashing forest for non-linear feature transform, learn objectness and image-level priors, and connect superpixels of different images within the multi-image model. Since the LV-SSVM optimization problem is not convex, the algorithm may get stuck at local minima. We initialize the parameters of LV-SSVM by the parameters of the SSVM trained on the fully-labelled part of the dataset, if there is one.

### 4.3 Adding Bounding Boxes and Seeds

*Generating Weak Annotation.* We generate two more annotations for the MSRC training data to test additional annotation-specific loss functions. Similar to image-level labels, we generate them from the full labelling. Tight bounding

**Table 1.** Accuracy and average per-class recall on the SIFT-flow dataset. The first two lines describe training on the subset of 256 fully labelled images of the models with and without pairwise potentials, respectively. The third line experiment used the whole dataset with image-level labels, but for only 256 of them full labelling is known. The bottom line shows the result when the whole dataset is fully labelled.

experiment	acc	rec
256/256 strong, local	0.574	0.167
256/256 strong, init loc.	0.620	0.176
256/2488 strong, init $\uparrow$	0.674	0.208
2488/2488 strong	0.696	0.246

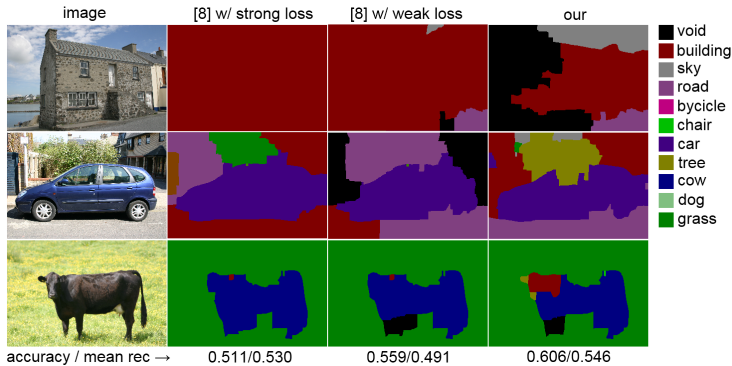
**Table 2.** Accuracy (first number in each cell) and average per-class recall (second number) on the MSRC dataset when during training i) only full labelling is available, ii) image-level (il) labels are also available for the rest of the data set, iii) object seeds (os) are additionally available, iv) bounding boxes (bb) for objects are available, v) both seeds and bounding boxes are available. Note that the numbers in the last column are all equal since the weak annotation does not add any information when all training set is fully labelled

il	bb	os	0/276 strong	5/276 strong	276 strong
-	-	-	n/a	0.300/0.170	0.648/0.599
+	-	-	0.385/0.178	0.478/0.273	0.648/0.599
+	-	+	0.559/0.346	0.574/0.370	0.648/0.599
+	+	-	0.597/0.543	0.606/0.546	0.648/0.599
+	+	+	0.531/0.567	0.542/0.564	0.648/0.599

boxes and object seeds are good for description of the object (‘thing’) categories, while do not add much information beyond image-level labels for the background (‘stuff’) categories. We divide the list of categories into two parts: background, which includes ‘grass’, ‘sky’, ‘mountain’, ‘water’, ‘road’, and ‘other’; and objects, which includes all other categories. There are two ambivalent categories—‘building’ and ‘tree’—which can instantiate either a target object of a photograph, or background. We used the following heuristic for each image: consider tree and building as background iff there are other objects in the image. We enhanced the image-level labelling with either tight bounding boxes or object seeds for segments of object categories only. For the other categories, only image-level labels were available. To generate seeds, for each segment we took its pole of inaccessibility—the point that maximizes its distance transform map.

*Results.* Table 2 summarizes the results. When the full labelling is unavailable, both object seed and bounding box annotations give significant improvement over just image-level labels. Bounding boxes notably increase per-class recall: they help to better learn ‘thing’ categories, which are numerous and typically have smaller area. Overall, learning with bounding boxes only 5% inferior to learning on fully labelled data both in terms of the accuracy and per-class recall. Object seed annotation gave more modest increase in performance, though is easier to obtain. We used the value  $\beta = 1$  to balance the impact of image-level vs. bounding box (or seed) loss functions: they seem to provide equal contribution to the objective function; Fig. 2c supports that hypothesis.

*Comparison to Kumar et al. [5].* Unfortunately, we cannot directly compare to Kumar et al. [5] since the type of input data for their framework is unorthodox. They use two different datasets to obtain segmentation maps (partial labellings)



**Fig. 3.** Qualitative results of the proposed algorithm and two variations of the algorithm by Kumar et al. [5] applied to three images from the MSRC test set

for the foreground and background categories, respectively. Our framework does not support this kind of annotation: we believe that it is easier to obtain segmentation for background and foreground categories using the same set of images. This eliminates the need to use the latent-variable SSVM for training the basic model; instead the global minimum of SSVM objective can be found efficiently. Also, when both image-level labels and bounding boxes (or seeds) are known for each weakly-annotated image, both background and foreground partial labellings can be inferred, and using latent-variable SSVM after adding weakly-annotated data is not necessary again. Thus, when given the data we use, the method of Kumar et al. [5] could look like this:

- train SSVM using the fully-labelled part of the training set,
- use the trained model to infer labelling consistent with the weak annotation,
- train SSVM using the hallucinated labelling obtained in the previous step.

This method is similar to running one outer iteration of our training algorithm, but it has one important difference: the loss function in the second SSVM. While our method uses the weak loss function, the modified method of Kumar et al. [5] uses the strong loss function w.r.t. the hallucinated labelling. To compare the methods, we use the MSRC training set with 5 fully-labelled images and the rest annotated with bounding boxes and image-level labels (row 4, column 2 in Table 2, excluding headers) to train both described modifications: with the weak bounding-box loss function (13), and with the strong loss function (4) (still different from the loss function of Kumar et al. [5]). The segmentation maps and numerical results in Fig. 3 show that the proposed simultaneous minimization of loss functions is superior both in terms of accuracy and per-class recall.

## 5 Conclusion

We presented the framework for learning structural classification from different types of annotations by minimizing annotation-specific loss functions. We applied

it to semantic image segmentation by introducing weak loss functions for for image-level, bounding box, and object seed annotations. Usage of weakly-annotated training data consistently improves the labelling. The results on the semantic segmentation datasets show that the joint annotation where background is given by image-level labels, and objects are given by bounding boxes, is the best trade-off between segmentation quality and annotation effort.

## References

- [1] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *PAMI* 23(11), 1222–1239 (2001)
- [2] Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast Approximate Energy Minimization with Label Costs. *IJCV* 96(1), 1–27 (2012)
- [3] Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
- [4] Joachims, T., Finley, T., Yu, C.: Cutting-plane training of structural SVMs. *Machine Learning* 77(1), 27–59 (2009)
- [5] Kumar, M.P., Turki, H., Preston, D., Koller, D.: Learning specific-class segmentation from diverse data. In: *ICCV*, pp. 1800–1807 (November 2011)
- [6] Ladický, Ľ., Sturges, P., Alahari, K., Russell, C., Torr, P.H.S.: What, Where and How Many? Combining Object Detectors and CRFs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part IV*. LNCS, vol. 6314, pp. 424–437. Springer, Heidelberg (2010)
- [7] Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: *ICCV*, pp. 277–284 (September 2009)
- [8] Liu, K., Raghavan, S., Nelesen, S., Linder, C.R., Warnow, T.: Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science* (New York, N.Y.) 324(5934), 1561–1564 (2009)
- [9] Pletscher, P., Kohli, P.: Learning low-order models for enforcing high-order statistics. In: *AISTATS* (2012)
- [10] Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T.: Hidden conditional random fields. *PAMI* 29(10), 1848–1853 (2007)
- [11] Schwing, A.G., Hazan, T., Pollefeys, M., Urtasun, R.: Efficient Structured Prediction with Latent Variables for General Graphical Models. In: *ICML* (2012)
- [12] Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: *CVPR* (June 2008)
- [13] Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: *textonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006, Part I*. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
- [14] Tarlow, D., Zemel, R.S.: Structured Output Learning with High Order Loss Functions. In: *AISTATS* (2012)
- [15] Taskar, B., Chatalbashev, V., Koller, D.: Learning associative Markov networks. In: *ICML*. pp. 102–109, Banff, Alberta, Canada (2004)
- [16] Tighe, J., Lazebnik, S.: SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part V*. LNCS, vol. 6315, pp. 352–365. Springer, Heidelberg (2010)

- [17] Tighe, J., Lazebnik, S.: Finding Things: Image Parsing with Regions and Per-Exemplar Detectors. In: CVPR, pp. 3001–3008 (June 2013)
- [18] Torralba, A., Russel, B.C., Yuen, J.: LabelMe: Online Image Annotation and Applications. *Proceedings of the IEEE* 98(8), 1467–1484 (2010)
- [19] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. *JMLR* 6, 1453–1484 (2006)
- [20] Vezhnevets, A., Ferrari, V., Buhmann, J.M.: Weakly Supervised Semantic Segmentation with a Multi-Image Model. In: ICCV, Barcelona, ES (2011)
- [21] Vezhnevets, A., Ferrari, V., Buhmann, J.M.: Weakly Supervised Structured Output Learning for Semantic Segmentation. In: CVPR, Providence, RI (2012)
- [22] Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: CVPR (June 2012)
- [23] Yu, C.N.J., Joachims, T.: Learning structural SVMs with latent variables. In: ICML, Montreal, Canada (2009)
- [24] Yuille, A., Rangarajan, A.: The concave-convex procedure (CCCP). In: NIPS (2002)

# Mapping the Energy Landscape of Non-convex Optimization Problems

Maira Pavlovskaja<sup>1</sup>, Kewei Tu<sup>2</sup>, and Song-Chun Zhu<sup>1</sup>

<sup>1</sup> Department of Statistics, University of California, Los Angeles,  
8125 Math Science Bldg, Los Angeles, CA 90095, USA  
{mariapavl, sczhu}@ucla.edu

<sup>2</sup> School of Information Science and Technology, Shanghai Tech University,  
No. 8 Building, 319 Yueyang Road, Shanghai 200031, China  
tukw@shanghaitech.edu.cn

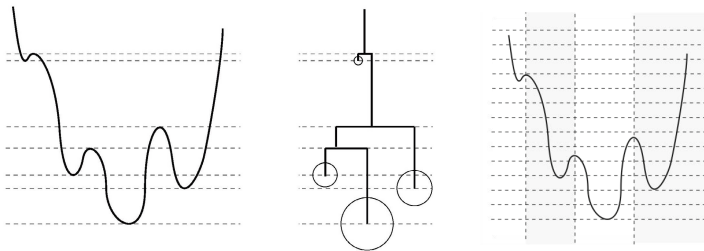
**Abstract.** An *energy landscape map* (ELM) characterizes and visualizes an energy function with a tree structure, in which each leaf node represents a local minimum and each non-leaf node represents the barrier between adjacent energy basins. We demonstrate the utility of ELMs in analyzing non-convex energy minimization problems with two case studies: clustering with Gaussian mixture models and learning mixtures of Bernoulli templates from images. By plotting the ELMs, we are able to visualize the impact of different problem settings on the energy landscape as well as to examine and compare the behaviors of different learning algorithms on the ELMs.

## 1 Introduction

In many computer vision, pattern recognition and learning problems, the energy function to be optimized is highly non-convex. A large body of work has been devoted to designing algorithms that are capable of efficiently finding a good local optimum in the non-convex energy landscape. On the other hand, much less work has been done in analyzing the properties of such non-convex energy landscapes.

In this paper, inspired by the success of visualizing the landscapes of Ising and Spin-glass models by [2] and [14], we compute *Energy Landscape Maps* (ELMs) in the high-dimensional hypothesis spaces for a few model learning problems in computer vision and pattern recognition — learning mixtures of Gaussian and learning mixtures of Bernoulli templates. An ELM is a tree structure in which each leaf node represents a local minimum whose energy determines the y-axis position of the leaf node; each non-leaf node represents the energy barrier between local minima. Figure 1 shows an example energy function and the corresponding ELM. The ELM of an energy landscape reveals important characteristics of the landscape, including

- the number of local minima and their energy levels;
- the energy barriers between adjacent local minima; and



**Fig. 1.** (Left) An energy function. (Middle) Its corresponding ELM. The y-axis of the ELM is the energy level. Each leaf node is a local minimum and the leaf nodes are connected at the energy barrier between their energy basins. The probability mass or volume of an energy basin is indicated by the size of the circle around the leaf node. (Right) Partition of the spaces into bins according to basins and energy levels.

- the probability mass and volume of each local minimum.

Such information can be very useful in analyzing the intrinsic complexity of the optimization problems (for either inference or learning tasks), analyzing the effects of various conditions on the complexity, and visualizing the behavior of different optimization algorithms (i.e. how they move in the landscape).

ELMs can be efficiently constructed by running a MCMC algorithm that features a dynamic reweighting scheme allowing the sampler to cross energy barriers and efficiently traverse the entire space. In the literature, Becker and Karplus [2] presents the first work for visualizing multidimensional energy landscapes for the spin-glass model. Liang [6,7] generalizes the Wang-Landau algorithm [13] for random walks in the state space. Zhou [14] uses the generalized Wang-Landau algorithm to plot the landscape for Ising model with hundreds of local minima and proposes an effective way for estimating the energy barriers. In contrast to the above work that compute the landscapes in “state” spaces for inference problems, our work is focused on the landscapes in “hypothesis” spaces (the sets of all models) for statistical learning problems. We modify the previous MCMC algorithm to handle several new issues that arise in plotting ELMs of continuous hypothesis spaces.

## 2 ELM Construction in Hypothesis Spaces

Let  $\mathcal{H}$  be a hypothesis space for a learning problem and let  $E(x)$  be the energy of a hypothesis  $x \in \mathcal{H}$ . For example, in a  $n$ -component mixture of Gaussian clustering problem, given a training dataset, a posterior probability  $\pi(x)$  is defined and  $x$  includes the model parameters such as the means and variances of the  $n$  unknown Gaussians; the landscape is defined by energy function  $E(x) = -\log \pi(x)$ . For simplicity, we bound  $\mathcal{H}$  by limiting  $x$  to a finite range calculated from the input data points.



As Figure 1 (right) shows, the finite hypothesis space is partitioned into energy basins  $D_i$  and each basin is further partitioned into energy intervals  $[u_{j+1}, u_j]$ . Thus the space  $\mathcal{H}$  is divided into bins  $D_{i,j}$

$$D_{i,j} = \{x : x \in D_i, E(x) \in [u_{j+1}, u_j]\}. \tag{1}$$

Let  $\phi(x)$  be the index mapping  $x$  to the bin index  $(i, j)$ , and  $\beta_{ij} = \pi(D_{ij})$  the probability mass of bin  $D_{i,j}$ . Our goal is to design an MCMC algorithm with equal probability visiting all bins, i.e. its state at time  $t$  follows a new equalized probability,

$$x_t \sim \pi^+(x) \propto \frac{\pi(x)}{\beta_{\phi(x)}}. \tag{2}$$

The generalized Wang-Landau algorithm estimates  $\beta_{ij}$  by  $\gamma_{ij}$  using stochastic gradient. The algorithm goes as follows:

1. Initialize a sample  $x_0 \in \mathcal{H}$  and the bin weights  $\gamma_{ij}^0$  for the bins  $D_{i,j}$ . Repeat step 2-6:
2. At step  $t$ , sample  $y \sim Q(x_t, y)$  from some proposal distribution  $Q$ .
3. Perform steepest descent initialized with  $y$  to find the energy basin that  $y$  belongs to. Let  $\phi(y)$  be the index of the bin containing  $y$ .
4. Accept proposal  $y$  with probability  $\alpha(x_t, y)$ :

$$\alpha(x_t, y) = \min \left( 1, \frac{Q(y, x_t) \pi(y) \gamma_{\phi(x_t)}^t}{Q(x_t, y) \pi(x_t) \gamma_{\phi(y)}^t} \right). \tag{3}$$

5. If the proposal is accepted, increase the weight  $\gamma_{\phi(y)}^{t+1} = \gamma_{\phi(y)}^t * f$  for some constant  $f > 1$ .
6. If  $x_t$  and  $y$  belong to different basins  $D_k$  and  $D_l$ , then perform ridge descent to update the estimated upper-bound of the energy barrier between the two basins. In ridge descent we search for a local minimum along the ridge between the two basins, by starting with  $a_0 = x_t, b_0 = y$  and iterating to find  $(a_t, b_t)$ :

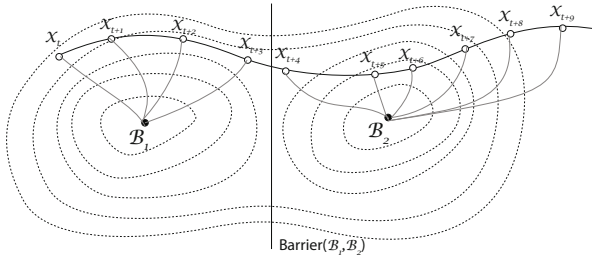
$$a_t = \operatorname{argmin}_a \{E(a) : a \in \text{Neighborhood}(b_{t-1}) \cap D_k\}$$

$$b_t = \operatorname{argmin}_b \{E(b) : b \in \text{Neighborhood}(a_t) \cap D_l\}$$

until  $b_{t-1} = b_t$ . The neighborhood of a sample is defined as the subspace surrounding the sample with its size controlled by an adaptive radius.

7. After the algorithm converges, construct the ELM based on the energy of the basins that have been discovered and the estimated energy barriers between them. We check the convergence of the algorithm using the multivariate extension of the Gelman and Rubin criterion [5].

Figure 2 illustrates the Markov chain produced by the algorithm. Note that the modified acceptance probability in eqn.(3) will reject sample  $y$  if the Markov chain has visited bin  $\phi(y)$  many times, forcing the sampler to move into less explored space.



**Fig. 2.** Sequential MCMC samples  $x_t, x_{t+1}, \dots, x_{t+9}$ . For each sample, we perform gradient descent to determine which energy basin the sample belongs to. If two sequential samples fall into different basins ( $x_{t+3}$  and  $x_{t+4}$  in this example), we estimate or update the upper-bound of the energy barrier between their respective basins ( $B_1$  and  $B_2$  in this example).

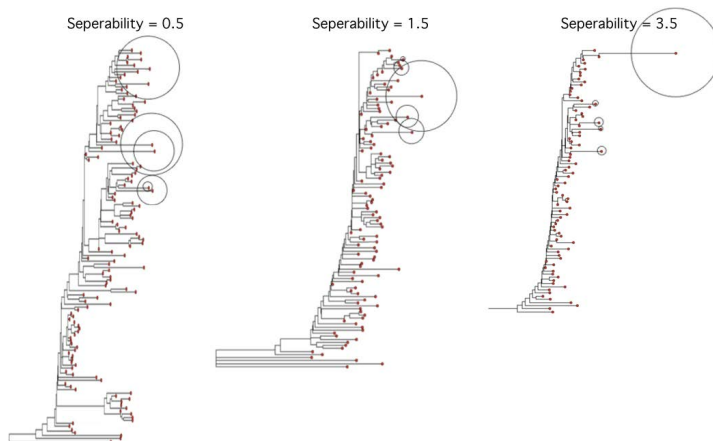
Unlike in previous work that samples from discrete state spaces, several new issues arise in plotting the ELMs of continuous hypothesis spaces. For example, many of the basins in the hypothesis space have a flat bottom which may result in a large number of false local minima, and thus we merge local minima identified by gradient descent based on the following criteria: (1) the distance between two local minima is smaller than a constant  $\epsilon$ ; or (2) there is no barrier along the straight line between two local minima. Besides, there may be constraints between parameters (e.g., a probability vector should lie on the surface of a unit simplex), and thus we may need to run our algorithm on a manifold. More details of our algorithm can be found in [8].

### 3 ELMs of Gaussian Mixture Models

An  $n$ -component Gaussian Mixture Model (GMM) is a weighted mixture of  $n$  Gaussians. The energy function of data clustering using GMM is the negative log of the posterior, given by  $E(x) = -\log P(x|z_i : i = 1 \dots m) - \log P(x)$  for  $m$  input data examples  $\{z_i\}$ . We use a Dirichlet prior on the weights of the model and the Normal-inverse-Wishart prior on the means and variances of the model components.

#### 3.1 Experiments on Synthetic Data

We synthesize a 2-dimensional, 3-component GMM, draw  $m$  samples from it, and run our algorithm to plot the ELM. We want to analyze how the separability  $c$  affects the energy landscape. The separability of the GMM represents the overlap between separate components of the model and is defined as  $c = \min\left(\frac{\|\mu_i - \mu_j\|}{\sqrt{n} \max(\sigma_1, \sigma_2)}\right)$  [4]. We also look at the effect of partial supervision on the energy landscape by assigning ground truth labels to a fraction of the samples.

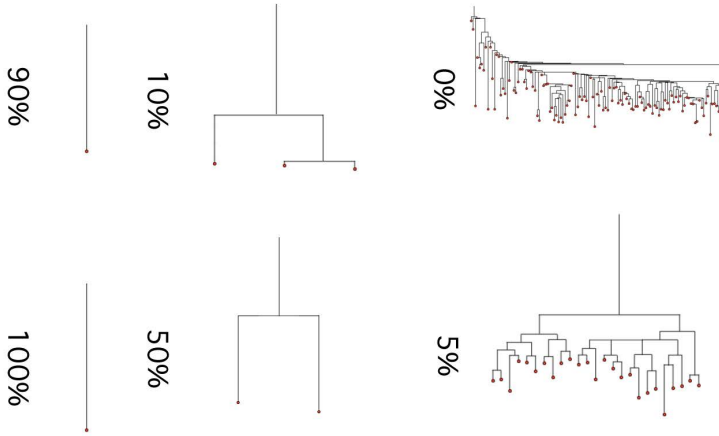


**Fig. 3.** ELMs for 100 samples drawn from GMMs with low, medium and high separability ( $c = 0.5, 1.5, 3.5$ ). The relative probability mass of the energy basins corresponding to the 5 lowest-energy minima are indicated by circle size around the local minima.

**Comparing Different Ground-Truth Models.** Figure 3 shows some of the ELMs with the separability being  $\{0.5, 1.5, 3.5\}$  for  $m = 100$  samples. The energy landscape becomes increasingly simple (containing fewer local minima) as the separability increases. The landscape for the high separability ( $c = 3.5$ ) case has relatively small energy barriers between the high-energy local minima and a pronounced low-energy global minimum. Conversely, the landscape for the low separability has a structure with high energy barriers between local minima and multiple local minima with similar energy to the global minimum. This indicates that the complexity of learning the GMM model should increase as the separability decreases, as we would expect.

The probability mass of the 5 energy basins corresponding to the lowest-energy local minima are shown in Figures 3 by the circles (similarly we can also show the volume of each basin). The ratio of the mass of the lowest energy basin to the mass of the remaining energy basins increases with separability. This is also consistent with the intuition that high-separability landscapes have lower complexity, as it is more likely that the global optimal solution can be found by gradient descent from a randomly sampled starting point.

We examine the affects of partial supervision by assigning ground truth labels (i.e. which Gaussian cluster a point belongs to) to a portion of the data samples. Figure 4 shows the ELMs of a synthesized GMM (dimension = 2, number of components = 3, separability  $c = 1.0$ , number of samples = 100) with  $\{0\%, 5\%, 10\%, 50\%, 90\%, 100\%\}$  labelled data points. Figure 5 shows the number of local minima in the ELM for the labeling of  $1, \dots, 100$  samples. This shows a significant decrease in landscape complexity for the first 10 labels, and diminishing returns from supervised input after the initial 10%.



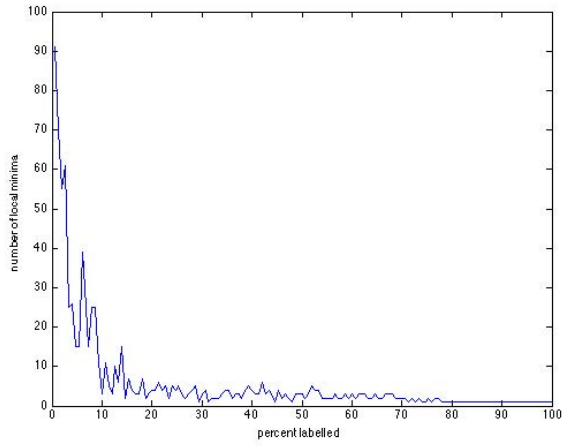
**Fig. 4.** ELMs of synthesized GMMs (separability  $c = 1.0$ ,  $n_{\text{Samples}} = 100$ ) with  $\{0\%, 5\%, 10\%, 50\%, 90\%, 100\%\}$  labelled data points

#### Behavior of Learning Algorithms: EM, K-mean and SW-Cut.

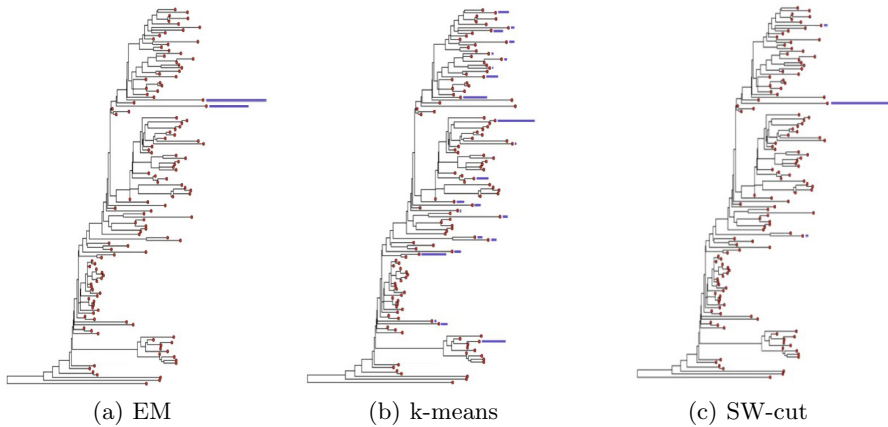
Expectation-maximization (EM) is one of the most popular algorithms for learning a GMM from data. K-means is another popular learning algorithm of GMM which can be seen as a degraded variant of EM with hard assignments in the E-step and the assumption of identical spherical Gaussian components. The Swendsen-Wang Cut (SW-cut) algorithm [1] is a generalization of the Swendsen-Wang method [11] to arbitrary probabilities. It is a MCMC method that has much faster convergence rates than classic Markov Chain Monte Carlo methods such as the Gibbs sampler in cases when model states are strongly coupled (such as the Ising-Potts model) [9].

For each synthetic dataset, we ran the three algorithms for 200 times and found the energy basins of the ELM that the learned models belong to. Hence we obtain a histogram of the learned models on the leaf nodes of the ELM for each learning algorithm as shown in Figure 6–7.

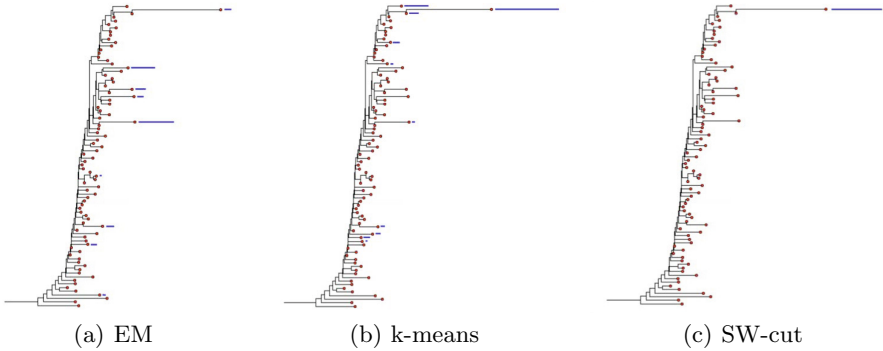
Figures 6 and 7 show a comparison of the EM, K-mean, and SW-cut algorithms for  $n = 100$  samples drawn from low ( $c = 0.5$ ) and high ( $c = 3.5$ ) separability GMMs. The SW-cut algorithm performs best in each situation, always converging to the global optimal solution. In the low separability case, the K-mean algorithm converges to one of the seven local minima, with a higher probability of converging to those with lower energy. The EM algorithm almost always finds the global minimum and thus outperforms K-mean. This can be explained by the fact that K-mean is a degraded variant of EM with extra assumptions that may not hold. However, in the high separability case, the K-mean algorithm converges to the true model the majority of the time, while the EM almost always converges to a local minimum with higher energy than the true



**Fig. 5.** Number of local minima versus the percentage of labelled data points for a GMM with separability  $c = 1.0$



**Fig. 6.** Low separability  $c = 0.5$ : histogram of EM, k-means, and SW-cut algorithm results on the ELM



**Fig. 7.** High separability  $c = 3.5$ : histogram of EM, k-means, and SW-cut algorithm results on the ELM

model. This can be explained by a recent theoretical result showing that the objective function of hard-EM (with k-means as a special case) is the summation of the standard energy function of GMM with an inductive bias in favor of high-separability models [12,10].

### 3.2 Experiments on Real Data

We ran our algorithm to plot the ELM for the well-known Iris data set from the UCI repository [3]. The data set contains 150 points in 4 dimensions and can be modeled as a 3-components 4-dimensional GMM. The three components each represent a type of iris plant and the true component labels are known. The points corresponding to the first component are linearly separable from the others, but the points corresponding to the remaining two components are not linearly separable.

Figure 8 shows the ELM of the Iris dataset. We visualize the local minima by plotting the ellipsoids of the covariance matrices centered at the means of each component in 2 of the 4 dimensions.

The 6 lowest energy local minima are shown on the right and the 6 highest energy local minima are shown on the left. The high energy local minima are less accurate models than the low energy local minima. The local minima (E) (B) and (D) have the first component split into two and the remaining two (non-separable) components merged into one. The local minima (A) and (F) have significant overlap between the 2nd and 3rd components and (C) has the components overlapping completely. The low-energy local minima (G-L) all have the same 1st components and slightly different positions of the 2nd and 3rd components.

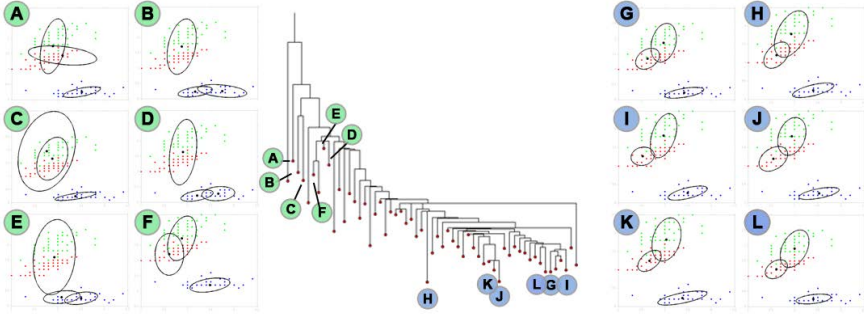


Fig. 8. ELM of the Iris dataset and corresponding local minima

## 4 Learning Mixtures of Bernoulli Templates

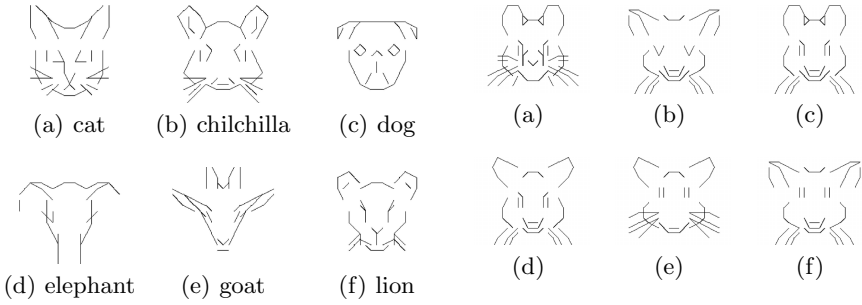
An object image can be converted to a dense edge map or a sparse sketch map using Gabor filters. We can quantize the edges/sketches into finite locations and orientations, and thus each input image is transformed to a binary vector. A Bernoulli template  $P \in \{0, 1\}^n$  is an  $n$ -dimensional binary vector. A sample  $x$  is generated from  $P$  with independent Bernoulli noise: the  $i$ -th coordinate  $x_i$  is equal to  $P_i$  with a fixed probability  $p$  and equal to  $1 - P_i$  with probability  $1 - p$ . An  $K$ -component Mixture of Bernoulli Templates (MBT)  $B$  is a weighted mixture of  $K$  Bernoulli templates defined by the set of templates  $\{P_i\}$  and weights  $\{w_i | w_i \in [0, 1]\}$  for  $i \in \{0, \dots, K\}$  with  $\sum w_i = 1$ . Samples  $s_j$  are drawn from  $B$  by first sampling a component  $P_i$  from the discrete distribution of weights  $\{w_i\}$ , then sampling from the template  $P_i$  as outlined above. We wish to compute the energy landscape map of the space of MBTs with a fixed noise level  $p$ . The energy function that we use is the negative log of the posterior, given by  $E(B) = -\log P(B | z_i : i = 1 \dots M)$  for  $M$  samples  $\{z_i\}$ . The probability of a sample  $z_i$  given a MBT is defined as:

$$P(z_i | B) = \sum_{i=1}^m w_i p^{\sum_{j=1}^n I(z_i(j)=P_i(j))} (1-p)^{\sum_{j=1}^n I(z_i(j) \neq P_i(j))},$$

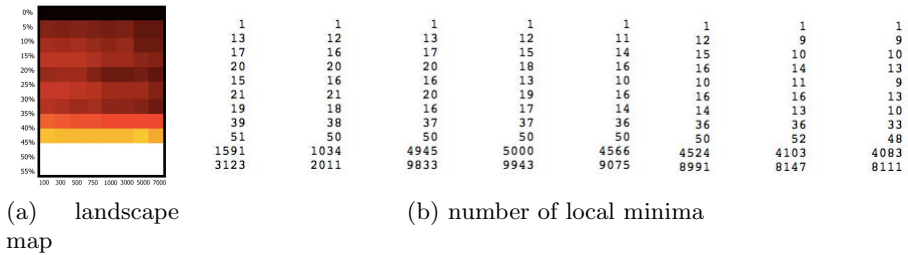
where  $P_i(j)$  is the  $j$ -th component of the  $i$ -th Bernoulli template in  $B$ , and  $z_i(j)$  is the  $j$ -th component of the  $i$ -th sample. When constructing the ELMs, we discretize the hypothesis space by allowing the weights to take values  $w_i \in \{0, 0.1, \dots, 1.0\}$ .

### 4.1 Experiment on Synthetic Data

We synthesized Bernoulli templates which represent animal faces as show in Figure 9. Each animal face is a  $9 \times 9$  grid with each cell containing up to 3 sketches. The dictionary of sketches contains 18 elements, each of which is a straight line



**Fig. 9.** Animal face templates - low overlap **Fig. 10.** Mouse face templates - high overlap



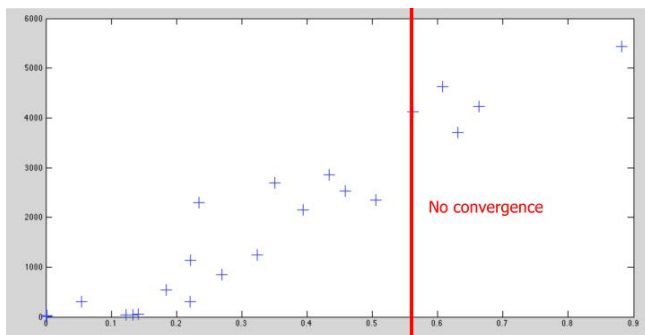
**Fig. 11.** The number of local minima in the energy landscape of learning MBT with varying values of noise level  $p$  and number of samples

connecting the endpoints or midpoints of the cell edges. The Bernoulli template can therefore be represented as a  $18 \times 9 \times 9$  dimensional binary vector. There are 10 animals in total, so we have a Bernoulli mixture model with the number of component  $M = 10$ .

We construct the energy landscape maps of the Bernoulli mixture model for varying numbers of samples  $n = 100, 300, \dots, 7000$  and varying noise level  $p = 0, 0.05, \dots, 0.5, 0.55$ . The number of local minima in each energy landscape is tabulated in Figure 11 (b) and drawn as a heat map in Figure 11 (a). As expected, the number of local minima increases as the noise level  $p$  increases, and decreases as the number of samples decreases. In particular, with no noise, the landscape is convex and with noise  $p > 0.45$ , there are too many local minima and the algorithm does not converge.

We repeat the same experiment using variants of a mouse face as shown in Figure 10. We swap out components of the mouse face (the eyes, ears, whiskers, nose, mouth, head top and head sides) for three different variants. We thereby generate 20 Bernoulli templates which have relatively high degrees of overlap. We generate the ELMs of various MBTs containing three of the 20 templates with noise level  $p = 0$ . In each MBT, the three templates have different degrees of overlap. Hence we plot the number of local minima in the ELMs versus the





**Fig. 12.** Number of local minima found for varying degrees of overlap in the Bernoulli templates

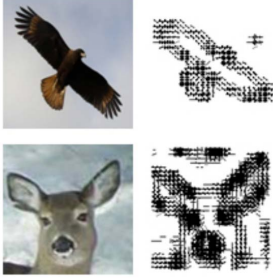
degree of overlap as show in Figure 12. As expected, the number of local minima increases with the degree of overlap, and there are too many local minima for the chains to converge past overlap  $c = 0.5$ .

## 4.2 Experiment on Real Data

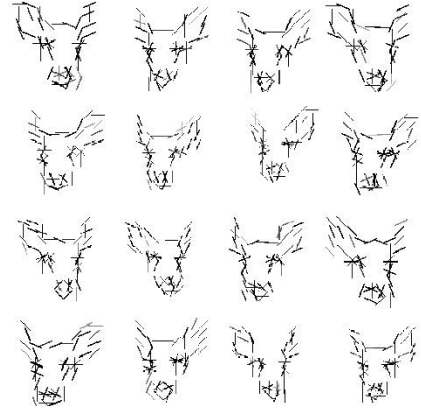
We perform the Bernoulli templates experiment on a set of real images of animal faces. We binarize the images by extracting the prominent sketches on a  $9 \times 9$  grid. Eight Gabor filters with eight different orientations centered in the centers and corners of each cell are applied to the image. The filters with a strong response above a fixed threshold correspond to edges detected in the figure; these are mapped to the dictionary of 18 elements. Thus each animal face is represented as a  $18 \times 9 \times 9$  dimensional binary vector. The Gabor filter responses on animal face pictures are shown in Figure 13. The binarized animal faces are shown in Figure 14.

We chose 3 different animal types – deer, dog and cat, with an equal number of images chosen from each category (Figure 15). The binarized versions of these can be modeled as a mixture of 3 Bernoulli templates - each template corresponding to one animal face type.

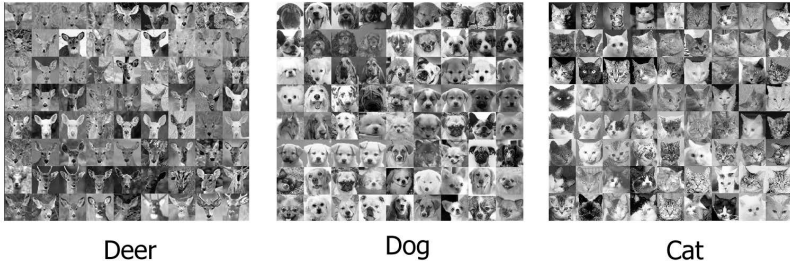
The ELM is shown in Figure 16 along with the Bernoulli templates corresponding to three local minima separated by large energy barriers. We make two observations: 1. the templates corresponding to each animal type are clearly identifiable, and therefore the algorithm has converged on reasonable local minima. 2. The animal faces have differing orientations across the local minima (the deer face on in the left-most local minimum is rotated and tilted to the right and the dog face in the same local minimum is rotated and lilted to the left), which explains the energy barriers between them.



**Fig. 13.** Animal face images and corresponding binary sketches indicates the existence of a Gabor filter response above a fixed threshold

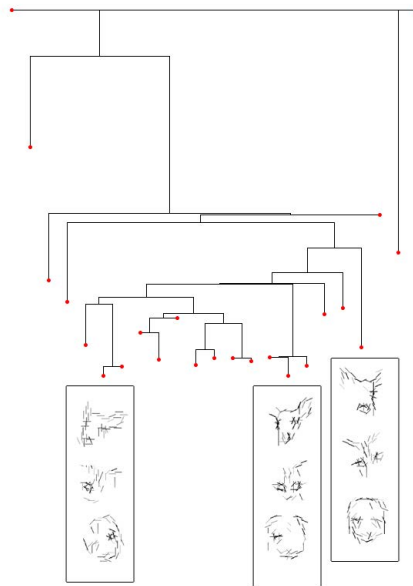


**Fig. 14.** Deer face sketches binarized from real images

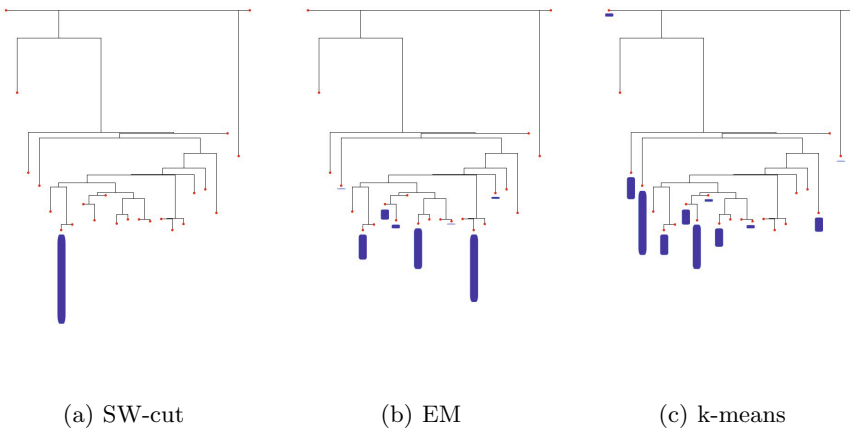


**Fig. 15.** Animal face images

Figure 17 shows a comparison of the SW-cut, k-means, and EM algorithm performance as a histogram on the ELM of animal face Bernoulli Mixture Model. The histogram is obtained by running each algorithm 200 times with a random initialization, then finding the closest local minimum in the ELM to the output of the algorithm. The counts of the closest local minima are then displayed as a bar plot next to each local minimum. It can be seen that SW-cut always finds the global minimum, while k-means performs the worst probably because of the high degree of overlap between the sketches of the three types of animal faces.



**Fig. 16.** ELM of three animal faces (dog, cat, and deer). We show the Bernoulli templates corresponding to three local minima with large energy barriers.



**Fig. 17.** Comparison of SW-cut, k-means, and EM algorithm performance on the ELM of animal face Bernoulli Mixture Model

## 5 Conclusion

We present a method for computing the energy landscape maps (ELMs) in hypothesis spaces and thus visualize for the first time the non-convex energy minimization problems in computer vision, pattern recognition and statistical learning. We demonstrate the methods in two cases: clustering with Gaussian mixture models in low dimensional space, and learning mixtures of Bernoulli templates from images in very high dimensional space. By plotting the ELMs, we have shown how different problem settings, such as separability and levels of supervision, impact the complexity of the energy landscape. We have also examined the behaviors of different learning algorithms in the ELMs. More experimental results and analysis can be found in our technical report [8].

**Acknowledgments.** The authors thank Dr. Qing Zhou for his tutorial on the algorithm and many helpful suggestions, thank Drs. Yingnian Wu and Adrian Barbu for their discussions, and acknowledge the support of a DARPA MSEE project FA 8650-11-1-7149.

## References

1. Barbu, A., Zhu, S.C.: Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1239–1253 (2005)
2. Becker, O.M., Karplus, M.: The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *The Journal of Chemical Physics* 106(4), 1495–1517 (1997)
3. Blake, C.L., Merz, C.J.: *UCI repository of machine learning databases* (1998)
4. Dasgupta, S., Schulman, L.J.: A two-round variant of em for gaussian mixtures. In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, UAI 2000*, pp. 152–159. Morgan Kaufmann Publishers Inc., San Francisco (2000)
5. Gelman, A., Rubin, D.B.: Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science* 7(4), 457–472 (1992)
6. Liang, F.: Generalized wang-landau algorithm for monte carlo computation. *Journal of the American Statistical Association* 100, 1311–1327 (2005)
7. Liang, F.: A generalized wang-landau algorithm for monte carlo computation. *JASA. Journal of the American Statistical Association* 100(472), 1311–1327 (2005)
8. Pavlovskaia, M., Tu, K., Zhu, S.C.: Mapping energy landscapes of non-convex learning problems. arXiv preprint arXiv:1410.0576 (2014)
9. Potts, R.B.: Some Generalized Order-Disorder Transformation. *Transformations, Proceedings of the Cambridge Philosophical Society* 48, 106–109 (1952)
10. Samdani, R., Chang, M.W., Roth, D.: Unified expectation maximization. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 688–698. Association for Computational Linguistics (2012)
11. Swendsen, R.H., Wang, J.S.: Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters* 58(2), 86–88 (1987)

12. Tu, K., Honavar, V.: Unambiguity regularization for unsupervised learning of probabilistic grammars. In: Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning, EMNLP-CoNLL 2012 (2012)
13. Wang, F., Landaul, D.: Efficient multi-range random-walk algorithm to calculate the density of states. *Phys. Rev. Lett.* 86, 2050–2053 (2001)
14. Zhou, Q.: Random walk over basins of attraction to construct ising energy landscapes. *Phys. Rev. Lett.* 106, 180602 (2011)

# Marked Point Process Model for Curvilinear Structures Extraction

Seong-Gyun Jeong, Yuliya Tarabalka, and Josiane Zerubia

Inria, Ayin team, 2004 Route des Lucioles, 06902 Sophia-Antipolis, France  
`{firstname.lastname}@inria.fr`

**Abstract.** In this paper, we propose a new *marked point process* (MPP) model and the associated optimization technique to extract curvilinear structures. Given an image, we compute the intensity variance and rotated gradient magnitude along the line segment. We constrain high level shape priors of the line segments to obtain smoothly connected line configuration. The optimization technique consists of two steps to reduce the significance of the parameter selection in our MPP model. We employ Monte Carlo sampler with delayed rejection to collect line hypotheses over different parameter spaces. Then, we maximize the consensus among line detection results to reconstruct the most plausible curvilinear structures without parameter estimation process. Experimental results show that the algorithm effectively localizes curvilinear structures on a wide range of datasets.

**Keywords:** curvilinear structure extraction, marked point process, Monte Carlo sampling with delayed rejection, aggregation algorithm.

## 1 Introduction

Curvilinear structures are widely observed in natural scenes. Thus, it is an important task to detect lines in many computer vision applications. For example, road network extraction algorithms [18, 30] have been developed for remote sensing. To find defects of the road pavement, an adaptive filtering and image segmentation algorithm has been proposed in [5]. For medical application, blood vessel detection [8, 25] aids diagnosis of disease. Localization of facial wrinkles [2, 16] provides visual cue of aging. However, these algorithms have a limitation of the use on different domains because the corresponding models of curvilinear structures have been specifically designed for their target applications.

Since linear structures correspond to image gradient information, image filtering with higher order derivatives [9, 15] is successful to grasp such image characteristics. Pixelwise segmentation for linear structure extraction measures the linearity for each pixel, and then sets up a threshold to remove out redundant outcomes [5, 8, 25, 30]. Supervised learning algorithm [3] has been proposed to find optimal convolution kernels for extracting linear features. Mathematical morphology operator [27] can enhance thin line structures based on shape information. However, criteria used for choosing the threshold values are ambiguous

if the image gradient information is corrupted by noise or rough textures. Tree-like representation [12, 29] has been recently proposed to automatically extract curvilinear structures. These algorithms initially define a set of seed points, and then grow branches based on a local tubularity measure [19]. However, the tree representation requires heavy computations, and a localization of seed points is crucial for the final result.

Curvilinear structures can be seen as a combination of small line segments. Sampling techniques with geometric priors have been exploited to detect multiple line segments in a scene [2, 16, 18, 22, 24, 28]. The *marked point process* (MPP) framework [6, 7, 20, 26] is helpful to enforce high level constraints on shape prior. However, the MPP model requires heavy formalization to interpret spatial distribution of the objects. Large number of parameters should be defined to describe the geometric shape of the objects (*modeling parameters*) and to control the relative importance of data and prior energy terms (*hyperparameters*). MPP modeling has been considered less practical to solve general problem because the performance is very sensitive for the selection of parameters. Although *stochastic expectation maximization* algorithm [4, 21] has been used to estimate modeling parameters, it exhibits both speed and scalability issues.

Although the contour grouping algorithms [1, 28] also examine image features corresponding to curves and lines, the goal is quite different from the curvilinear structures extraction techniques. The contour grouping algorithms seek closed contour lines to divide an image into meaningful regions. On the other hand, we look for multiple curvilinear structures, which are not necessarily closed, within a homogeneous texture. While the contours are associated with salient edges around objects boundaries, the curvilinear structures are subtle local image features in the same plane. Unlike [1], we cannot exploit global texture cues for the data energy term; therefore, an accurate design of the shape prior energy is essential to solve our problem.

In this paper, we propose a new MPP model for curvilinear structures extraction in a fully automatic way, where the performance is not biased by the hyperparameter selection. Indeed, our MPP model can detect wide types of input data without a sophisticated parameter tuning process. To fit in with a dataset, we analyze image gradients and homogeneity of intensities along the line segments. The prior energy is defined on local configuration to implement smooth connection among line segments (Sec. 2). To avoid the burden of hyperparameter selection, we first generate multiple candidates of the line configuration with different hyperparameter settings. Markov chain Monte Carlo sampler [11, 13, 14, 23] with delayed rejection scheme [14] is employed to optimize the proposed probability density function. Next, we combine the whole set of line candidates in a way that maximizes the consensus among line detection results (Sec. 3). Extensive experiments on various datasets including facial wrinkles, road cracks, DNA filaments, and blood vessels demonstrate the effectiveness of the proposed MPP model for extracting thin curvilinear structures (Sec. 4).

## 2 Marked Point Process Modeling

### 2.1 MPP Revisited

We briefly review the definition of MPP [20, 26] to provide a mathematical description of the proposed model.

**Definition 1 (Spatial point process).** *A realization of point process consists of an unordered set of points in a compact set  $\mathcal{F} \subset \mathbb{R}^d$ . A point process on  $\mathcal{F}$  maps from a measurable probability space  $(\mathcal{F}, \mathcal{B}, \mu)$  onto the configuration space  $\Omega = \cup_{n=0}^{\infty} \Omega_n$ , where  $\mathcal{B}$  denotes  $\sigma$ -algebra of subset of  $\mathcal{F}$ , and  $\mu$  is the Lebesgue measure. In other words, for all bounded Borel sets  $B \subseteq \mathcal{B}$ , the number of points falling in  $B$  is a finite random variable.*

**Definition 2 (Marked point process).** *In the MPP framework, each point is associated with additional information which describes a shape of the object. Specifically, we reconstruct curvilinear structures as smoothly connected line segments. Let  $s_i = (\mathbf{x}_i, \mathbf{m}_i)$  be a line segment specifying its center point  $\mathbf{x}_i = (x_i, y_i)$  in the image sites  $\mathcal{F}$  with a label of the length and the orientation  $\mathbf{m}_i = (\ell_i, \theta_i)$ , where the label is sampled from the mark space  $M$  with a probability measure  $\mu_M$ . We now define a marked point process on  $\mathcal{F} \times M$  as a finite random configuration  $\mathbf{s} = \{s_1, \dots, s_n\} \in \Psi$ .*

The probability distribution of the MPP is defined based on an image  $I$  and spatial interactions between line segments. Given an image, we look for an optimal configuration  $\hat{\mathbf{s}}$  which maximizes the unnormalized probability density  $f(\mathbf{s})$  as follows:

$$\hat{\mathbf{s}} = \operatorname{argmax}_{\mathbf{s} \in \Psi} f(\mathbf{s}) = \operatorname{argmin}_{\mathbf{s} \in \Psi} \sum_{i=1}^{\#(\mathbf{s})} U_d(s_i) + \sum_{i \sim j} U_p(s_i, s_j), \quad (1)$$

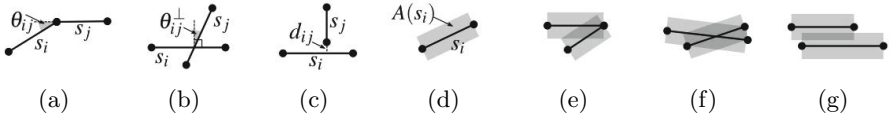
where  $\#(\mathbf{s})$  is the cardinality of the configuration, and  $i \sim j$  represents the symmetry relationship between interacting line segments  $s_i$  and  $s_j$ .  $U_d$  and  $U_p$  denote the data likelihood and the prior energy, respectively. In general, Monte Carlo samplers [10, 13, 14, 31] are employed in MPP models to maximize the proposed density function  $f(\mathbf{s})$ . Each state of a discrete Markov chain  $(X_t)_{t \in \mathbb{N}}$  corresponds to a random configuration on the  $\Psi$ . The chain is locally perturbed by transition kernels, and is evolved to converge to the stationary distribution which is identical to the proposed probability density.

### 2.2 Data Likelihood

We define the data likelihood of the line segment  $s_i$  as a weighted sum of the rotated gradient magnitudes  $U_d^m$  and the intensity variance  $U_d^v$  along the line:

$$U_d(s_i) = \omega_d^m U_d^m(s_i) + \omega_d^v U_d^v(s_i), \quad (2)$$





**Fig. 1.** Examples of the line configurations with different prior energies: (a)–(c) show preferable line configurations composed of aligned lines (a), almost perpendicular lines (b), and adjacent lines (c). (d)–(g) depict unfavourable line configurations which are penalized because of a singular segment (d), acute corner (e), overlap (f), and parallel (g), respectively.

where  $\omega_d^m$  and  $\omega_d^v$  are weighting coefficients corresponding to  $U_d^m$  and  $U_d^v$ , respectively.

We obtain the rotated gradient information by convolving the input image with steerable filters [9, 15]. Steerable filters are generated from a linear combination of basis filters. In this work, we use second-order derivatives of an isotropic Gaussian function as the basis filters. Let  $g_{\theta_i}(\mathbf{x}; \sigma^2)$  be a steerable filter associated with an orientation  $\theta_i$  and  $\nabla I_{\theta_i} = g_{\theta_i} * I$  be its filtering response, which adaptively accentuates gradient magnitudes corresponding to the angle  $\theta_i$ . Then, the gradient magnitude energy  $U_d^m$  is defined as

$$U_d^m(s_i) = \int_0^1 |\nabla I_{\theta_i}(\mathbf{p}_i(t))| dt, \tag{3}$$

where  $\mathbf{p}_i(t)$  represents points on the line segment  $s_i$ . Note that  $\mathbf{p}_i(t) = (1 - t)\mathbf{u}_i + t\mathbf{v}_i$  is a function of the endpoints  $\mathbf{u}_i$  and  $\mathbf{v}_i$  with parameter  $t \in [0, 1]$ .

When the input image is heavily corrupted by noise or composed of uneven textures, observing gradient distribution often fails to detect linear structures. To ease this problem, we also measure the intensity variance along the line segment. This is because intensities are likely to be homogeneous, if pixels are laid on the same line. We can write:

$$U_d^v(s_i) = \frac{1}{\ell_i} \int_0^1 (I(\mathbf{p}_i(t)) - \mathbb{E}[I(s_i)])^2 dt, \tag{4}$$

where  $\mathbb{E}[I(s_i)]$  denotes the intensity mean of the line segment  $s_i$ , and  $\ell_i$  is the line length.

### 2.3 Prior Energy

In this section, we propose the prior energy to define spatial interactions on a local configuration. We want to obtain smoothly connected lines with a small curvature as a final solution. We compute the overlapping area  $\Upsilon(s_i, s_j)$  to reject congestion of lines and the coupling energy states  $\mathbf{c}_{ij}$  to evaluate attraction between line segments (see Fig. 1). The prior energy  $U_p(s_i, s_j)$  is defined as

$$U_p(s_i, s_j) = \Upsilon(s_i, s_j) + \mathbf{w}_p^T \mathbf{c}_{ij}, \quad \forall i \sim j, \tag{5}$$

where  $\mathbf{w}_p$  denotes a vector of weighting factors which control relative importance of each element in  $\mathbf{c}_{ij}$ . We assume that a line segment only correlates with the other ones within a certain distance. Thus, a neighborhood system consists of pairs of line segments, such that their center distance is smaller than half the sum of their lengths. In other words,

$$i \sim j = \left\{ (s_i, s_j) \in \Psi^2 : 0 < \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq \frac{\ell_i + \ell_j}{2} + \epsilon \right\}, \quad (6)$$

where  $\epsilon$  denotes the marginal distance to be connected with each other.

In order to evaluate an overlapping area between line segments, we dilate the line segments with a three pixel-radius disk, and then count up the number of pixels falling in the same image site. Suppose that we have a set of points  $A(s_i)$  which is a dilated version of the line segment  $s_i$ , and  $n(A(s_i))$  denotes the number of pixels in  $A(s_i)$ . As shown in Fig. 1 (e)–(g), we penalize a configuration  $\{s_i, s_j\}$ , when a portion of the overlapping area is greater than 10% of  $\min\{n(A(s_i)), n(A(s_j))\}$ . However, almost perpendicular line segments are excluded from this penalty. The criteria for rejection are then given as

$$\Upsilon(s_i, s_j) = \begin{cases} 0 & \text{if } \theta_{ij}^\perp < \tau, \\ 0 & \text{if } \frac{n(A(s_i) \cap A(s_j))}{\min\{n(A(s_i)), n(A(s_j))\}} < 0.1, \\ \infty & \text{otherwise,} \end{cases} \quad (7)$$

where  $\theta_{ij}^\perp = \frac{\pi}{2} - \theta_{ij}$  represents an angle difference between  $s_i$  and the perpendicular line of  $s_j$ ,  $\tau$  is the maximum angle difference for segments to be aligned.

The coupling energy states  $\mathbf{c}_{ij}$  of the lines are composed of the singularity, connectivity, curvature, and perpendicularity:

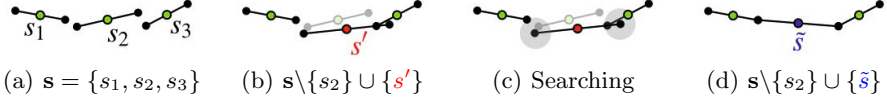
$$\mathbf{c}_{ij} = [1, \varphi(d_{ij}, \epsilon), \varphi(\theta_{ij}, \tau), \varphi(\theta_{ij}^\perp, \tau)]^\top, \quad \varphi(u, v) = \min\{0, (u/v)^2 - 1\}, \quad (8)$$

where  $d_{ij}$  denotes the minimum distance from endpoints of  $s_i$  to a point on the line  $s_j$ , and  $\theta_{ij}$  is the angle difference between line segments. The function  $\varphi(u, v)$  tests a firmness of the coupling state  $u$  by comparing with the given tolerance value  $v$ .

The weighting factors  $\mathbf{w}_p = [\omega_p^s, \omega_p^c, \omega_p^a, \omega_p^r]^\top$  can be derived from their role in the prior energy. Specifically,  $\omega_p^s$  penalizes birth of a single line segment in the final configuration; hence its value is affected by the average gradient magnitude and the noise level of the input.  $\omega_p^c$  encourages adjacent segments within  $\epsilon$  to become connected.  $\omega_p^a$  promotes segments being aligned with a small curvature in the final configuration.  $\omega_p^r$  supports perpendicularly approaching line segments. Although the selection of  $\mathbf{w}_p$  values is critical for the performances of the MPP model, it is hard to estimate the coefficients because of hidden dependencies among them.

## 2.4 Monte Carlo Sampler with Delayed Rejection

We employ the *Reversible jump Markov chain Monte Carlo* (RJCMC) sampler [13] to obtain an optimal line configuration which maximizes the probability



**Fig. 2.** Given configuration (a), if a line segment  $s'$  proposed by LT kernel is rejected (b), the delayed rejection kernel searches for the nearest extremes in the rest of line segments (c). An alternative line segment  $\tilde{s}$ , which enforces connectivity, will be proposed by interpolation of the retrieved points (d).

density function. The RJMCMC sampler is an iterative method that locally perturbs a current configuration  $\mathbf{s}$  with a transition kernel. The transition kernel consists of multiple sub-transition kernels, namely, *birth-and-death* (BD) and *linear transform* (LT). A new configuration  $\mathbf{s}'$  is proposed according to the transition kernel, given by

$$\xi(\mathbf{s}, \mathbf{s}') = \sum_m p_m \xi_m(\mathbf{s}, \mathbf{s}'), \quad (9)$$

where  $p_m$  denotes a probability to choose  $m$ -th type of sub-transition kernel  $\xi_m(\mathbf{s}, \mathbf{s}')$ . For each sub-transition kernel, the detailed balance condition [13] is required to ensure the reversibility of the Markov chain. Acceptance ratio  $\alpha_m(\mathbf{s}, \mathbf{s}')$  is compared with a stochastic value  $\text{rand}[0, 1]$  to take a new configuration into account. The RJMCMC sampler is coupled with the *simulated annealing* (SA) algorithm [17] to secure the convergence of the Markov chain via relaxation parameter  $T$  (temperature); the temperature gradually decreases as the iteration goes on. To compute an acceptance ratio of the transition kernel, we use a density  $f(\mathbf{s})^{1/T}$  instead of  $f(\mathbf{s})$ . The acceptance ratio is

$$\alpha_m(\mathbf{s}, \mathbf{s}') = \min \left( 1, \frac{\xi_m(\mathbf{s}', \mathbf{s}) f(\mathbf{s}')^{1/T}}{\xi_m(\mathbf{s}, \mathbf{s}') f(\mathbf{s})^{1/T}} \right). \quad (10)$$

The BD kernel changes the dimensionality of the current configuration  $\mathbf{s}$  by adding a new line segment or removing an existing line segment. When the birth kernel proposes a new configuration  $\mathbf{s}' = \mathbf{s} \cup \{s\}$ , the length and the orientation of the new line segment are uniformly sampled from the mark space  $M = [\ell_{\min}, \ell_{\max}] \times [\theta_{\min}, \theta_{\max}]$ , where  $\ell_{\min}$  and  $\ell_{\max}$  are the minimum and maximum length of the line segment, respectively.  $\theta_{\min}$  and  $\theta_{\max}$  denote the minimum and maximum orientation of the line segment, respectively. Note that we refuse a birth of the line lying on *singular points*, which have zero gradient magnitudes. On the other hand, the death kernel removes a line segment which is randomly picked from the current configuration. Thus, a new configuration  $\mathbf{s}' = \mathbf{s} \setminus \{s\}$  is proposed by the death kernel. We compute the acceptance ratio of the birth kernel  $\alpha_B$  and the death kernel  $\alpha_D$  in the same way as proposed in [18], given by

$$\alpha_B(\mathbf{s}, \mathbf{s}') = \min \left( 1, \frac{p_D}{p_B} \frac{\mu(\mathcal{F})}{\#\mathbf{s} + 1} \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right), \quad (11)$$

$$\alpha_D(\mathbf{s}, \mathbf{s}') = \min \left( 1, \frac{p_B}{p_D} \frac{\#\mathbf{s}}{\mu(\mathcal{F})} \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right). \quad (12)$$

**Algorithm 1.** RJMCMC sampler with delayed rejection

---

```

1: Initialize:  $X_0 \leftarrow \mathbf{s}_0$  (or  $X_0 \leftarrow \emptyset$ ),  $t \leftarrow 0$ ,  $T \leftarrow T_0$ 
2: while  $T > T_{\min}$  do
3:    $\mathbf{s} \leftarrow X_t$ 
4:   Choose a transition kernel  $\xi_m$  according to probability  $p_m$ 
5:   Propose a new configuration  $\mathbf{s}'$  with  $\xi_m(\mathbf{s}, \mathbf{s}')$ 
6:   if  $\alpha_m(\mathbf{s}, \mathbf{s}') > \text{rand}[0, 1]$  then
7:      $X_{t+1} \leftarrow \mathbf{s}'$ 
8:   else
9:     Propose an alternative segment  $\tilde{\mathbf{s}}$  based on  $\xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}})$ 
10:    if  $\alpha_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) > \text{rand}[0, 1]$  then
11:       $X_{t+1} \leftarrow \tilde{\mathbf{s}}$ 
12:    else
13:       $X_{t+1} \leftarrow \mathbf{s}$ 
14:    end if
15:  end if
16:   $t \leftarrow t + 1$ 
17:  Decrease the temperature:  $T \leftarrow T_t$ 
18: end while

```

---

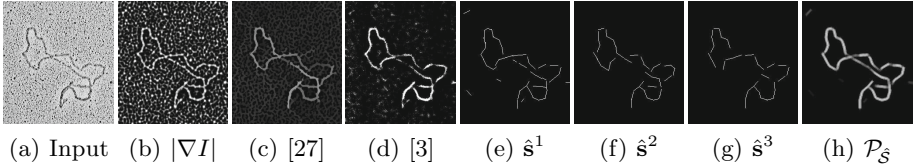
The LT kernel chooses a line segment  $s$  randomly, and then modifies its model parameters:  $s = (\mathbf{x}, (\ell, \theta)) \rightarrow s' = (\mathbf{x} \pm d\mathbf{x}, (\ell \pm d\ell, \theta \pm d\theta))$ , where  $d\mathbf{x}$ ,  $d\ell$ , and  $d\theta$  denote changes of center position, length, and orientation, respectively. The LT kernel draws a new configuration  $\mathbf{s}' = \mathbf{s} \setminus \{s\} \cup \{s'\}$ . The acceptance ratio of the LT kernel is defined by

$$\alpha_{\text{LT}}(\mathbf{s}, \mathbf{s}') = \min \left( 1, \frac{f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T}} \right). \quad (13)$$

The LT kernel can be extended by the delayed rejection scheme [14]. The main idea of the delayed rejection scheme is to give a second chance to a rejected sample point. The acceptance ratio of delayed rejection is defined by

$$\begin{aligned} \alpha_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) &= \min \left( 1, \frac{\xi_{\text{LT}}(\tilde{\mathbf{s}}, \mathbf{s}') \xi_{\text{LT}}^2(\mathbf{s}', \tilde{\mathbf{s}}, \mathbf{s}) [1 - \alpha_{\text{LT}}(\tilde{\mathbf{s}}, \mathbf{s}')] f(\tilde{\mathbf{s}})^{1/T}}{\xi_{\text{LT}}(\mathbf{s}, \mathbf{s}') \xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}}) [1 - \alpha_{\text{LT}}(\mathbf{s}, \mathbf{s}')] f(\mathbf{s})^{1/T}} \right), \\ &\simeq \min \left( 1, \frac{f(\tilde{\mathbf{s}})^{1/T} - f(\mathbf{s}')^{1/T}}{f(\mathbf{s})^{1/T} - f(\mathbf{s}')^{1/T}} \right). \end{aligned} \quad (14)$$

where  $\mathbf{s}' = \mathbf{s} \setminus \{s\} \cup \{s'\}$ ,  $\tilde{\mathbf{s}} = \mathbf{s} \setminus \{s\} \cup \{\tilde{s}\}$ , and  $\xi_{\text{LT}}^2(\mathbf{s}, \mathbf{s}', \tilde{\mathbf{s}})$  is the transition kernel for the delayed rejection. In order to reduce the burn-in time, we add heuristics to design the delayed rejection kernel. When we propose an alternative line segment  $\tilde{s}$ , we look for the closest endpoints from both ends of  $s'$ , which is rejected from the first trial. The line segment  $\tilde{s}$  is generated by interpolation of the retrieved points; we force the connectivity of the neighboring segments, so that a probability of being accepted increases in terms of prior energy. Fig. 2 summarizes the process of the delayed rejection kernel, and Algorithm 1 provides the pseudo-code of the RJMCMC sampler with delayed rejection.



**Fig. 3.** Given the input image (a), we compute the gradient magnitude (b). Mathematical morphology operator, path opening [27], is applied on such gradient magnitude image (c). Linearity score of each pixel is drawn by the supervised feature learning algorithm [3] (d). We provide line hypotheses (e)–(g) associated with different hyperparameter vectors. Composition result (h) is equivalent to mixture probability density, and it highlights pixels corresponding to linear structures.

### 3 Curvilinear Structure Extraction via Integration of Line Hypotheses

While the MPP allows to design complex prior knowledge of the object distribution, its performance is very sensitive to the selection of modeling parameters and hyperparameters. For clarity, we note that the modeling parameters are related to the physical characteristics of the line segments (*e.g.*, range of length and orientation). The hyperparameters denote the weighting coefficients of energy terms (*i.e.*,  $w_d^m$ ,  $w_d^v$ , and  $\mathbf{w}_p$ ). The modeling parameters can be chosen empirically since the values are related to the image resolution (see Sec. 4); however, it is hard to estimate the hyperparameters via trial-and-error for different types of dataset. Our goal is to maximize the probability density without estimating hyperparameters.

#### 3.1 Generation of $K$ Line Hypotheses

Let  $\mathbf{w} = [\omega_d^m, \omega_d^v, \omega_p^s, \omega_p^c, \omega_p^a, \omega_p^r]^\top$  be a hyperparameter vector which consists of the weighting coefficients of the proposed probability density. Suppose that we have  $K$  different hyperparameter vectors,  $\mathbf{w}^1, \dots, \mathbf{w}^K$ . For each hyperparameter vector, we substitute  $k$ -th hyperparameter vector  $\mathbf{w}^k$  into the proposed probability density  $f(\mathbf{s}; \mathbf{w}^k)$ . Then, we look for its optimal configuration  $\hat{\mathbf{s}}^k$  via Monte Carlo sampler proposed in Sec. 2.4.

For a practical reason related to the implementation, we bound the values of  $\mathbf{w}$ . Specifically, we sweep the weighting coefficients of the prior energy  $\mathbf{w}_p$  according to the gradient magnitude and noise level of the input image. Let  $\chi = -\ell_{\min} \times \mathbb{E}[\nabla I] + \text{Var}[I_{\sigma^2}]$  be a baseline to accept a new line segment into the current configuration without considering spatial interaction, where  $I_{\sigma^2}$  denotes a smoothed image using a Gaussian kernel with  $\sigma^2 = \{1.5, 2.25, 3.5\}$ . To reduce computation overhead, we fix the weighting factors of data likelihood energy as  $\omega_d^m = -1$  and  $\omega_d^v = 1$ . We set  $\mathbf{w}^1 = [-1, 1, \chi, 0.1\chi, 0.01\chi, 0.01\chi]^\top$ , and gradually change  $\chi$  by 10% of increments, *i.e.*,  $\mathbf{w}^2 = [-1, 1, \chi_2, 0.1\chi_2, 0.01\chi_2, 0.01\chi_2]^\top$ , where  $\chi_2 = 1.1\chi$ . In our experiments, we set  $K = 15$  to create line hypotheses.

### 3.2 Combination of Line Hypotheses into a Probability Map

We now have a family of line hypotheses  $\hat{\mathcal{S}} = \{\hat{\mathbf{s}}^1, \dots, \hat{\mathbf{s}}^K\}$  obtained from  $K$  different hyperparameter vectors. We jointly use the image data and the line hypotheses. More specifically, the final solution  $\mathbf{s}^*$  maximizes not only the probability density but also the consensus among line hypotheses. For each optimal configuration  $\hat{\mathbf{s}}^k$ , we compute a probability map  $\mathcal{P}_k$  of being a line in the image site. Then, we integrate  $K$  probability maps into a mixture density  $\mathcal{P}_{\hat{\mathcal{S}}}$ :

$$\mathcal{P}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in s_i^k, \\ \frac{1}{2} & \text{if } \exists s_i^k \in \hat{\mathbf{s}}^k, \mathbf{x} \in A(s_i^k), \\ 0 & \text{otherwise,} \end{cases} \quad \mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \mathcal{P}_k(\mathbf{x}). \quad (15)$$

Fig. 3 compares image gradient magnitude, morphological filtering [27], supervised feature learning [3], line hypotheses, and the mixture density. Since the input image contains many high frequency components, its gradient also highlights non-linear structures in the background. While the morphological filter accentuates linear structures, its performance depends on the setting of path length. Supervised learning method requires high quality of a training dataset and corresponding ground truth images. Depending on the setting of hyperparameter vectors, the MPP model leads incomplete detection results as shown in Fig. 3. (e)–(g). We integrate line hypotheses of the proposed MPP model into a mixture density  $\mathcal{P}_{\hat{\mathcal{S}}}$ . The mixture density shows the consensus between line hypotheses in the sense that the pixels corresponding to line structures are more highlighted when compared to [3, 27].

We assume that the most promising hyperparameter vector draws a configuration which is more akin to the mixture density. We compute the *correlation-coefficient* (CC) between  $\mathcal{P}_{\hat{\mathcal{S}}}$  and  $\mathcal{P}_k$ 's to analyze coherence of line detection results. That is

$$k^* = \operatorname{argmax}_{k=\{1, \dots, K\}} \operatorname{CC}(\mathcal{P}_{\hat{\mathcal{S}}}, \mathcal{P}_k), \quad (16)$$

$$\operatorname{CC}(\mathcal{P}_{\hat{\mathcal{S}}}, \mathcal{P}_k) = \frac{\sum_{\mathbf{x}} (\mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\hat{\mathcal{S}}}])(\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k])}{\sqrt{\sum_{\mathbf{x}} (\mathcal{P}_{\hat{\mathcal{S}}}(\mathbf{x}) - \mathbb{E}[\mathcal{P}_{\hat{\mathcal{S}}}]^2 \sum_{\mathbf{x}} (\mathcal{P}_k(\mathbf{x}) - \mathbb{E}[\mathcal{P}_k])^2}}, \quad (17)$$

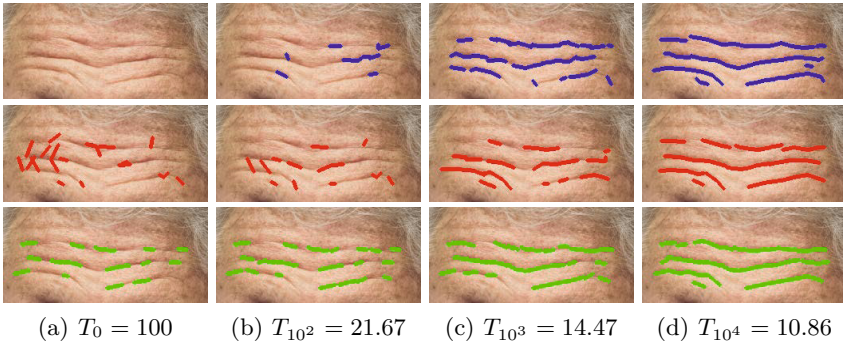
where  $k^*$  represents the index of the most reliable hyperparameter vector.

### 3.3 Curvilinear Structure Extraction from Reduced Sampling Space

The line hypotheses span a configuration space  $\mathbb{S} \subset \Psi$  which will be considered as a new sample space. Since the size of  $\mathbb{S}$  is significantly reduced compared to the original sample space  $\Psi$ , the optimization process becomes more tractable in terms of convergence time and detection accuracy.

We redefine the data likelihood energy by adding a new energy term as follows:

$$U'_d(s_i) = U_d(s_i) + U_d^h(s_i), \quad U_d^h(s_i) = \int_0^1 -\log \mathcal{P}_{\hat{\mathcal{S}}}(s_i(t)) dt, \quad (18)$$



**Fig. 4.** We provide intermediate sampling processes when the temperature parameter  $T_t$  is decreasing. The results shown in first row are obtained without specifying seed segment. For the second row, we randomly set 20 seed segments and run the algorithm. For the third row, we initialize 20 line segments which are highly corresponding to underlying curvilinear structures. The algorithm converges toward almost the same solution regardless of the initial state.

where  $U_d^h(s_i)$  quantifies the consensus among line hypotheses with respect to the line segment  $s_i$ . We stimulate the modified probability density over the reduced sample space  $\mathbb{S}$  with the most promising hyperparameter vector  $\mathbf{w}^{k^*}$ :

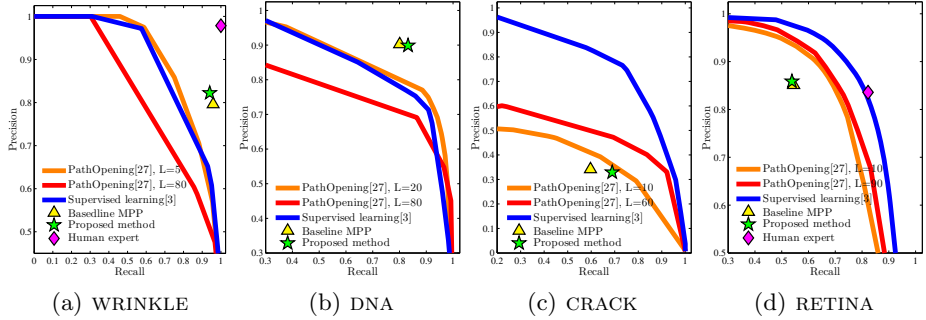
$$\mathbf{s}^* = \operatorname{argmin}_{\mathbf{s} \in \mathbb{S}} \sum_{i=1}^{\#(\mathbf{s})} U_d'(s_i) + \sum_{i \sim j} U_p(s_i, s_j; \mathbf{w}^{k^*}). \quad (19)$$

## 4 Experiments

We test the proposed algorithm on a wide range of datasets: facial wrinkles, DNA filaments<sup>1</sup>, road cracks, and retinas. The facial wrinkle images are collected on the Internet, and forehead areas are manually selected for the experiments. Test images of the defects on the road pavements and ground-truth are courtesy of Chambon *et al.* [5]. We use the DRIVE dataset [25] to test the proposed algorithm on retina images.

For all test sequences, we fix the modeling parameters as follows:  $\ell_{\min}$  is set to 5 pixels and  $\ell_{\max} = 20$  pixels. The orientation  $\theta$  is varying from  $-90^\circ$  to  $90^\circ$  with increments of  $2^\circ$ . The marginal distance of connected segments  $\epsilon$  is fixed to 2 pixels, and the maximum angular difference of aligned segments  $\tau$  is  $30^\circ$ . For the SA, the initial temperature  $T_0$  is set to 100, and it follows the logarithm cooling schedule  $T_t = T_0 / \log(1+t)$ , where  $t$  denotes the number of the current iteration. We start the sampling process with the empty configuration. However, careful choice of initial segments can speed up the convergence of the algorithm (see Fig. 4). The computational time depends on the image resolutions; it takes less than a minute for the experimental images having  $300 \times 400$  pixels, approximately. We use a PC with a 2.9 GHz CPU (4 cores) and 8 GB RAM.

<sup>1</sup> <https://www.biochem.wisc.edu/faculty/inman/empics/dna-prot.htm>



**Fig. 5.** Precision-and-recall curves for pixelwise segmentation of curvilinear structures using path opening operator [27] with different setups of length, supervised feature learning [3], baseline MPP, and the proposed method

To compare the performances of the proposed method with the state-of-the-art techniques, we apply the path opening operator [27] on the gradient magnitude images by controlling the length parameters. For the supervised feature learning algorithm [3], we train 15 images for each dataset. In our experiments, we use the original implementations of path opening operator<sup>2</sup> and supervised feature learning algorithm<sup>3</sup>.

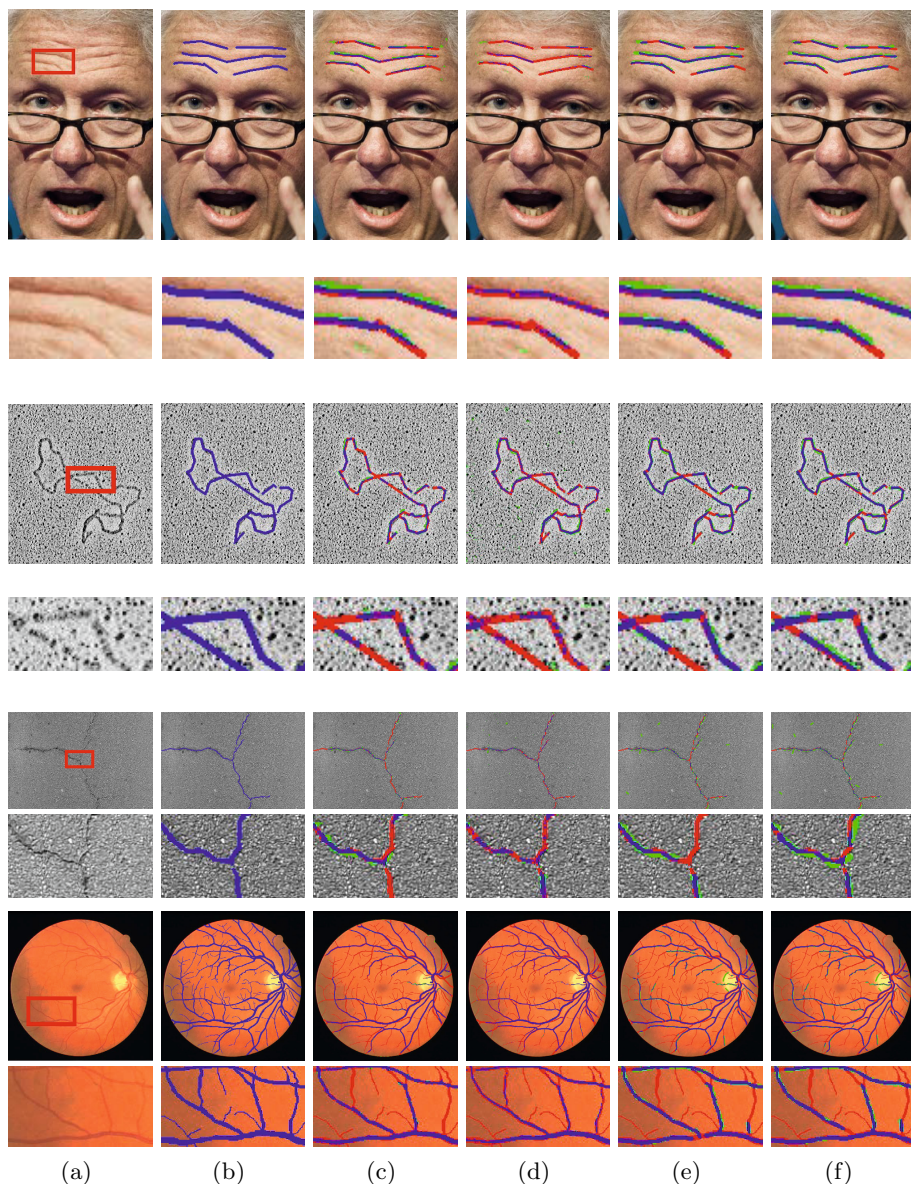
Fig. 5 shows the precision-and-recall curves for four test images. To obtain the curve of the comparison methods [3, 27], we tune thresholds on line detection results. The baseline MPP is selected from the line hypotheses among which it shows the best performance. The performances of the supervised learning algorithm are controlled by the quality of the training set; hence, it shows low performances on WRINKLE and DNA datasets, which are composed of noisy images with various sizes. In particular, the ground truth set of the WRINKLE dataset is based on subjective perception. While the morphology operator enhances linear structures on gradient magnitude images, it is required to specify the length of the linear structures according to the target applications. Since the pixelwise comparison fails to incorporate the geometry similarity with the ground-truth, the proposed algorithm shows lower scores on the CRACK and RETINA datasets. More specifically, the proposed algorithm detects slightly shifted lines for the CRACK image.

Fig. 6 compares the detection results of the proposed MPP model with the manually labeled image by human expert, morphology operator [27], supervised feature learning [3], and baseline MPP. For a fair comparison, we set the threshold values of the competing algorithms [3, 27] to obtain the closest recall scores to the proposed algorithm. Blue pixels denote perfectly matching regions as compared with the ground-truth. Green and red pixels show over-detected and under-detected results, respectively. The main strength of the proposed algorithm is that it ensures stable performances for all datasets without any parameter estimation procedure. The proposed algorithm extracts the most salient line structures in the input image. On the other hand, the proposed algorithm suffers from

<sup>2</sup> <http://hugues.zahlt.info/91.html>

<sup>3</sup> <http://cvlab.epfl.ch/page-108936-en.html>





**Fig. 6.** We visualize the localization of the curvilinear structures on input images (a). We compare with the results of a manually labeled image by a human expert (b), morphological filtering [27] (c), supervised feature learning [3] (d), baseline MPP (e), and the proposed algorithm (f). Threshold values of (c) and (d) are chosen to achieve the closest recall scores to the proposed method. We use blue pixels to indicate areas which are completely corresponding to (b). Green and red pixels denote over-detected and under-detected areas, respectively, as compared with ground-truth. The name of the test images is from top to bottom: WRINKLE, DNA, CRACK, and RETINA.

under-detection when the width of the line structure is varying, for example, see the result for the RETINA. Such drawback can be overcome if we introduce an additional parameter for width of the line segment in our MPP model.

## 5 Conclusions

We have developed a new MPP model to reconstruct curvilinear structures via vectorized line segments. For the data likelihood, the density function computes rotated gradient magnitude and intensity variance. Prior energies of the proposed MPP model define interactions of the local configuration in terms of coupling energy states and overlapping areas. We have presented a new optimization scheme which is not biased by the parameter selection in the MPP model. We used an advanced RJMCMC sampler with different hyperparameter vectors to obtain line hypotheses. The line hypotheses span a feasible sample space, so that the final solution interprets underlying curvilinear structures more faithfully. We have shown line detection results on a wide range of datasets, and compared the performances of the proposed method with morphological filtering [27], supervised learning [3], and baseline MPP method. The whole optimization process is friendly designed to the parallel implementation; thus, the computational time can be further reduced by applying the parallel Monte Carlo sampler [31]. We plan to extend our model for time-varying sequences in order to analyze the temporal changes of the linear structures. While the heuristically proposed values for modeling parameters detect lines in practice, it is one of our future research topics to generate an optimal parameter vector using learning methods.

## References

1. Arbe ez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE TPAMI* 33(5), 898–916 (2011)
2. Batool, N., Chellappa, R.: Modeling and detection of wrinkles in aging human faces using marked point processes. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) *ECCV 2012 Ws/Demos, Part II*. LNCS, vol. 7584, pp. 178–188. Springer, Heidelberg (2012)
3. Becker, C., Rigamonti, R., Lepetit, V., Fua, P.: Supervised feature learning for curvilinear structure segmentation. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) *MICCAI 2013, Part I*. LNCS, vol. 8149, pp. 526–533. Springer, Heidelberg (2013)
4. Celeux, G., Chauveau, D., Diebolt, J.: Stochastic versions of the EM algorithm: an experimental study in the mixture case. *J. Statist. Comput. Simulation* 55(4), 287–314 (1996)
5. Chambon, S., Gourraud, C., Moliard, J.M., Nicolle, P.: Road crack extraction with adapted filtering and Markov model-based segmentation. In: *VISAPP(2)*, pp. 81–90 (May 2010)
6. Chatelain, F., Descombes, X., Lafarge, F., Lantuejoul, C., Mallet, C., Minlos, R., Schmitt, M., Sigelle, M., Stoica, R., Zhizhina, E.: *Stochastic geometry for image analysis*. Wiley-ISTE (2012)
7. Descombes, X., Zerubia, J.: Marked point process in image analysis. *IEEE SPM* 19(5), 77–84 (2002)
8. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) *MICCAI 1998*. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)

9. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE TPAMI* 13(9), 891–906 (1991)
10. Gamal-Eldin, A., Descombes, X., Charpiat, G., Zerubia, J.: Multiple birth and cut algorithm for point process optimization. In: *SITIS* (2010)
11. Gilks, W.R., Richardson, S., Spiegelhalter, D.: *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC (1995)
12. González, G., Türetken, E., Fleuret, F., Fua, P.: Delineating trees in noisy 2D images and 3D image-stacks. In: *CVPR*, pp. 2799–2806 (June 2010)
13. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4), 711–732 (1995)
14. Green, P.J., Mira, A.: Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika* 88(4), 1035–1053 (2001)
15. Jacob, M., Unser, M.: Design of steerable filters for feature detection using Canny-like criteria. *IEEE TPAMI* 26(8), 1007–1019 (2004)
16. Jeong, S.G., Tarabalka, Y., Zerubia, J.: Marked point process model for facial wrinkle detection. In: *ICIP*, pp. 1391–1394 (October 2014)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220(4598), 671–680 (1983)
18. Lacoste, C., Descombes, X., Zerubia, J.: Point processes for unsupervised line network extraction in remote sensing. *IEEE TPAMI* 27(10), 1568–1579 (2005)
19. Law, M.W.K., Chung, A.C.S.: Three dimensional curvilinear structure detection using optimally oriented flux. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 368–382. Springer, Heidelberg (2008)
20. Van Lieshout, M.N.M.: *Markov point processes and their application*. Imperial College Press (2000)
21. Møller, J., Waagepetersen, R.P.: *Statistical inference and simulation for spatial point processes*. Chapman & Hall/CRC (2003)
22. Predoehl, A., Barnard, K.: A statistical model for recreational trails in aerial images. In: *CVPR*, pp. 337–344 (June 2013)
23. Robert, C.P., Casella, G.: *Monte Carlo statistical methods*. Springer (2004)
24. Schlecht, J., Barnard, K., Spriggs, E., Pryor, B.: Inferring grammar-based structure models from 3D microscopy data. In: *CVPR*, pp. 1–8 (June 2007)
25. Staal, J.J., Abramoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. *IEEE TMI* 23(4), 501–509 (2004)
26. Stoyan, D., Kendall, W.S., Mecke, J.: *Stochastic geometry and its applications*. Wiley (1987)
27. Talbot, H., Appleton, B.: Efficient complete and incomplete path openings and closings. *Image and Vision Computing* 25(4), 416–425 (2007)
28. Tu, Z., Zhu, S.C.: Parsing images into regions, curves, and curve groups. *IJCV* 69(2), 223–249 (2006)
29. Türetken, E., Benmansour, F., Andres, B., Pfister, H., Fua, P.: Reconstructing loopy curvilinear structures using integer programming. In: *CVPR*, pp. 1822–1829 (June 2013)
30. Valero, S., Chanussot, J., Bendiktsson, J., Talbot, H., Waske, B.: Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Lett.* 31(10), 1120–1127 (2010)
31. Verdíé, Y., Lafarge, F.: Detecting parametric objects in large scenes by Monte Carlo sampling. *IJCV* 106(1), 57–75 (2014)

# Randomly Walking Can Get You Lost: Graph Segmentation with Unknown Edge Weights

Hanno Ackermann<sup>1</sup>, Björn Scheuermann<sup>1</sup>, Tat-Jun Chin<sup>2</sup>,  
and Bodo Rosenhahn<sup>1</sup>

<sup>1</sup> Institute for Information Processing, Leibniz University Hannover, Germany

<sup>2</sup> The University of Adelaide, Australia

**Abstract.** Spectral graph clustering is among the most popular algorithms for unsupervised segmentation. Applications include problems such as speech separation, segmenting motions or objects in video sequences and community detection in social media. It is based on the computation of a few eigenvectors of a matrix defining the connections between the graph nodes.

In many real world applications, not all edge weights can be defined. In video sequences, for instance, not all 3d-points of the observed objects are visible in all the images. Relations between graph nodes representing the 3d-points cannot be defined if these never co-occur in the same images. It is common practice to simply assign an affinity of zero to such edges.

In this article, we present a formal proof that this procedure decreases the separation between two clusters. An upper bound is derived on the second smallest eigenvalue of the Laplacian matrix. Furthermore, an algorithm to infer missing edges is proposed and results on synthetic and real image data are presented.

## 1 Introduction

Grouping similar data without any knowledge about the possible labeling is an important problem. This so-called unsupervised segmentation task is necessary in bioinformatics, machine learning, pattern recognition and computer vision.

One known technique for unsupervised segmentation is spectral clustering. It rests upon the segmentation of a graph capturing the relations between the data. Minimum cuts are used to decide the segmentation of the the graph into two [1, 2] or more sub-graphs [3].

Constructing the graph requires that affinities are computed between the vertices representing the data. If more than two data items are necessary to estimate the affinity, the corresponding *hyper*-edge connects all involved vertices [4]. For such hyper-graphs, the number of edges is exponential in the number of data items necessary to compute the affinity. Since many real-world problems induce prohibitively large hyper-edge sets, a commonly used approach is to estimate a subset of edges only [5–8], ie. many edge weights remain undefined.

In applications such as motion segmentation from video sequences [9] 2d-trajectories corresponding to different 3d-points have to be compared. Due to

occlusion or tracking failure, 2d-projections of different 3d-points may never occur in the same images so affinities between the corresponding graph nodes cannot be defined.

This situation is handled by state-of-the-art algorithms [6–9] by setting the corresponding edge weight to 0. This procedure is equivalent to assuming *maximum dissimilarity* between the two trajectories even though they may belong to the same group.

**The contribution** made here is twofold: (1) We model the impact of unknown edge weights in the context of spectral clustering. A *lower* bound on the separation between the two clusters and the *upper* bound on the eigenvalue gap is derived and proved. (2) We propose an algorithm to infer the weights of unknown edges using the known edges.

Whereas the effect of noise on the affinities has been investigated before [10, 11], this is the first work that considers the impact caused by undefined edges on spectral graph clustering.

The structure of this article is as follows: Some definitions are made in Section 2. Some facts of spectral clustering and the NCut-algorithm are shortly explained in the same section. Our first contribution, the derivation of upper and lower bounds on the cluster separability and the eigenvalue gap is presented in Sec. 3. Sections 4 and 5 present the proposed algorithm for inferring unknown edges of the graph and an application on motion segmentation. Experimental results using synthetic and real image data are shown in Sec. 6. The article concludes with a discussion in Sec. 7.

## 2 Definitions

For matrices, we use capital letters, such as matrix  $W$ . Vectors are indicated by lower-case letters, e.g.  $w$ . By  $w(i)$  we denote the  $i$ th entry of a vector  $w$  whereas  $w_i$  denotes the  $i$ th vector, for instance given  $W = [w_1 \cdots w_n]^\top$  the vector  $w_3$  implies the third row of matrix  $W$ . The  $(i, j)$ th entry of matrix  $W$  is indicated by  $w_{ij}$ . By  $\|v\|$ , we mean the  $L_2$ -norm of  $v$ .

Let  $G = (V, E)$  be an undirected graph consisting of  $|V| = n$  nodes  $V$  and a set of edges  $E$  connecting the nodes. Let there be subsets  $V_1 \subset V$  and  $V_2 \subset V$  such that  $V_1 \cap V_2 = \emptyset$  and  $|V_1| = n_1$ ,  $|V_2| = n_2$ ,  $n_1 + n_2 = n$ .

The real-valued weight  $w_{ij}(e_{ij})$  of an edge  $e_{ij}$  between two vertices  $v_i$  and  $v_j$  equals  $c_1$  if and only if both  $v_i$  and  $v_j$  are vertices of either  $V_1$  or  $V_2$ . Otherwise, its edge weight equals  $c_2 < c_1$ . We may further assume that  $c_1, c_2 \in [0, 1]$ . The idea motivating  $c_1$  and  $c_2$  is that the clusters do not need be perfectly separated. Noise on the edge weights can be considered by decreasing  $c_1$  and increasing  $c_2$ , respectively.

Denote by  $W$  the matrix consisting of the edges weights between all nodes. Assuming without loss of generality that the vertices are sorted, the  $n_1 \times n_1$  upper left block and lower right  $n_2 \times n_2$  block of  $W$  are all but  $c_1$  whereas the remaining entries of  $W$  equal  $c_2$ .

Let  $d_i = \sum_{j=1}^n w_{ij}$  and  $D$  the matrix with its  $(i, i)$ th diagonal entry equal to  $d_i$ . The Laplacian  $L$  of  $W$  can be defined as

$$L = I - D^{-1}W, \tag{1}$$

where  $I$  denotes the identity matrix. The segmentation is given by the eigenvector  $x_2$  to the second smallest eigenvalue of  $L$ . Assuming that the nodes are sorted and the clusters do not overlap, it is a piece-wise constant vector  $[1, 3, 10]$ . The labelling can be obtain by *kmeans*, for instance.

### 3 An Upper Bound on the Eigenvalue Gap

This section introduces the first contribution of this work, namely bounds on the second smallest eigenvalue of the Laplacian if not all edge weights are defined.

Let  $w$  be a vector consisting of the entries of a particular row of  $W$ . Denote by  $w'$  the same vector with some yet not all of its entries set to zero. It is possible to establish upper and lower bounds on the angle between  $w$  and  $w'$ :

**Lemma 1.** *For the angle between  $w$  and  $w'$  we have*

$$0^\circ < \angle(w, w') < 90^\circ \tag{2}$$

*Proof.* Let  $\mathcal{P}(w')$  denote the set of non-zero entries of  $w'$ . If a particular  $i$  is chosen such that  $w'(i) \neq 0$ , we have  $w(i) \cdot w'(i) = w(i)^2$ , hence we have for the scalar product between  $w$  and  $w'$

$$\frac{w^T \cdot w'}{\|w\| \cdot \|w'\|} > 0. \tag{3}$$

Regarding the upper bound on the scalar product, we have

$$w^\top \cdot w' = \sum_{i \in \mathcal{P}(w')} w'(i)^2 = \|w'\|^2. \tag{4}$$

we can see that Eq. (3) cannot attain a value of 1 since the first factor of the denominator is  $\|w\|$  and we have  $\|w\| > \|w'\|$ . □

These vectors  $w_i$  induced by the vertices  $v \in V_k$ , with either  $k = 1$  or  $k = 2$ , can be regarded to span an  $n_k$ -dimensional subspace

$$\mathcal{S}_k = \text{span} \left( w_{i_1}, \dots, w_{i_{n_k}} \right). \tag{5}$$

Denote by the matrix  $S_k$  an orthonormal basis of  $\mathcal{S}_k$ . For the angle between  $w'$  and the corresponding subspace  $\mathcal{S}_k$  we obtain that

**Lemma 2.**  $\angle(w', \mathcal{S}_k) > \angle(w, \mathcal{S}_k)$ .

*Proof.* Noticing that  $\angle(w, S_k) = 0 \Leftrightarrow \left\| \frac{w^\top}{\|w\|} \cdot S_k \right\| = 1$  we can show that  $\left\| \frac{w^\top}{\|w\|} \cdot S_k \right\| > \left\| \frac{w'^\top}{\|w'\|} \cdot S_k \right\|$  as in the proof of Lemma 1.  $\square$

Let  $w'_p$  be the vector with  $p$  of its entries being set to zero. From the above two lemmata, we immediately obtain that

**Corollary 1.** *The angle between  $w'_p$  and  $S_k$  increases with the weight of the “zeroed” entries of  $w'_p$ ,  $\sum_{i \notin \mathcal{P}(w'_p)} w(i)^2$*

$$\angle(w'_1, S_k) < \angle(w'_2, S_k) < \dots \tag{6}$$

Notice that the gap between the two clusters – measured by the difference between the second and the third smallest eigenvalue of the Laplacian  $L$  – is maximum if and only if  $S_1 \perp S_2$ . Increasingly perturbed vectors  $w'_i, i = 1, \dots, n_k$ , thus cause increasingly perturbed subspaces  $S'_k$  spanned by the vectors  $w'_i$ .

The question we are interested in is what happens to the angle between  $S'_1$  and  $S'_2$  compared with  $\angle(S_1, S_2)$ . If we are interested in exactly determining the gap, the convex hulls of the two sets of vectors  $w'_i$  need be compared.

However, analyzing the worst-case turns out to be much easier. This worst-case is defined by the *ex-radius*, ie. the largest distance between the centroid  $b_k$  of the points  $w_i \in S_k$  and the points  $w_i$ .

To derive an upper bound we need to determine the distance between  $w$  and  $w'_p$ . Obviously, the Euclidean distance between  $w$  and  $w'_p$  is largest if and only if  $c_1$  entries are zeroed. Here, we further assume that  $p < n_k$ , ie. not all edges to other vertices of the same cluster are removed.

**Theorem 1.** *The squared Euclidean distance between  $w$  and  $w'_p$  is bounded by*

$$\|w - w'_p\|^2 \leq p \cdot c_1^2. \tag{7}$$

With probability  $\rho_1 > \left(\frac{1}{1+\frac{n_2}{n_1}}\right)^p$  if  $w \in S_1$ , and  $\rho_2 > \left(\frac{1}{1+\frac{n_1}{n_2}}\right)^p$  if  $w \in S_2$ , respectively, the distance  $\|w - w'_p\|^2$  is smaller.

*Proof.* Since  $c_1 > c_2$ , the distance  $\|w - w'_p\|^2$  is largest if and only if all zeroed edges have weight  $c_1$ . The probability to sample  $c_1$  entries of vectors  $w \in S_1$  is given by

$$\rho'_1 = \frac{n_1!}{p!(n_1 - p)!} \cdot \frac{p!(n_1 + n_2 - p)!}{(n_1 + n_2)!} = \frac{n_1!(n_1 + n_2 - p)!}{(n_1 - p)!(n_1 + n_2)!} \tag{8}$$

$$= \frac{n_1!}{(n_1 - p)! \cdot \prod_{i=1}^p (n_1 + n_2 - p + i)} = \prod_{i=1}^p \frac{n_1 - p + i}{n_1 + n_2 - p + i}. \tag{9}$$

Defining  $d_i = n_1 - p + i$  we obtain

$$\rho'_1 = \prod_{i=1}^p \frac{d_i}{d_i + n_2} = \prod_{i=1}^p \frac{d_i}{d_i(1 + \frac{n_2}{d_i})} = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{d_i}} = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1 - p + i}}. \tag{10}$$

Using an upper bound of each factor we finally arrive at the claim:

$$\rho'_1 = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1 - p + i}} \tag{11}$$

$$< \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1 - p + p}} \tag{12}$$

$$= \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1}} = \left( \frac{1}{1 + \frac{n_2}{n_1}} \right)^p. \tag{13}$$

The derivation of the probability to only sample  $c_1$  entries of vectors  $w \in \mathcal{S}_2$  is equivalent.  $\square$

In other words, theorem 1 implies that the probability that the distance  $\|w - w'_p\|^2$  is smaller than  $p \cdot c_1^2$  reduces exponentially as the number of zeroed entries  $p$  in  $w'_p$  grows. In the following, we call this distance the *separation* between  $\mathcal{S}_1$  and  $\mathcal{S}_2$ .

Let the centroids  $b_1 = [c_1 \cdots c_1 \ c_2 \cdots c_2]$ ,  $b_2 = [c_2 \cdots c_2 \ c_1 \cdots c_1]$ , and a vector  $v$  parallel to  $b_1 - b_2$  with  $\|v\| = 1$ . Assuming that  $p$  entries of each vector  $w'_p \in \{\mathcal{S}_1, \mathcal{S}_2\}$  are zeroed where  $p < \{n_1, n_2\}$ , we arrive at

**Theorem 2.** *The separation between the two perturbed clusters equals  $s = s_1 + s_2$  where*

$$s_k \geq \left\| (b_k^\top v) v \right\|^2 - p \cdot c_1^2, \tag{14}$$

and the angle equals  $\alpha = \alpha_1 + \alpha_2$  where

$$\alpha_k \geq \tan^{-1} \frac{\sqrt{s_k}}{\|l\|} \tag{15}$$

with  $l = b_1 - (b_1^\top v) v = b_2 - (b_2^\top v) v$ .

*Proof.* The vector  $v$  indicates the line between the two cluster centroids  $b_k$ . If the orthogonal projection of the vector  $b_k$  onto  $v$  is subtracted from  $b_k$ , we obtain the vector  $l$  from the origin to the perpendicular point on the line between  $b_1$  and  $b_2$ .

The length of the line segment between one of the two centroids and the perpendicular point is given by the length of the orthogonal projection of the vector  $b_k$  onto  $v$ , ie.  $\|(b_k^\top v) v\|^2$ . Subtracting the radius of the sphere around  $b_k$  by lemma 1 yields the expression in Eq. (14).

Since the perpendicular line and the line between the perpendicular point and the sphere around each  $b_k$  form a right triangle, we can compute the angle between the perpendicular  $l$  and the vector between origin and the closest point on the sphere around  $b_k$  by Eq. (15).  $\square$

The probability that the separation  $s$  is in fact larger than the minimum stated in theorem 2 is  $1 - \rho_1 \rho_2$ , ie. usually very large.



Let  $v'_1$  and  $v'_2$  be the vectors from the origin to the intersection of the line between  $b_1$  and  $b_2$  with the sphere around each cluster center. Let the first two rows of the  $n \times n$  matrix  $T'$  be the vectors  $v'_1$  and  $v'_2$ , and the other rows being zero.

Let further  $v_1$  and  $v_2$  be the two 2d-vectors resulting from rotating  $[\sqrt{2} \ \sqrt{2}]$  by  $\frac{\alpha}{2}$  and  $-\frac{\alpha}{2}$ , respectively. Assume further that  $v_1$  and  $v_2$  are normalized such that  $\sum_i v_1(i) = \sum_i v_2(i) = 1$ . Let the  $2 \times 2$  matrix  $T$  consist of  $v_1$  and  $v_2$  as first and second row.

Using theorem 2 we are now able to derive a bound on the second largest eigenvalue of the Laplacian. The following theorem constitutes the first of our two main contributions.

**Theorem 3.** *The second smallest eigenvalue  $\lambda_2$  of the Laplacian matrix  $L$  is bounded by*

$$\lambda_2 \leq 1 - \lambda_{min} \tag{16}$$

where  $\lambda_{min}$  is the smaller of the two solutions  $0 \leq \lambda_{min} \leq 1$  of the quadratic equation

$$(t_{11} - \lambda_{min})(t_{22} - \lambda_{min}) - t_{12}t_{21} = 0 \tag{17}$$

where the scalars  $t_{ij}$  are the entries of the matrix  $T$ .

*Proof.* We can see that the two eigenvalues of  $T$  and  $T'$  are identical: Obviously both share the eigenvalue 1. The second eigenvalue is also identical since  $\alpha = \angle(v'_1, v'_2) = \angle(v_1, v_2)$ , and the principal angle  $\theta$  and the singular value and eigenvalue are related by

$$0 \leq \cos \theta = \sigma(v_1^\top v_2) = \lambda(|v_1^\top v_2|) \leq 1 \tag{18}$$

where  $\sigma(\cdot)$  denotes the singular value of the argument and  $\lambda(\cdot)$  the eigenvalue.

Lastly, as the eigenvalues of the Laplacian  $L$  are related to those of  $D^{-1}W$  by  $\Lambda(L) = 1 - \Lambda(D^{-1}W)$ , we obtain the claim in Eq. (16). □

## 4 Graph Completion

Let the edge  $\tilde{e}_{i,j}$  between vertices  $v_i$  and  $v_j$  be unknown. Denote by  $\mathcal{T}_u = (e_{i,\mathcal{L}_1}, e_{\mathcal{L}_1,\mathcal{L}_2}, \dots, e_{\mathcal{L}_n,j})$  the  $u$ th path between  $v_i$  and  $v_j$  of length  $n_l > 1$ .

**Proposition 1.** *The unknown weight  $\tilde{w}_{i,j}$  of an edge  $e_{i,j}$  between vertices  $v_i$  and  $v_j$  can be inferred by*

$$\tilde{w}_{i,j} = \max_u \{ \min \{ w(e), e \in \mathcal{T}_u \} \}. \tag{19}$$

This is motivated by the following idea: Let  $v_i$  and  $v_j$  both be vertices of the same cluster  $\mathcal{S}_k$ . Then, there exists at least a single path  $\mathcal{T}(i, j)$  such that all

edges along this path have large weight. Assume, conversely, that  $v_i \in \mathcal{S}_{k_1}$  and  $v_j \in \mathcal{S}_{k_2}$ ,  $k_1 \neq k_2$ . Then, all paths  $\mathcal{T}(i, j)$  contain at least a single edge with low weight.

As the probability that a path contains an edge with low weight increases with the path length, it suffices to search the  $q$  shortest paths  $\mathcal{T}_u$ ,  $u = 1, \dots, q$ , between  $v_i$  and  $v_j$ .

## 5 Application and Algorithm

In this section we present an algorithm for motion segmentation. Suppose that several sets of 3d-points move independently and are projected into images by a camera possibly also rotating and translating. Due to occlusion within the scene or failure of the feature point tracker, not every 2d-projection of a 3d-point is visible in all the images.

The problem is then to assign each trajectory – temporally consecutive 2d-projections of a particular 3d-point – a label indicating which group of 3d-points it belongs to. Associating one graph node with each trajectory, it requires to define affinities between each two vertices. If two trajectories do not overlap sufficiently such an affinity cannot be defined.

In the following we explain the procedure how to estimate affinities between nodes. It strongly rests upon the guided-sampling algorithm proposed in [12] yet only uses random sampling in a strict sense.

Low-dimensional subspaces are fitted to a subset of vertices representing the trajectories. Since subspace fitting is susceptible to missing data, a random vertex is selected first. We then discard vertices that are not visible at exactly those images the first vertex is visible at. From this subset,  $F - 1$  vertices are randomly chosen and the model parameters are computed by SVD. Here, we define  $F > D$ .

Given this model, the error is computed for *all* vertices which are visible in at least 8 of the images the subspace model is valid for. As error measure we use the Euclidean distance between each visible trajectory and the subspace. The resulting error is appended to a residual matrix. If a 3d-point is not visible in at least 8 of the images used for estimating the subspace, the corresponding entry in the residual matrix is set to undefined.

These steps are repeated  $R$  times. The error matrix is then sorted similarly as in [12]. The difference to [12] is that undefined entries are discarded at the sorting. If error vectors corresponding to two vertices  $v_i$  and  $v_j$  share the same  $b$  models among those  $H$  with lowest error, the weight of the edge between vertices  $v_i$  and  $v_j$  is set to  $e_{ij} = b/H$ . The scalar parameter  $H$  controls the connectivity of graph.

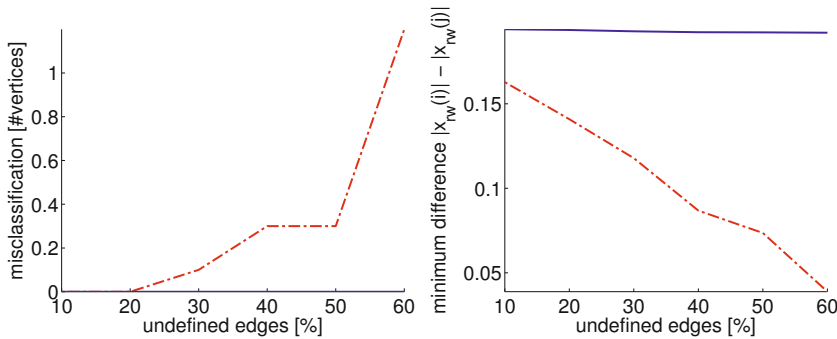
Finally, weights of edges which could not be defined are inferred by using the algorithm proposed in Sec. 4. The resulting complete graph is segmented using NCut spectral clustering.

## 6 Experiments

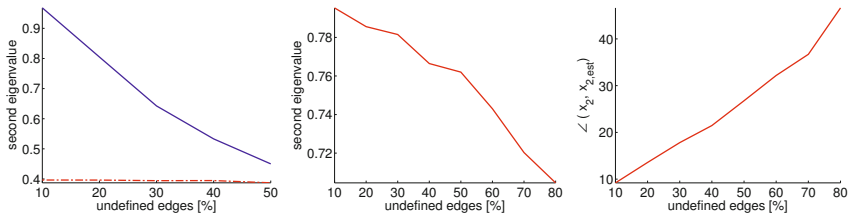
### 6.1 Synthetic Data

The algorithm proposed in Sec. 5 was evaluated using an artificial graph consisting of  $n_1 = 50$  and  $n_2 = 50$  nodes. While the data is ordered, neither the algorithm proposed in Sec. 5 nor the spectral clustering have any knowledge about the labeling.

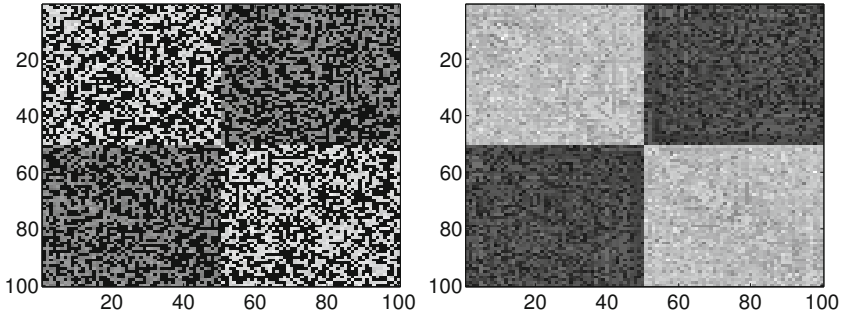
The effect which undefined but “zeroed” edge weights have on the spectral clustering can be seen by the dash-dotted red curve in the left plot of Fig. 1. Here, for fixed, normally distributed noise applied to the non-zero entries of  $W$ ,



**Fig. 1.** Dash-dotted red line: spectral clustering (undefined edge weights are set to zero). Solid blue line: proposed graph completion followed by NCut. Left plot: the number of incorrectly classified graph vertices. Right plot: minimum absolute sum between any two of the entries  $|x_2(i)|$  and  $|x_2(j)|$  of  $x_2$ .  $i$  and  $j$  indicate vertices of different clusters. Larger values of the minimum absolute sum indicate better separability of the two clusters.



**Fig. 2.** Left plot: The solid blue line indicates the predicted upper bound on the eigenvalue corresponding to the second smallest eigenvalue of the Laplacian while the dash-dotted red line indicates the true value. The noise was fixed to 2% and  $c_1 = 0.8$ ,  $c_2 = 0.2$ . Middle plot: The solid red line shows the measured second eigenvalue for  $c_1 = 0.6$  and  $c_2 = 0.4$ . Right plot: The solid line shows the angle between the corresponding eigenvector and the ground truth.



**Fig. 3.** The weight matrix  $W$  shown in the left image was created by taking  $c_1 = 0.6$ ,  $c_2 = 0.4$ , setting the noise level to a standard deviation 3% and randomly removing 60% of all edges. The right image shows the recovered graph by the proposed algorithm. The recovered eigenvector to the second smallest eigenvalue of the Laplacian  $L$  differed from the ground truth by approximately  $2.76^\circ$  whereas that vector corresponding to the right image differed by about  $36.87^\circ$ .

fixed separation  $c_1 - c_2$  and gradually increasing percentage of zeroed weights, the misclassification (measured by the number of misclassified vertices) increases (shown is the average of ten trials with different noise).

For this experiment, noise and separation were fixed to  $\sigma = 2\%$  and  $c_1 = 0.7$ ,  $c_2 = 0.3$ , respectively. The dash-dotted red lines indicate the results using a traditional spectral clustering, ie. the entries of the weight matrix  $W$  are simply set to zero. The solid blue lines indicate the results after the proposed graph completion followed by the spectral clustering.

The right plot in Fig. 1 shows the minimum

$$\min |x_2(i)| - |x_2(j)|, \quad \forall i, j \tag{20}$$

of the eigenvector  $x_2$  to the second smallest eigenvalue of  $L$  between any two of its entries  $|x_2(i)|$  and  $|x_2(j)|$  where  $i$  and  $j$  indicate vertices of different clusters. Larger values indicate better separability of the two clusters.

As can be seen, the amount of undefined edge weights decreases the separation between the two clusters for standard spectral clustering (dash-dotted red line). The proposed graph completion is not affected that strongly.

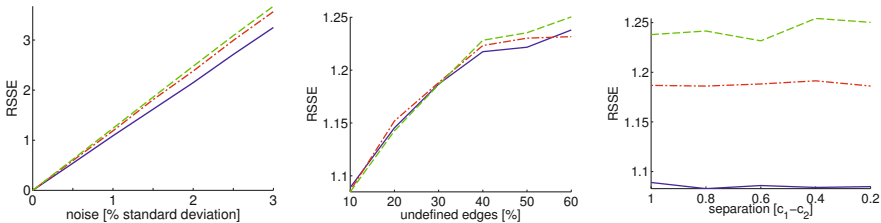
The left plot of Fig. 2 shows the average (of ten trials) of the predicted upper bound on the eigenvalue (solid blue line) corresponding to the second smallest eigenvalue of the Laplacian while the dash-dotted red line indicates the true value. The noise was fixed to 2% and  $c_1 = 0.8$ ,  $c_2 = 0.2$ . It can be seen that for increasing noise both the theoretical upper bound and the measured values decrease. The middle plot in the same figure shows the average of the measured second eigenvalue for  $c_1 = 0.6$  and  $c_2 = 0.4$ . Apparently, the eigenvalue decreases more strongly. The right plot shows the angle between the corresponding eigenvector and the ground truth. This perturbation causes the misclassification.

The left image of Fig. 3 shows an example of a weight matrix  $W$  if 60% of all edges are randomly removed. The right image in the same image shows the graph recovered by the proposed algorithm.

The three plots in Fig. 4 show averages of the Frobenius norm between the ground truth weight matrix and the recovered one of ten trials with different, normally distributed noise each time. In the left plot, the standard deviation was increased from 0 to 0.03 in steps of 0.005. The solid blue line indicates 10% randomly removed edges, the dash-dotted red line 30%, and the dashed green one 60%.

In the middle plot, the percentage of missing edges was increased. Here, the solid blue line was obtained by setting  $c_1 = 1$  and  $c_2 = 0$ , the dash-dotted red line  $c_1 = 0.8$  and  $c_2 = 0.2$ ; the dashed green line  $c_1 = 0.6$  and  $c_2 = 0.4$ . The noise was kept fixed to 1%. It can be seen that the amount of missing edges is more important than the difference between  $c_1$  and  $c_2$ .

The right figure corresponds to an experiment where the difference between  $c_1$  and  $c_2$  was decreased. The solid blue line indicates 10% randomly removed edges, the dash-dotted red line 30%, and the dashed green one 60%. The noise was kept fixed to 1%. Since all of the three lines are relatively constant while  $c_1 - c_2$  varies we can conclude that the determining factor is the amount of undefined edge weights.

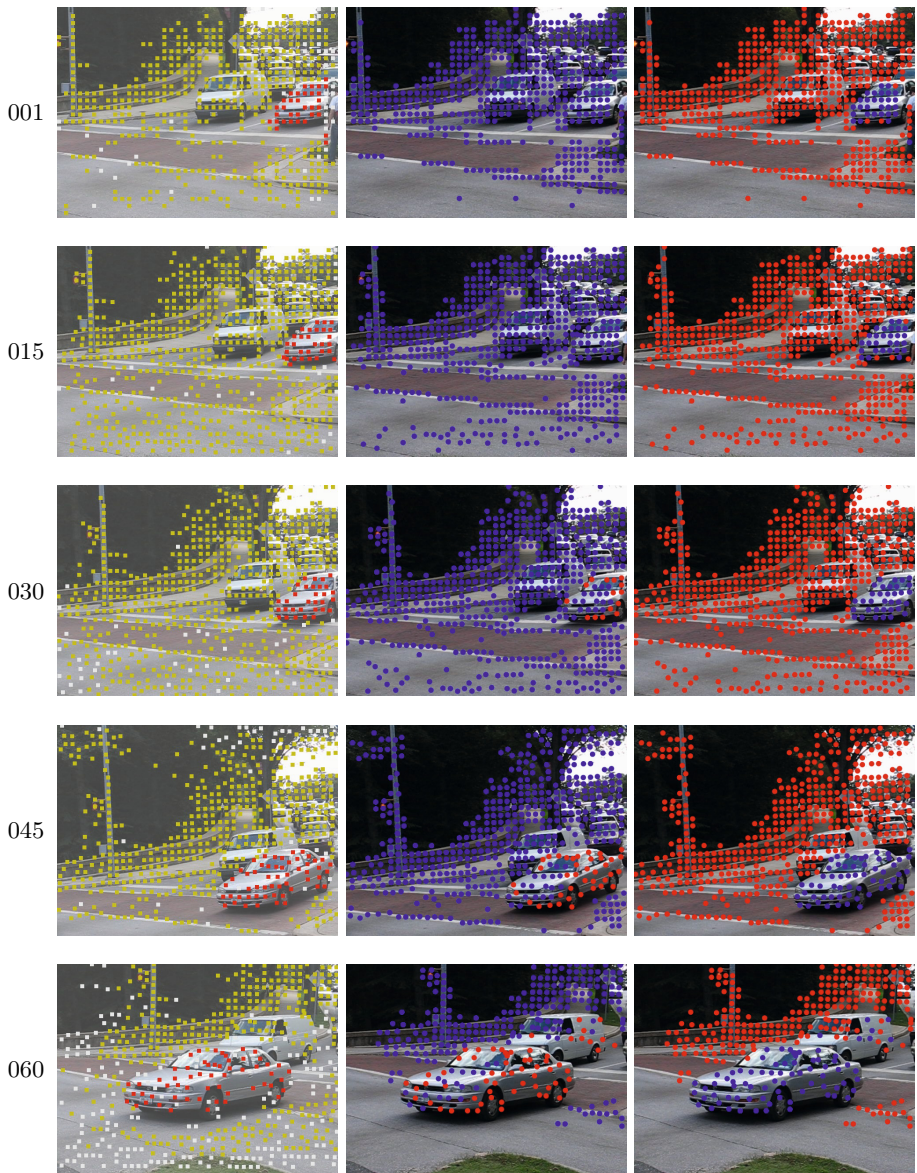


**Fig. 4.** Each of the three plots was obtained by varying a particular parameter: for the left plot, the noise was increased; edges were randomly removed for the middle one; the difference between  $c_1$  and  $c_2$  was decreased for the right plot. As error measure we used the Frobenius norm between the recovered weight matrix and the ground truth (root of sum of squared errors, RSSE). For the definitions of the different lines of each plot please see the explanation in Sec. 6.1.

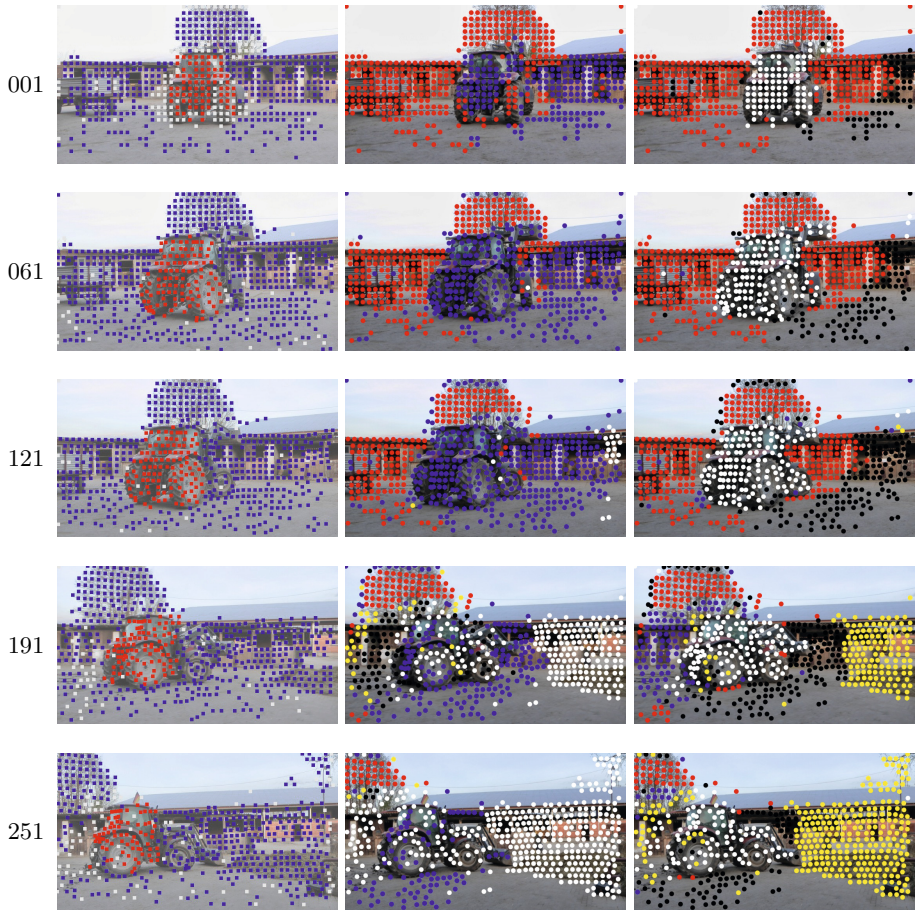
## 6.2 Real Image Experiments

Two real image sequences, *cars9* and *farm01*, were selected from the extended Berkeley motion database<sup>1</sup>. These sequences were selected because they are longer (60 and 252 images, respectively) so not all trajectories co-occur at a given number of frame. That also is the reason why the sequences from the popular Hopkins benchmark were not considered here as they simply do not contain sufficiently many missing correspondences.

<sup>1</sup> <http://lmb.informatik.uni-freiburg.de/resources/datasets>



**Fig. 5.** *cars9* sequence from the extended Berkeley motion database. Images in the left column indicate segmentation results by the algorithm of [6]; middle column: spectral clustering followed by spectral clustering *without* graph completion before; right column: proposed (graph completion followed by spectral clustering)



**Fig. 6.** *farm01* sequence from the extended Berkeley motion database. Images in the left column indicate segmentation results by the algorithm of [6]; middle column: spectral clustering followed by spectral clustering *without* graph completion before; right column: proposed (graph completion followed by spectral clustering)

The software from [6]<sup>2</sup> was used for tracking feature points. Tracks shorter than 8 images were discarded as the subspace distance measure is not reliable then.

For the *cars9* sequence 80% of the theoretically possible edges can be defined. The remaining ones correspond to trajectories that do not co-occur at sufficiently many images. For the *farm01* sequence, 40% of the trajectories do not overlap sufficiently.

Figures 5 and 6 show segmentation results on the *farm01* and the *cars9* sequence for a state-of-the-art algorithm for motion segmentation (left column) [6], the proposed algorithm for motion segmentation *without* graph completion before the spectral clustering (middle column), and the proposed algorithm *with* graph completion followed by spectral clustering (right column).

Differently than the algorithm of [6], the proposed algorithm does not merge multiple oversegmentations.

As can be seen, the proposed graph completion greatly improves segmentation results. The wheels of the tractor in the *farm01* sequence are somewhat mis-segmented as they define a separate rigid motion in terms of the rigid subspace measure as affinity.

## 7 Discussion

This article investigated the effect of undefined edge weights on spectral graph clustering. Upper bounds on the squared distance between two clusters were derived. It was possible to establish a lower bound on the second smallest eigenvalue of the Laplacian.

A practical algorithm was proposed to infer undefined edge weights. Its performance was evaluated using synthetic data. Using challenging sequences from a standard benchmark, it was shown that the proposed method outperforms both a state-of-the-art algorithm for motion segmentation and a spectral clustering without the graph completion.

Up to the best knowledge of the authors, this is the first work which considers the effect of undefined edges on the performance of spectral clustering.

## References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE TPAMI* 22, 888–905 (2000)
2. Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* 17, 395–416 (2007)
3. Maila, M., Shi, J.: A random walks view of spectral segmentation. In: *AISTATS 2001* (2001)
4. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: *Advances in Neural Information Processing Systems* (NIPS), pp. 1601–1608 (2006)

---

<sup>2</sup> The software is provided at <http://lmb.informatik.uni-freiburg.de/resources/binaries/>



5. Agarwal, S., Branson, K., Belongie, S.: Higher order learning with graphs. In: ICML 2006, pp. 17–24. ACM, New York (2006)
6. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR, pp. 614–621. IEEE Computer Society, Washington, DC (2012)
7. Purkait, P., Chin, T.-J., Ackermann, H., Suter, D.: Clustering with hypergraphs: The case for large hyperedges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 672–687. Springer, Heidelberg (2014)
8. van Gennip, Y., Hunter, B., Ahn, R., Elliott, P., Luh, K., Halvorson, M., Reid, S., Valasik, M., Wo, J., Tita, G.E., Bertozzi, A.L., Brantingham, P.J.: Community detection using spectral clustering on sparse geosocial data. *SIAM Journal of Applied Mathematics* 73, 67–83 (2013)
9. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 282–295. Springer, Heidelberg (2010)
10. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS, pp. 849–856 (2001)
11. Balakrishnan, S., Xu, M., Krishnamurthy, A., Singh, A.: Noise thresholds for spectral clustering. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 954–962 (2011)
12. Chin, T.J., Yu, J., Suter, D.: Accelerated hypothesis generation for multistructure data via preference analysis. *TPAMI* 34, 625–638 (2012)

# Training of Templates for Object Recognition in Invertible Orientation Scores: Application to Optic Nerve Head Detection in Retinal Images

Erik Bekkers<sup>1</sup>, Remco Duits<sup>1</sup>, and Marco Loog<sup>2</sup>

<sup>1</sup> Department of Biomedical Engineering and Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

<sup>2</sup> Pattern Recognition Laboratory, Delft University of Technology, The Netherlands

**Abstract.** A new template matching scheme for the detection of objects on the basis of orientations is proposed. The matching scheme is based on correlations in the domain  $\mathbb{R}^2 \times S^1$  of complex valued invertible orientation scores. In invertible orientation scores, a comprehensive overview of how an image is decomposed into local orientations is obtained. The presented approach allows for the efficient detection of orientation patterns in an intuitive and direct way. Furthermore, an energy minimization approach is proposed for the construction of suitable templates. The method is applied to optic nerve head detection in retinal images and extensive testing is done using images from both public and private databases. The method correctly identifies the optic nerve head in 99.7% of 1737 images.

**Keywords:** template matching, multi-orientation, invertible orientation scores, optic nerve head, optic disk, retina.

## 1 Introduction

We propose a new cross-correlation based template matching scheme for the detection of objects on the basis of local orientations. Template matching based on (normalized) cross correlation is a common approach to object recognition. The use of a similarity measure based on cross correlation is intuitive, easy to implement, and with the existence of optimization schemes for real-time processing [1, 2] a popular method to consider in computer vision tasks. However, the usual approach using pixel intensities as features for object recognition has its limitations, especially in applications where line-structures play an important role. In this case, template matching on the basis of geometrical information, e.g. local orientations, might be more appropriate (see e.g. [3]). We therefore generalize the concept of cross-correlations on position space  $\mathbb{R}^2$  to the joint space  $\mathbb{R}^2 \times S^1$  of positions and orientations ( $\equiv SE(2)$ , the Euclidean motion group). To this end, we represent an image  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  in the form of an *orientation score*  $U_f : \mathbb{R}^2 \times S^1 \rightarrow \mathbb{C}$ , i.e., a complex valued function on the extended domain  $\mathbb{R}^2 \times S^1$ . In an orientation score [4], we obtain a comprehensive overview

of how an image is decomposed into local orientations, see Fig. 1. We thus stay in the conventional and convenient framework of template matching via cross-correlation, however, the extension to orientation scores enables us to match patterns of orientation distributions, rather than pixel intensities.

For the construction of suitable templates we minimize an energy functional, where we pay attention to the following criteria:

1. The template should give a high response for inner products with positive object patches.
2. The template should ideally be perpendicular to negative object patches, i.e., the inner product with a negative object patch should be zero.
3. The template should be sufficiently smooth, as to prevent overfitting.

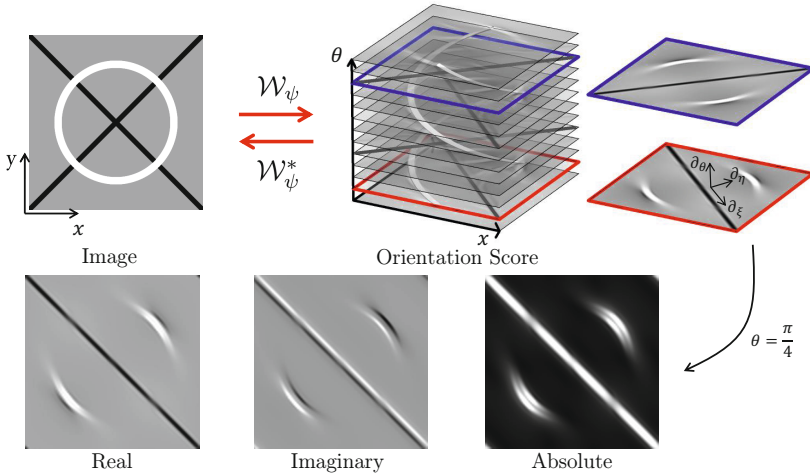
To enable 1 and 2 the energy functional contains a data-term. Here we make use of a representative training set of image patches, in which patches of the object of interest, as well as of objects not to be detected are included. To accommodate point 3, a Sobolev-type regularization-term is added to the energy functional. That is, regularization is done on the basis of gradients. In our extension to  $SE(2)$  we make use of left-invariant gradients, i.e., a derivative frame that rotates with the orientation of the group elements  $(\mathbf{x}, \theta) \in SE(2)$  in the orientation score. Consider to this end the  $(\partial_\xi, \partial_\eta, \partial_\theta)$ -frame in the upper right figure of Fig. 1. The left-invariant gradients allow for (anisotropic) regularization in the direction of oriented structures.

The proposed generic template matching framework on  $SE(2)$  is applied in the detection of the optic nerve head (ONH) in retinal images. Automated detection of the ONH is a challenging task and has therefore been the subject of many previous studies [5–11]. For a recent and extensive overview of ONH detection algorithms see [11]. Correct identification of the ONH is crucial in (automated) retinal image analyses, as the optic nerve head is either part of the analysis itself (classification of glaucoma [12]), or is used as a reference point in measurement protocols [13]. On conventional fundus (CF) images the ONH appears as a bright disk-like feature, but appears generally dark on images obtained by scanning laser ophthalmoscopy (SLO) cameras. Conventional ONH detection algorithms are designed for use with CF images, and are based on the analysis of pixel intensities [5, 6]. These approaches are fast, however, the performance typically decreases in the presence of pathologies. As an alternative, methods have been developed that include more contextual information and consider the typical pattern of blood vessels emerging from the optic nerve head [7–9]. These methods generally perform better than traditional methods; however, they often follow an elaborate processing pipeline, with high computational times as a consequence. Recently, methods have been proposed that are both fast and accurate, see [10, 11]. Our method is intuitive, easy to implement, fast and outperforms recent state-of-the-art methods on publicly available benchmark databases.

In this paper, we improve our recent work [14] on ONH detection by including: 1) training of templates; 2) regularization in  $SE(2)$ ; and 3) a thorough investigation on the combination of complementary templates. The generic template

matching framework is especially beneficial for the detection of objects characterized by orientations/line structures, as is demonstrated in our application to optic nerve head detection in retinal images.

**Structure of This Article.** The remainder of this article is organized as follows: In Section 2 the reader is provided with the necessary prerequisites. The section starts with an explanation of orientation scores (Subsection 2.1), followed by normalized cross-correlation and the concept of cross-correlation on orientation scores (Subsection 2.2). An optimization scheme for the construction of templates for cross-correlation based template matching is provided (Subsection 2.3). Section 3 describes our approach to optic nerve head detection in retinal images. In Section 4 the performance of the method is reported and discussed. General conclusions can be found in Section 5.



**Fig. 1.** Top row: Exemplary image and corresponding orientation score. Bottom row: Respectively the real part, imaginary part and modulus of a slice of the score at  $\theta = \frac{\pi}{4}$ .

## 2 Theory

### 2.1 Invertible Orientation Scores

An orientation score, constructed from image  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , is defined as a function  $U_f : \mathbb{R}^2 \times S^1 \rightarrow \mathbb{C}$  and depends on two variables  $(\mathbf{x}, \theta)$ , where  $\mathbf{x} = (x, y) \in \mathbb{R}^2$  denotes position and  $\theta \in [0, 2\pi]$  denotes the orientation variable. An orientation score  $U_f$  of image  $f$  can be constructed by means of correlation with some anisotropic wavelet  $\psi$  via

$$U_f(\mathbf{x}, \theta) = (\mathcal{W}_\psi f)(\mathbf{x}, \theta) = (\overline{\psi}_\theta \star f)(\mathbf{x}) = \int_{\mathbb{R}^2} \overline{\psi(\mathbf{R}_\theta^{-1}(\tilde{\mathbf{x}} - \mathbf{x}))} f(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}, \quad (1)$$

where  $\psi \in \mathbb{L}_2(\mathbb{R}^2)$  is the correlation kernel, aligned with the  $x$ -axis, where  $\mathcal{W}_\psi$  denotes the transformation between image  $f$  and orientation score  $U_f$ , and  $\star$  denotes correlation. The overline denotes complex conjugation,  $\psi_\theta(\mathbf{x}) = \psi(\mathbf{R}_\theta^{-1}\mathbf{x})$  and  $\mathbf{R}_\theta$  is a counter clockwise rotation over angle  $\theta$ . Note that  $\tilde{\mathbf{x}} \in \mathbb{R}^2$  denotes a location in the image domain, whereas  $(\mathbf{x}, \theta)$  denotes a location in the orientation score domain. The domain of an orientation score is essentially the classical Euclidean motion group  $SE(2)$  of planar translations and rotations, equipped with product  $g \cdot g' = (\mathbf{x}, \theta) \cdot (\mathbf{x}', \theta') = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \theta + \theta')$ .

In our work we choose cake wavelets [4] for  $\psi$ . Cake wavelets are designed to cover the entire Fourier domain, and have thereby the advantage over other oriented wavelets (s.a. Gabor wavelets) that they allow for a stable inverse transformation  $\mathcal{W}_\psi^*$  from the orientation score back to the image. As such, cake wavelets ensure that no data-evidence is lost during the transformation.

## 2.2 Template Matching via Normalized Cross Correlation

**Normalized Cross Correlation in  $\mathbb{R}^2$ .** Let us consider a template and an image,  $t, f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . We will denote translation by  $\mathbf{x}$  and rotation by  $\theta$  of template  $t$  using the representation  $(\mathcal{U}_g t)(\tilde{\mathbf{x}}) = t(\mathbf{R}_\theta^{-1}(\tilde{\mathbf{x}} - \mathbf{x}))$  and write  $g = (\mathbf{x}, \theta) \in SE(2)$ . The cross correlation coefficient as a function of translation and rotation of the template by  $g$  is then defined as follows:

$$c_{t,f}(g) = (\mathcal{U}_g t, f)_{\mathbb{L}_2(\mathbb{R}^2)} = \int_{\mathbb{R}^2} \overline{t(\mathbf{R}_\theta^{-1}(\tilde{\mathbf{x}} - \mathbf{x}))} f(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}} = (\bar{t}_\theta \star f)(\mathbf{x}), \quad (2)$$

where  $(\cdot, \cdot)_{\mathbb{L}_2(\mathbb{R}^2)}$  denotes the  $\mathbb{L}_2$  inner product.

In order to make the correlation measure invariant to intensity scalings, both slots in the inner product can be normalized to zero mean and unit standard deviation. This is known as *normalized cross correlation*. To be able to normalize the image locally we make use of an additional mass function  $m : \mathbb{R}^2 \rightarrow \mathbb{R}^+$  with  $\int m(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}} = 1$ , which indicates the relevant region of the template, and define the  $\mathbb{L}_2(\mathbb{R}^2)$  inner product using probability measure  $m(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}$  as follows:

$$(t, f)_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})} = \int_{\mathbb{R}^2} \overline{t(\tilde{\mathbf{x}})} f(\tilde{\mathbf{x}}) m(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}. \quad (3)$$

The normalized cross correlation coefficient  $\hat{c}_{t,f}(g)$  is then defined as follows:

$$\hat{c}_{t,f}(g) = (\mathcal{U}_g \hat{t}, \hat{f}_g)_{\mathbb{L}_2(\mathbb{R}^2, \mathcal{U}_g m d\tilde{\mathbf{x}})}, \quad (4a)$$

$$\hat{t}(\tilde{\mathbf{x}}) = \frac{t(\tilde{\mathbf{x}}) - \langle t \rangle_m}{\|t - \langle t \rangle_m\|_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})}}, \quad (4b)$$

$$\hat{f}_g(\tilde{\mathbf{x}}) = \frac{f(\tilde{\mathbf{x}}) - \langle f \rangle_{\mathcal{U}_g m}}{\|f - \langle f \rangle_{\mathcal{U}_g m}\|_{\mathbb{L}_2(\mathbb{R}^2, \mathcal{U}_g m d\tilde{\mathbf{x}})}}, \quad (4c)$$

with  $\langle t \rangle_m = (1, t)_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})}$  the local average with respect to the area covered by  $m$ , and with  $\|\cdot\|_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})} = \sqrt{(\cdot, \cdot)_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})}}$ .

Since the normalized image  $\hat{f}_g$  depends on  $g$  it needs to be calculated for every translation of the template, making this approach computationally expensive. Therefore, we will instead approximate (4c) by assuming that the local average is approximately constant in the area covered by  $m$  and that the mass is rotation invariant (i.e.,  $m(\mathbf{R}_\theta^{-1}\mathbf{x}) = m(\mathbf{x})$ ). That is, assuming  $\langle f \rangle_{\mathcal{U}_{(\mathbf{x},\theta)}m}(\tilde{\mathbf{x}}) \approx \langle f \rangle_{\mathcal{U}_{(\tilde{\mathbf{x}},\theta)}m}(\tilde{\mathbf{x}}) = (m \star f)(\tilde{\mathbf{x}})$  for  $\|\tilde{\mathbf{x}} - \mathbf{x}\|_{\mathbb{L}_2(\mathbb{R}^2)} < r$ , with  $r$  the radius that determines the extent of  $m$ , we approximate (4c) as follows:

$$\hat{f}_g(\tilde{\mathbf{x}}) \approx \frac{f(\tilde{\mathbf{x}}) - (m \star f)(\tilde{\mathbf{x}})}{\sqrt{(m \star (f - (m \star f)))(\tilde{\mathbf{x}})}} \tag{5}$$

**Normalized Cross Correlation in  $SE(2)$ .** Analogue to the  $\mathbb{R}^2$  case, for two normalized orientation scores  $\hat{T}, \hat{U}_f \in \mathbb{L}_2(SE(2))$  the normalized correlation is given by

$$\hat{C}_{T,U_f}(g) = \left( \mathcal{L}_g \hat{T}, \hat{U}_f \right)_{\mathbb{L}_2(SE(2)), \mathcal{L}_g M d\mathbf{x}d\theta} \tag{6}$$

There we take the  $SE(2)$  inner product with probability measure  $M(\mathbf{x}, \theta)d\mathbf{x}d\theta$ :

$$\left( \hat{T}, \hat{U}_f \right)_{\mathbb{L}_2(SE(2), M d\mathbf{x}d\theta)} = \int_{\mathbb{R}^2} \int_0^{2\pi} \overline{\hat{T}(\mathbf{x}, \theta)} \hat{U}_f(\mathbf{x}, \theta) M(\mathbf{x}, \theta) d\theta d\mathbf{x}, \tag{7}$$

and the shift-twist operator  $(\mathcal{L}_g T)(\mathbf{x}, \theta) = T(\mathbf{R}_\alpha^{-1}(\mathbf{x} - \mathbf{b}), \theta - \alpha)$ . Rotations by  $\alpha$  followed by a translation  $\mathbf{b}$  via  $\mathcal{L}_g$ , with  $g = (\mathbf{b}, \alpha)$ , of orientation scores is done since  $(\mathcal{W}_\psi \mathcal{U}_g f)(\mathbf{x}, \theta) = (\mathcal{L}_g \mathcal{W}_\psi f)(\mathbf{x}, \theta)$ . Normalized template  $\hat{T}$  and orientation score  $\hat{U}_f$  are calculated in a similar fashion as described in Eq. (4b) and (5), where one can replace all inner products  $(\cdot, \cdot)_{\mathbb{L}_2(\mathbb{R}^2, m d\tilde{\mathbf{x}})}$  by  $(\cdot, \cdot)_{\mathbb{L}_2(SE(2), M d\mathbf{x}d\theta)}$  and where the correlation operator  $\star$  can be replaced by its  $SE(2)$  equivalent:

$$(T \star_{SE(2)} U_f)(\mathbf{x}, \theta) = (\mathcal{L}_{(\mathbf{x},\theta)} T, U_f)_{\mathbb{L}_2(SE(2), M d\mathbf{x}d\theta)} \tag{8}$$

**Matching of Patterns of Orientation Distributions Using  $|U_f|$ .** Since both the orientation score transform (1) and template matching schemes, (4a) and (6), rely on a series of linear operators (correlations), it is possible to show that both Eq. (4a) and (6) produce the same results if the orientation score objects originate from their image equivalents. That is, there is no gain in performing template matching in  $SE(2)$  if  $U_f = \mathcal{W}_\psi f$  and  $T = \mathcal{W}_\psi t$ , since then  $\operatorname{argmax}_{g \in SE(2)} \hat{c}_{t,f}(g) = \operatorname{argmax}_{g \in SE(2)} \hat{C}_{T,U_f}(g)$ . However, in this work we find the ONH location  $g_o = (\mathbf{x}_o, \theta_o) \in SE(2)$  via

$$g_o = \operatorname{argmax}_{g \in SE(2)} \left( \mathcal{L}_g \hat{T}, |\widehat{U}_f| \right)_{\mathbb{L}_2(SE(2), M d\mathbf{x}d\theta)} \tag{9}$$

Here template matching in  $SE(2)$  is done via the modulus of the orientation scores. This adaptation makes that the *appearance* (encoded in the phase) of

structures is not measured, it is rather the *presence* of structures that is being detected, consider to this end the bottom row of Fig. 1. We remove the image DC component before applying the orientation score transform. This guarantees a low response at locally constant regions where no orientation preference is expected. The absolute orientation score  $|U_f(\mathbf{x}, \theta)|$  can then be regarded as a measure for finding an oriented structure at position  $\mathbf{x}$  and orientation  $\theta$ . Note also that similar techniques for linear structure detection have been used before by Freeman et al. using (steerable) quadrature filter pairs [15].

### 2.3 Template Training

We describe a framework for the construction of suitable templates via the minimization of an energy functional. First, the energy functionals for both the  $\mathbb{R}^2$  and the  $SE(2)$  are described. Then, the templates will be represented in a B-spline basis. This allows for efficient and accurate optimization of the energy functionals. Finally, the minimizers corresponding to the energy functionals are presented in matrix-vector notation. A simple conjugate gradient approach can be used to solve for the B-spline coefficients.

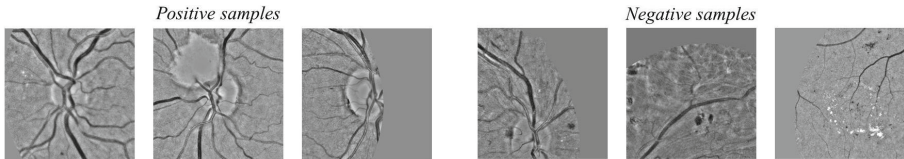


Fig. 2. Exemplary retinal image patches used for template training

**Energy Functional.** For the optimization of 2D image templates  $t$ , the following energy functional is minimized:

$$E(t) = \underbrace{\sum_{i=1}^P \left( \left( t, \hat{f}_i \right)_{\mathbb{L}_2(\mathbb{R}^2, m d\bar{x})} - y_i \right)^2}_{\text{data term}} + \lambda \underbrace{\int_{\mathbb{R}^2} \|\nabla t\|_{\mathbb{L}_2(\mathbb{R}^2)}^2 dx dy}_{\text{regularization term}}, \quad (10)$$

where  $\hat{f}_i$  is one of  $P$  normalized patches representing either the optic nerve head, in which case the corresponding label  $y_i = 1$ , or a negative samples, in which case  $y_i = 0$ . Patches with label  $y_i = 1$  will be referred to as positive patches, those with label  $y_i = 0$  as negative patches. Examples of training patches are given in Fig. 2. The data-term of the functional trains the template to give a response of 1 if the inner-product is taken with a positive patch, and to give response 0 otherwise. The regularization term ensures that the template is smooth enough. I.e., sharp transitions in image intensities are punished using the squared gradient magnitude  $\|\nabla t\|^2$ . Parameter  $\lambda$  balances the data- and regularization-term.

Similar to Eq. (10), for the optimization of the orientation score template  $T$  the following functional is minimized:

$$\mathcal{E}(T) = \underbrace{\sum_{i=1}^P ((T, \hat{U}_{f_i})_{\mathbb{L}_2(SE(2), Mdx d\theta)} - y_i)^2}_{\text{data term}} + \lambda \underbrace{\iint\limits_{SE(2)} \|\nabla T\|_D^2 dx dy d\theta}_{\text{regularization term}}, \quad \text{with}$$

$$\iint\limits_{SE(2)} \|\nabla T\|_D dx dy d\theta = \iint\limits_{SE(2)} D_{\xi\xi} \left| \frac{\partial T}{\partial \xi} \right|^2 + D_{\eta\eta} \left| \frac{\partial T}{\partial \eta} \right|^2 + D_{\theta\theta} \left| \frac{\partial T}{\partial \theta} \right|^2 dx dy d\theta, \quad (11)$$

and with the left-invariant gradient  $\nabla T = \left( \frac{\partial T}{\partial \xi}, \frac{\partial T}{\partial \eta}, \frac{\partial T}{\partial \theta} \right)^T$  defined by

$$\partial_\xi := \cos \theta \partial_x + \sin \theta \partial_y, \quad \partial_\eta := -\sin \theta \partial_x + \cos \theta \partial_y, \quad \text{and } \partial_\theta. \quad (12)$$

Note that  $\partial_\xi$  gives the spatial derivative in the direction aligned with the orientation score kernel used at layer  $\theta$ , recall Fig. 1. The parameters  $D_{\xi\xi}$ ,  $D_{\eta\eta}$  and  $D_{\theta\theta}$  are used to balance the regularization in the three directions. Similar to this problem, first order Tikhonov-regularization on  $SE(2)$  is related<sup>1</sup>, via temporal Laplace transforms, to left-invariant diffusions on the group  $SE(2)$ . In which case  $D_{\xi\xi}$ ,  $D_{\eta\eta}$  and  $D_{\theta\theta}$  denote the diffusion constants in  $\xi$ ,  $\eta$  and  $\theta$  direction. Here we set  $D_{\xi\xi} = 1$ ,  $D_{\eta\eta} = 0$ , and thereby we get Laplace transforms of hypo-elliptic diffusion processes [16, 17]. Parameter  $D_{\theta\theta}$  can be used to tune between isotropic (large  $D_{\theta\theta}$ ) and anisotropic (low  $D_{\theta\theta}$ ) diffusion. See Fig. 3, where we have illustrated the Green’s function of hypo-elliptic diffusion processes and the effect of regularization parameter  $D_{\theta\theta}$  in the score domain. Note that anisotropic diffusion, via a low  $D_{\theta\theta}$ , is preferred as we want to maintain line structures in orientation scores.

**B-Spline Basis.** In order to efficiently minimize (10) and (11), the templates are described in a B-spline basis of direct products of  $n$ -th order B-splines  $B^n$ :

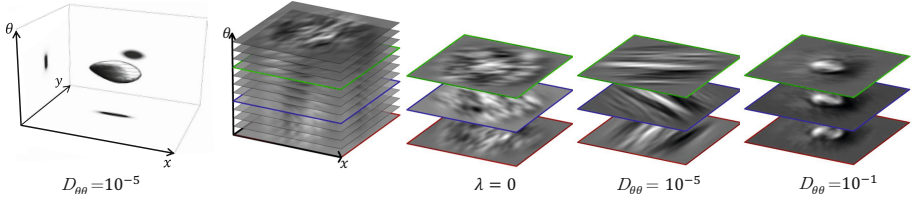
$$t(x, y) = \sum_{k=1}^{N_k} \sum_{l=1}^{N_l} c_{k,l} B^n \left( \frac{x}{s_k} - k \right) B^n \left( \frac{y}{s_l} - l \right), \quad (13a)$$

$$T(x, y, \theta) = \sum_{k=1}^{N_k} \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} c_{k,l,m} B^n \left( \frac{x}{s_k} - k \right) B^n \left( \frac{y}{s_l} - l \right) B^n \left( \frac{\theta \bmod 2\pi}{s_m} - m \right), \quad (13b)$$

with  $B^n(x) = \left( 1_{[-\frac{1}{2}, \frac{1}{2}]} \ast^{(n)} 1_{[-\frac{1}{2}, \frac{1}{2}]} \right) (x)$  a  $n$ -th order B-splines obtained by  $n$ -fold convolution of the indicator function  $1_{[-\frac{1}{2}, \frac{1}{2}]}$ , and  $c_{k,l}$  and  $c_{k,l,m}$  the coefficients belonging to the shifted B-splines for  $\mathbb{R}^2$  respectively  $SE(2)$ .

<sup>1</sup> In which case  $\|T - U_f\|_{\mathbb{L}_2(SE(2))}^2$  is used for the data term instead of the one in (11).





**Fig. 3.** The Green’s function of hypo-elliptic diffusion with  $D_{\theta\theta} = 10^{-5}$  and an orientation score template with different regularization settings. From left to right: no regularization ( $\lambda = 0$ ), anisotropic ( $D_{\theta\theta} = 10^{-5}$ ), and isotropic regularization ( $D_{\theta\theta} = 10^{-1}$ ).

**The Minimizer for the  $\mathbb{R}^2$  Case in Matrix-Vector Notation.** By substitution of (13a) in (10), the energy functional can be expressed in matrix-vector notation as follows:

$$E(t) = E_D(\mathbf{c}) := \|\mathbf{S}\mathbf{c} - \mathbf{y}\|^2 + \mathbf{c}^\dagger \mathbf{R}\mathbf{c}. \tag{14}$$

The corresponding minimizer is given by

$$(\mathbf{S}^\dagger \mathbf{S} - \lambda \mathbf{R})\mathbf{c} = \mathbf{S}^\dagger \mathbf{y}. \tag{15}$$

Here  $\mathbf{S}$  is a  $[P \times N_k N_l]$  matrix given by

$$S = \{ (s_{1,1}^i, s_{1,2}^i, \dots, s_{1,N_l}^i, s_{2,1}^i, s_{2,2}^i, \dots, s_{2,N_l}^i, \dots, \dots, s_{N_k,2}^i, \dots, s_{N_k,N_l}^i) \}_{i=1}^P, \tag{16}$$

$$s_{k,l} = (B_{s_k s_l}^n * (m \hat{f}_i))(k, l),$$

with  $B_{s_k s_l}^n(x, y) = B^n\left(\frac{x}{s_k}\right) B^n\left(\frac{y}{s_l}\right)$ , for all  $(x,y)$  on the discrete spatial grid on which the discrete input image  $f_D : \{1, N_x\} \times \{1, N_y\} \rightarrow \mathbb{R}$  is defined. Here  $N_k$  and  $N_l$  denote the number of splines in resp.  $x$  and  $y$  direction, and  $s_k = \frac{N_x}{N_k}$  and  $s_l = \frac{N_y}{N_l}$  are the corresponding resolution parameters. The  $[N_k N_l \times 1]$  column vector  $\mathbf{c}$  contains the B-spline coefficients, and the  $[P \times 1]$  column vector  $\mathbf{y}$  contains the labels, stored in the following form

$$\begin{aligned} \mathbf{c} &= (c_{1,1}, c_{1,2}, \dots, c_{1,N}, c_{2,1}, c_{2,2}, \dots, c_{2,N}, \dots, \dots, c_{M,2}, \dots, c_{M,N})^T \\ \mathbf{y} &= (y_1, y_2, \dots, y_P)^T. \end{aligned} \tag{17}$$

The  $[N_k N_l \times N_k N_l]$  regularization matrix  $\mathbf{R}$  is given by

$$\mathbf{R} = R_x^{s_k} \otimes R_x^{s_l} + R_y^{s_k} \otimes R_y^{s_l}, \tag{18}$$

where  $\otimes$  denotes the Kronecker product, and with

$$\begin{aligned} R_x^{s_k}(k, k') &= -\frac{1}{s_k} \frac{\partial^2 B^{2n+1}}{\partial x^2}, (k' - k) & R_y^{s_k}(k, k') &= s_k B^{2n+1}(k' - k), \\ R_x^{s_l}(l, l') &= s_l B^{2n+1}(l' - l), & R_y^{s_l}(l, l') &= -\frac{1}{s_l} \frac{\partial^2 B^{2n+1}}{\partial y^2}(l' - l), \end{aligned} \tag{19}$$

with  $k, k' = 1, 2, \dots, N_k$  and  $l, l' = 1, 2, \dots, N_l$ .

**The Minimizer for the  $SE(2)$  Case in Matrix-Vector Notation.** For the  $SE(2)$  case, the shape of the energy functional  $\mathcal{E}(T)$  and the corresponding minimizer are the same as for  $E(t)$  on the  $\mathbb{R}^2$  case, and are given by (14) and (15). However, the definitions of  $S$ ,  $R$  and  $\mathbf{c}$  are different. In this case  $S$  is a  $[P \times N_k N_l N_m]$  matrix given by

$$S = \left\{ (s_{1,1,1}^i, s_{1,1,2}^i, \dots, s_{1,1,N_m}^i, s_{1,2,1}^i, \dots, s_{1,2,N_m}^i, \dots, s_{1,N_l,N_m}^i, \dots, s_{N_k,N_l,N_m}^i) \right\}_{i=1}^P, \\ s_{k,l,m} = (B_{s_k s_l s_m}^n * (M \hat{U}_{f_i}))(k, l, m), \quad (20)$$

with  $B_{s_k s_l s_m}^n(x, y, \theta) = B^n\left(\frac{x}{s_k}\right) B^n\left(\frac{y}{s_l}\right) B^n\left(\frac{\theta \bmod 2\pi}{s_m}\right)$ , with angular resolution parameter  $s_m = 2\pi/N_m$ . Vector  $\mathbf{c}$  is a  $[N_k N_l N_m \times 1]$  column vector containing the B-spline coefficients and is stored as follows:

$$\mathbf{c} = (c_{1,1,1}, c_{1,1,2}, \dots, c_{1,1,N_m}, c_{1,2,1}, \dots, c_{1,2,N_m}, \dots, c_{1,N_l,N_m}, \dots, c_{N_k,N_l,N_m})^T. \quad (21)$$

The explicit expression of  $[N_k N_l N_m \times N_k N_l N_m]$  matrix  $R$  is given in Appendix A.

### 3 Optic Nerve Head Detection

#### 3.1 Processing Pipeline

The location of the ONH is found through the following five steps:

1. The input image  $f$  is (locally) normalized via the Luminosity-Contrast normalization method described by Foracchia et al. [18], giving  $f_{lcn}$ .
2. To further reduce sensitivity to high intensity structures, we apply the following intensity mapping  $f_{lcn}^{erf} = \text{erf}(8 f_{lcn})$ , with  $\text{erf}(i) = \frac{2}{\sqrt{\pi}} \int_0^i e^{-x^2} dx$  the error function. The effect is a soft binarization of the image, by which more emphasis is put on contextual information rather than intensity information.
3. In case of a 2D template  $t$ ,  $\hat{f}_{lcn}^{erf}$  is approximated by (5). In case of an orientation score template  $T$ , the score  $U_{f_{lcn}^{erf}}$  is calculated via (1), and is normalized after taking the modulus giving  $|\hat{U}|_{f_{lcn}^{erf}}$ .
4. An ONH probability map  $P^t(g) = \hat{c}_{t,f}(g)$  is calculated via (4a), or  $P^T(g) = \hat{C}_{T,U_f}(g)$  is calculated via (6) in case of an orientation score template.
5. An ONH probability map constructed using template  $\tau$  (with  $\tau = t$  or  $\tau = T$ ) is denoted by  $P^\tau$ . In case multiple templates are used, each probability map  $P^\tau$  is rescaled to a range of  $[0, 1]$ . The final optic nerve head location is then calculated as  $g_o = \underset{g \in SE(2)}{\text{argmax}} \left( \sum_{\tau \in \mathcal{T}} P^\tau(g) \right)$ , with  $\mathcal{T}$  the set of templates used.

Since the ONH generally appears under the same orientation in every image, we restrict our search for the ONH location  $g_o = (\mathbf{x}_o, \theta_o)$  to translations  $\mathbf{x}_o$  only, and assume  $\theta_o = 0$ . To reduce computation time the image is rescaled by a factor of  $\frac{r_{target}}{r_{est}}$ , with  $r_{target} = 20$  pixels and  $r_{est}$  the estimated optic disk radius. For normalization in step 1 we used a window size of  $\frac{1}{2}r_{target}$ , for the orientation score transforms we used cake wavelets [4] with angular resolution  $s_\theta = \frac{\pi}{12}$ . For normalization we have used isotropic mass functions  $M(\mathbf{x}, \theta) = m(\mathbf{x})$ , for details see [14, Section 2.4].

### 3.2 Templates

In our experiments we have considered three different types of templates: model templates, average templates and trained templates. In total we will be investigating 6 different templates, labeled **A-F**, see Table. 1. Template **A** is a disk filter and models the shape of the optic disk, and has been used for ONH detection in [8]. Template **B** is a template that models the pattern of blood vessels radiating outwards from the ONH. This template is described in our previous work [14]. Templates **C** and **D** are average templates, and are respectively found by  $t_C = \frac{1}{P} \sum_{i=1}^P f_i$  and  $T_D = \frac{1}{P} \sum_{i=1}^P |U_{f_i}|$ , with  $\{f_1, \dots, f_P\}$  the set of positive ONH image patches, and  $\{U_{f_1}, \dots, U_{f_P}\}$  the set of orientation scores hereof.

Templates **E-F** are trained using the the methods described in Subsection 2.3. The number of B-splines was set to  $N_k = N_l = 50$  and  $N_m = 12$ . Template **E** is constructed in the  $\mathbb{R}^2$  domain with regularization parameter  $\lambda = 10^{-1.5}$ . The orientation score template **F** is constructed using regularization parameters  $\lambda = 10$  and  $D_{\theta\theta} = 10^{-3.5}$ .

Templates **C-F** require a training set. The set is constructed using the first  $P = 100$  images of the publicly available MESSIDOR database (<http://messidor.crihan.fr/index-en.php>). Each positive optic nerve head patch  $f_i$  (with label  $y_i = 1$ ) is centered at the ONH and has a square window size of  $8 r_{target}$ . The negative patches were selected based on the critical areas for template **C**. Each negative patch  $f_i$  (with label  $y_i = 0$ ) is centered around the largest local maximum of the image filtered with template **C**, and which does not lie within the circumference of the optic disk. See Fig. 2 for examples of positive and negative patches. The images used in the training underwent the same first three processing steps as described in Subsection 3.1. For processing of conventional (RGB) fundus images we used the green channel. For SLO images we used the near-infrared color channel of the first  $P = 100$  images of our private SLO image database, which will be described in the next section.

## 4 Results and Discussion

**Data.** For validation, we made use of a private database consisting of 208 SLO images taken with an EasyScan (i-Optics B.V., the Netherlands) and 208 CF images taken with a Topcon NW200 (Topcon Corp., Japan). For full details see [14]. The two sets of images are labeled as "ES" and "TC" respectively. Our method is also tested on three widely used public databases: MESSIDOR, DRIVE (<http://www.isi.uu.nl/Research/Databases/DRIVE>) and STARE (<http://www.ces.clemson.edu/~ahoover/stare>), consisting of 1200, 40 and 81 images respectively. For each image, the detected ONH position was marked as correct if it was located within the circumference of the actual ONH. To this end we used the annotations kindly provided by the authors of [6] (<http://www.uhu.es/retinopathy>), and manually outlined the ONH border for the other databases.

**Table 1.** Results of (combinations) of templates for optic nerve head detection (number of fails in parentheses)

Template ID	Domain	ES (SLO) 208	TC 208	MESSIDOR 1200	DRIVE 40	STARE 81	All Images 1737
<b>model templates</b>							
A	$\mathbb{R}^2$	65.38% (72)	95.19% (10)	82.00% (216)	87.50% (5)	53.09% (38)	80.37% (341)
B	$SE(2)$	62.50% (78)	77.40% (47)	86.67% (160)	67.50% (13)	65.43% (28)	81.23% (326)
<b>average templates</b>							
C	$\mathbb{R}^2$	99.52% (1)	99.04% (2)	98.00% (24)	95.00% (2)	67.90% (26)	96.83% (55)
D	$SE(2)$	99.52% (1)	100.0% (0)	99.50% (6)	97.50% (1)	93.83% (5)	99.25% (13)
<b>trained templates</b>							
E	$\mathbb{R}^2$	92.79% (15)	98.56% (3)	94.58% (65)	92.50% (3)	50.62% (40)	92.75% (126)
F	$SE(2)$	100.0% (0)	98.56% (3)	99.67% (4)	100.00% (0)	90.12% (8)	99.14% (15)
<b>combinations of two templates</b>							
D + F		100.0% (0)	100.0% (0)	99.75% (3)	100.0% (0)	97.53% (2)	99.71% (5)
C + D		100.0% (0)	100.0% (0)	99.58% (5)	100.0% (0)	85.19% (12)	99.02% (17)
C + F		100.0% (0)	100.0% (0)	99.58% (5)	100.0% (0)	85.19% (12)	99.02% (17)
E + F		100.0% (0)	100.0% (0)	99.50% (6)	100.0% (0)	83.95% (13)	98.91% (19)
E + D		100.0% (0)	100.0% (0)	99.42% (7)	100.0% (0)	81.48% (15)	98.73% (22)
...							

**Results and Discussion.** Results of our ONH detection framework are given in Table 1. The results for single template methods are categorized in three categories: model templates, average templates and trained templates. For combinations of two templates only the best five combinations are shown.

Firstly, we observe that templates acting in the domain  $SE(2)$  of an orientation score considerably outperform their 2D equivalents. The orientation score templates put more emphasis on the pattern of blood vessels, rather than intensity features, and are therefore more robust against bright lesions and other pathologies. The advantage of our extension to  $SE(2)$  is best observed on the challenging STARE database, which contains a wide variety of severely pathological images.

Secondly, from Table 1 we see that average as well as trained templates outperform basic model templates, with the average templates slightly outperforming the trained templates. While the two best templates **D** and **F** individually give excellent performances, an even higher performance can be achieved in combining templates. With an accuracy of 99.71%, only 5 fails out of 1737 images, this combination **D+F** outperforms all other combinations of the templates used in this paper. Here we stress the crucial role of our proposed template optimization scheme that provides additional means for the construction of complementary templates; with single templates alone such high accuracy could not have been achieved. Exemplary results of matching with templates **D** and **F** are shown in in Fig. 4.

Thirdly, we note that when combining templates it is favorable to stay within the  $SE(2)$  framework. From the results we see that detection with  $\mathbb{R}^2$ -type templates is improved by combination with  $SE(2)$ -type templates. However, with respect to the use of single  $SE(2)$ -type templates, these combinations are not favorable (compare e.g. the result of **C**, **D** and **C+D**). In future work we will



**Fig. 4.** A selection of successful ONH detections in challenging images

therefore put more focus on the design of complementary  $SE(2)$ -type templates. As such, we will investigate the ability of regularization parameter  $D_{\theta\theta}$  to tune the template to anisotropic or isotropic structures (see Fig. 3) as a means to construct different/complementary  $SE(2)$  templates.

Finally, in Table 2 we compare our method to the state of the art on ONH detection (for full comparison to literature see [11, 14, and references therein]). Although our correlation-based method is rather basic in nature, it competes well with the state of the art. Only the method by Lu slightly outperforms our method by ones less false detection on the STARE database. Furthermore, as our detection framework merely relies on a sequence of correlations the method is highly parallelizable for speed optimization. Our current implementation (in Python) of the entire processing pipeline, including preprocessing (see Subsection 3.1), takes on average 1.1s per image.

**Table 2.** Comparison to state of the art: Optic nerve head detection results (number of fails in parentheses). The most recent five methods were selected for comparison. For a full comparison to literature see [11].

Database	Size	Proposed	Ramakanth et al. [11]	Yu et al. [8]	Lu et al. [7]	Lu. [9]	Mahfouz et al. [10]
DRIVE	40	100.0% (0)	100.0% (0)	-	97.5% (1)	-	100.0% (0)
STARE	81	97.53% (2)	93.83% (5)	-	96.3% (3)	98.77% (1)	92.59% (6)
MESSIDOR	1200	99.75% (3)	99.42% (7)	99.0% (12)	-	99.75% (3)	-
Av. time (s)		1.1	0.21	4.7	40	5	0.65

## 5 Conclusion

In this paper we have extended the concept of object detection via (normalized) cross correlation in the image domain  $\mathbb{R}^2$ , to the domain  $\mathbb{R}^2 \times S^1$  of orientation scores. The extension allows for the efficient detection of orientation patterns, while staying in the intuitive and efficient framework of template matching via cross correlation. Furthermore we have presented a method for the construction of templates to be used in this matching framework. The method was tested in the application to optic nerve head detection in retinal images. Here we achieved a success rate of 99.71% on a set of 1737 images, with an average processing time of 1.1s per image. The method is generically applicable, and is especially beneficial for the detection of objects characterized by orientated/line structures.

**Acknowledgements.** This work is part of the Hé Programme of Innovation, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). Also, the research leading to these results has received funding from the ERC council under the EC's 7th Framework Programme (FP7/2007–2013) / ERC grant agr. No. 335555.

## References

1. Lewis, J.: Fast normalized cross-correlation. *Vision Interface* 10(1), 120–123 (1995)
2. Yoo, J.C., Han, T.: Fast normalized cross-correlation. *CSSP* 28(6), 819–843 (2009)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 886–893 (June 2005)
4. Duits, R., Felsberg, M., Granlund, G.H., ter Haar Romeny, B.M.: Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the euclidean motion group. *IJCV* 72(1), 79–102 (2007)
5. Sinthanayothin, C., Boyce, J.F., Cook, H.L., Williamson, T.H.: Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images. *The British Journal of Ophthalmology* 83(8), 902–910 (1999)
6. Aquino, A., Gegundez-Arias, M.E., Marin, D.: Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques. *IEEE TMI* 29(11), 1860–1869 (2010)
7. Lu, S., Lim, J.: Automatic optic disc detection from retinal images by a line operator. *Biomedical Engineering, IEEE TBME* 58(1), 88–94 (2011)
8. Yu, H., et al.: Fast localization and segmentation of optic disk in retinal images using directional matched filtering and level sets. *IEEE TITB* 16(4), 644–657 (2012)
9. Lu, S.: Accurate and efficient optic disc detection and segmentation by a circular transformation. *IEEE TMI* 30(12), 2126–2133 (2011)
10. Mahfouz, A., Fahmy, A.: Fast localization of the optic disc using projection of image features. *IEEE TIP* 19(12), 3285–3289 (2010)
11. Ramakanth, S.A., Babu, R.V.: Approximate nearest neighbour field based optic disk detection. *Computerized Medical Imaging and Graphics* 38(1), 49–56 (2014)
12. Jonas, J.B., Budde, W.M., Panda-Jonas, S.: Ophthalmoscopic Evaluation of the Optic Nerve Head. *Survey of Ophthalmology* 43(4) (1999)
13. Hubbard, L.D., et al.: Methods for evaluation of retinal microvascular abnormalities associated with hypertension/sclerosis in the Atherosclerosis Risk in Communities Study. *Ophthalmology* 106(12), 2269–2280 (1999)
14. Bekkers, E., Duits, R., ter Haar Romeny, B.: Optic nerve head detection via group correlations in multi-orientation transforms. In: *Campilho, A., Kamel, M. (eds.) ICIAR 2014. LNCS*, vol. 8815, pp. 293–302. Springer, Heidelberg (2014)
15. Freeman, W., Adelson, E.: The design and use of steerable filters. *IEEE TPAMI* 13(9), 891–906 (1991)
16. Citti, G., Sarti, A.: A cortical based model of perceptual completion in the roto-translation space. *JMIV* 24(3), 307–326 (2006)
17. Duits, R., Franken, E.: Left-invariant parabolic evolutions on  $SE(2)$  and contour enhancement via invertible orientation scores part I: Linear left-invariant diffusion equations on  $SE(2)$ . *Quarterly of Applied Mathematics* 68(2), 255–292 (2010)
18. Foracchia, M., Grisan, E., Ruggeri, A.: Luminosity and contrast normalization in retinal images. *MEDIA* 9(3), 179–190 (2005)

## A Explicit Expression of the Regularization Matrix

Matrix  $R$  for regularization of templates in  $SE(2)$ , see Section 2.3, is given by

$$R = D_{11}R_\xi + D_{22}R_\eta + D_{33}R_\theta, \quad (22)$$

with regularization matrix

$$\begin{aligned} R_\xi = & \left( R_\xi^{I s_k} \otimes R_\xi^{I s_l} \otimes R_\xi^{I s_m} \right) + \left( R_\xi^{II s_k} \otimes R_\xi^{II s_l} \otimes R_\xi^{II s_m} \right) \\ & + \left( R_\xi^{III s_k} \otimes R_\xi^{III s_l} \otimes R_\xi^{III s_m} \right) + \left( R_\xi^{IV s_k} \otimes R_\xi^{IV s_l} \otimes R_\xi^{IV s_m} \right) \end{aligned} \quad (23)$$

of which the elements of the matrices are given by

$$\begin{aligned} R_\xi^{I s_k}(k, k') &= -\frac{1}{s_k} \frac{\partial^2 B^{2n+1}}{\partial u^2}(k' - k), & R_\xi^{I s_l}(l, l') &= s_l B^{2n+1}(l' - l), \\ R_\xi^{I s_m}(m, m') &= \int_0^\pi \cos^2(\theta) B^n\left(\frac{\theta}{s_m} - m\right) B^n\left(\frac{\theta}{s_m} - m'\right) d\theta, \\ R_\xi^{II s_k}(k, k') &= -R_\xi^{III s_k}(k, k') = \frac{\partial B^{2n+1}}{\partial u}(k' - k), \\ R_\xi^{II s_l}(l, l') &= -R_\xi^{III s_l}(l, l') = -\frac{\partial B^{2n+1}}{\partial v}(l' - l), \\ R_\xi^{II s_m}(m, m') &= R_\xi^{III s_m}(m, m') = \int_0^\pi \cos(\theta) \sin(\theta) B^n\left(\frac{\theta}{s_m} - m\right) B^n\left(\frac{\theta}{s_m} - m'\right) d\theta, \\ R_\xi^{IV s_k}(k, k') &= s_k B^{2n+1}(k' - k), & R_\xi^{IV s_l}(l, l') &= -\frac{1}{s_l} \frac{\partial^2 B^{2n+1}}{\partial v^2}(l' - l), \\ R_\xi^{IV s_m}(m, m') &= \int_0^\pi \sin^2(\theta) B^n\left(\frac{\theta}{s_m} - m\right) B^n\left(\frac{\theta}{s_m} - m'\right) d\theta, \end{aligned} \quad (24)$$

with regularization matrix

$$\begin{aligned} R_\eta = & \left( R_\xi^{II s_k} \otimes R_\xi^{II s_l} \otimes R_\xi^{IV s_m} \right) - \left( R_\xi^{II s_k} \otimes R_\xi^{II s_l} \otimes R_\xi^{II s_m} \right) \\ & - \left( R_\xi^{III s_k} \otimes R_\xi^{III s_l} \otimes R_\xi^{III s_m} \right) + \left( R_\xi^{IV s_k} \otimes R_\xi^{IV s_l} \otimes R_\xi^{I s_m} \right), \end{aligned} \quad (25)$$

and with regularization matrix

$$R_\theta = (R_\theta^{s_k} \otimes R_\theta^{s_l} \otimes R_\theta^{s_m}), \quad (26)$$

of which the elements of the matrices are given by

$$\begin{aligned} R_\theta^{s_k}(k, k') &= s_k B^{2n+1}(k' - k), \\ R_\theta^{s_l}(l, l') &= s_l B^{2n+1}(l' - l), & R_\theta^{s_m}(m, m') &= -\frac{1}{s_m} \frac{\partial^2 B^{2n+1}}{\partial w^2}(m' - m). \end{aligned} \quad (27)$$

Note that the four separate terms  $I - IV$  of Eq. (23) arise from the left invariant derivative  $\partial_\xi$ :  $\left| \frac{\partial T}{\partial \xi} \right|^2 = \left| \cos(\theta) \frac{\partial T}{\partial x} + \sin(\theta) \frac{\partial T}{\partial y} \right|^2$ .

# A Technique for Lung Nodule Candidate Detection in CT Using Global Minimization Methods

Nóirín Duggan<sup>1,2</sup>, Egil Bae<sup>3</sup>, Shiwen Shen<sup>4</sup>, William Hsu<sup>4</sup>, Alex Bui<sup>4</sup>,  
Edward Jones<sup>1</sup>, Martin Glavin<sup>2</sup>, and Luminita Vese<sup>3</sup>

<sup>1</sup> Electrical and Electronic Engineering, National University of Ireland, Galway, Ireland

<sup>2</sup> Department of Mathematics, Simon Fraser University, BC, Canada

<sup>3</sup> Department of Mathematics, University of California, Los Angeles, USA

<sup>4</sup> Department of Radiological Sciences, Medical Imaging Informatics Group,  
University of California, Los Angeles, USA

**Abstract.** The first stage in computer aided pulmonary nodule detection schemes is a candidate detection step designed to provide a simplified representation of the lung anatomy, such that features like the lung wall, and large airways are removed leaving only data which has greater potential to be a nodule. Nodules which are connected to blood vessels tend to be characterized by irregular geometrical features which can result in their remaining undetected by rule-based classifiers relying only local image metrics. In the current paper a novel approach for lung nodule candidate detection is proposed based on the application of global segmentation methods combined with mean curvature minimization and simple rule-based filtering. Experimental results indicate that the proposed method can accurately detect nodules displaying a diverse range of geometrical features.

## 1 Introduction

Every year, deaths due to lung cancer outnumber those related to other types of cancers around the world [1]. The most important indicator of the disease is the presence of pulmonary lung nodules [2,3], the early detection of which is essential to increase the chances of successful treatment [4].

The most popular modality for imaging the thorax is Computed Tomography (CT) [2]. Currently the most common method for quantifying lesion development using CT is through manual detection and measurement of the nodule diameter. In addition to being error-prone and subjective [5], this technique is limiting because a 1-D measure is used to describe a 3-D non-symmetric, non-spherical object. At the same time, manually characterizing the tumor using all of the 3-D data available would be extremely time-consuming [6].

Numerous studies have shown that computer aided detection (CAD) systems can effectively assist radiologists in detecting lung nodules [7,8,9,10,11,12]. In studies by Martin et al. [13] and Lee et al. [14] it was shown that the sensitivity



of CAD systems for detecting small, isolated nodules was greater than achieved by a radiologist, but sensitivity was lower than that of a radiologist for nodules with vascular attachment.

In [15] a nodule detection scheme is presented in which the first step is lung segmentation. This is achieved by thresholding the lung volume on a frame-by-frame basis. Noting that the volume histogram displays 2 prominent peaks, corresponding respectively to pixels inside the lungs and to pixels representing soft tissue and bone, for each frame, the authors in [15] select as a threshold the broad minimum existing between these peaks. The next step is a corrective stage which has the purpose of excluding structures such as airways and including juxta-pleura nodules (*i.e.* lung wall-connected) excluded by the thresholding step. To re-include juxta-pleura nodules they apply morphological opening and to exclude the airways, they apply a 2-D region growing technique. This is followed by a region-labeling technique designed to group contiguous structures in three dimensions. Finally, to obtain the candidacy mask the authors apply a volume threshold to these contiguous structures.

In Tan *et al.* [16] the first step is lung segmentation, performed using a similar technique as proposed in [15]. In the next step the authors compute the divergence of normalised gradient (DNG) of the volume to estimate the center of nodules, they then use this in combination with nodule and vessel enhancement filters proposed in [17,18] to detect nodule candidates. To obtain the nodule candidacy mask, the authors apply a different thresholding/filtering combination to each nodule type; *i.e.* isolated, juxtavascular (or vessel-connected), and juxtapleural nodules. For example for isolated nodules they apply a threshold of -600 Hounsfields Units (a quantitative scale for describing radiodensity) to the output of the lung segmentation, they then apply the nodule enhancement filter to this result. Subsequently another (gray level) threshold of 6 is applied to this nodule enhanced image. The output of this system is then combined with the result of the DNG method. Finally to this result, another volume threshold of 9 voxels is applied. The outputs of this last step are taken to be the isolated nodule candidates. The steps to extract both juxtavascular and juxtapleural nodules are similar to those outlined above and similarly involve a specific set of threshold parameters. The result of the procedure described above is multiple thresholded volumes consisting of nodule clusters corresponding to isolated, juxtavascular, and juxtapleural nodules. A logical OR-ing operation is then carried out to consolidate the results in one volume. In [19], the detection scheme starts with the generation of a lung mask in a scheme similar to that proposed in [15]. The authors then apply multi-level thresholding [15] to the remainder of the volume to produce multiple 3-D lung nodule candidate masks. To remove vessels, they apply a morphological opening operation with specific radius to each mask, which is followed by another rule based filter (with sphericity and area criteria) to remove false positives. The final nodule mask is generated by logically OR-ing these intermediate masks. The thresholds as well as the radii of the structuring elements are determined empirically. In [20] the authors

apply a 2-D filter to each axial slice image to highlight structures similar to discs or half-discs. To reduce the false positives, six 3-D features based on size, compactness, sphericity and gradient-intensity, are calculated for each candidate. The authors use a support vector machine (SVM) to classify the data.

In consecutive CT slices, blood vessels often appear as circular objects closely resembling nodules. Using this fact, the authors in [21] proposed a scheme to generate multiple 2-D images based on different spherical viewpoints of each 3-D nodule candidate. The authors show that these different viewpoints allow the noncircular linear structure of components corresponding to vessels to be more easily identified. The authors then combined features generated from these images with 3-D features such as diameter and compactness [22]. They then employ a linear classifier [22,23] to classify the results.

Murphy *et al.* [24] proposed a detection scheme which uses shape index and curvedness to detect nodule candidates. Using these features the authors filtered the datasets to produce seed points in areas of high filter response and expanded these points using hysteresis thresholding to produce region clusters. To reduce false positives, they applied two consecutive classification steps using k-Nearest-Neighbour. In analysing the results according to nodule size, the authors reported that for nodules with a diameter greater than 8.6mm, the sensitivity rate was under 45%. These findings highlight the difficulty in detecting nodules characterized by irregular shapes by means of local image features alone, a fact which the authors themselves acknowledged.

The purpose of this paper is to present an algorithm for the detection of the lung lobe interior with particular emphasis on detecting nodules with vascular attachment. The output of the algorithm is a set of regions that can be analyzed further either manually or with an advanced classifier to determine whether they represent a true nodule. As can be observed in the previous review, several of the proposed schemes make use of a combination of multi-thresholding methods and as well as spherical shape filters to isolate nodules [15,19,21,16], however as noted in [25], when nodules are connected to other high density structures, separating them with intensity thresholds alone is in most cases, impossible. In the same way, incorporating spherical constraints early into a detection scheme can be limiting especially in the case of nodules which exhibit a high degree of vascular attachment and which therefore represent quite a complex geometry. In this paper, we make use of more sophisticated variational models [26] and a recently developed efficient convex optimization algorithm for obtaining solutions numerically. The entire algorithm consists of several successive steps that are described in detail below.

## 2 Methodology

A challenge for obtaining a good segmentation is that many objects inside the lung have very similar intensity distributions to the nodule, in particular blood vessels and the chest wall. This makes it difficult to separate the intensity profile of the nodule from other tissue classes using a multiregion segmentation framework. We develop an algorithm where a two region segmentation model is first

used to capture the nodules, chest wall and other objects of similar intensity in a foreground region, and the air and remaining objects in a background region. The remaining parts of the algorithm attempt to separate the potential nodules from the rest of the tissue that were captured in the foreground region. This concerns mainly the chest wall and surrounding blood vessels, which is handled in two separate steps.

### 2.1 Computation of a Global 2-Phase Segmentation Output

The first step of our method aims to extract the chest wall, nodules, blood vessels and other tissue of similar intensity values into one region using the active contour model with two regions [26]:

$$\min_{S, c_1, c_2} \int_{\Omega \setminus S} |I(x) - c_1|^2 dx + \int_S |I(x) - c_2|^2 dx + \nu |\partial S|. \tag{1}$$

In recent work, efficient algorithms have been proposed for computing global minimizers to this model. In [27] it was shown that (1) can be exactly minimized via the convex problem

$$\min_{\phi(x) \in [0,1]} \int_{\Omega} |I(x) - c_1|^2 \phi(x) + |I(x) - c_2|^2 (1 - \phi(x)) dx + \nu \int_{\Omega} |\nabla \phi(x)| dx. \tag{2}$$

It was shown that if  $\phi^*$  is a minimizer of (2) and  $t \in (0, 1]$  is any threshold level, the partition  $S = \{x \in \Omega : \phi(x) \geq t\}$ ,  $\Omega \setminus S = \{x \in \Omega : \phi(x) < t\}$  is a global minimizer to the model (1). The binary function

$$\phi^t(x) := \begin{cases} 1, & \phi(x) \geq t \\ 0, & \phi(x) < t \end{cases}, \tag{3}$$

is the characteristic function of the region  $S$ .

We make use of an efficient augmented Lagrangian algorithm for solving a dual formulation of (1) proposed in [28,29], which could be interpreted as a maximum flow problem. By introducing a Lagrange multiplier for the flow conservation constraint, the following augmented Lagrangian primal-dual problem was obtained

$$\sup_{\phi} \inf_{p_s, p_t, p} \int_{\Omega} p_s dx + \int_{\Omega} \phi (\operatorname{div} p - p_s + p_t) dx - \frac{c}{2} \|\operatorname{div} p - p_s + p_t\|^2 \tag{4}$$

such that

$$|p(x)|_2 \leq \nu, \quad \forall x \in \Omega; \quad p_s(x) \leq |I(x) - c_1|^2, \quad p_t(x) \leq |I(x) - c_2|^2, \quad \forall x \in \Omega \tag{5}$$

where  $p_s, p_t : \Omega \mapsto \mathbb{R}$  and  $p : \Omega \mapsto \mathbb{R}^N$  and  $N$  is the dimension of  $\Omega$ . By applying the augmented Lagrangian method, the following algorithm was derived for solving (2)

$$- p_s^{k+1} := \arg \max_{p_s(x) \leq |I(x) - c_1|^2 \forall x \in \Omega} \left( \int_{\Omega} p_s dx - \frac{c}{2} \|p_s - p_t^k - \operatorname{div} p^k + \phi^k / c\|^2 \right)$$

which can easily be computed pointwise in closed form.

$$- p^{k+1} := \arg \max_{\|p\|_{\infty} \leq \nu} -\frac{c}{2} \|\operatorname{div} p - p_s^{k+1} + p_t^k - \phi^k / c\|^2,$$

where  $\|p\|_{\infty} = \sup_{x \in \Omega} |p(x)|_2$ . This problem can either be solved iteratively or approximately in one step via a simple linearization [29]. In our implementation we used the linearization.

$$- p_t^{k+1} := \arg \max_{p_t(x) \leq |I(x) - c_2|^2 \forall x \in \Omega} -\frac{c}{2} \|p_t - p_s^{k+1} + \operatorname{div} p^{k+1} - \phi^k / c\|^2$$

This problem can also easily be computed in closed form pointwise.

$$- \phi^{k+1} = \phi^k - c (\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1});$$

- Set  $k = k + 1$  and repeat.

The output  $\phi$  at convergence will be a solution to (2) and one can obtain a partition which solves (1) by the thresholding procedure described in the previous section. More details can be found in [29].

In simple cases, the two region segmentation algorithm may separate out the nodule as a single connected component. In more difficult scenarios, the nodule region may be connected to either the chest wall or surrounding blood vessels.

## 2.2 Lung Wall Removal Process

The 3-D global segmentation described above essentially segments the volume into 2 classes: tissue and air. The next step is to separate the lung wall from the structures that are interior to the lung. This is done using a combination of connected component labeling as well as morphological opening.

First a connected component labeling operation is used to identify the largest component in the volume. This step identifies the lung wall together with additional structures connected via vessels to the lung wall. A correction step which consists of a morphological opening operation [30] is used to remove these additional structures. The result of the morphological opening operation, which corresponds to the lung wall is then subtracted from the 3-D segmentation result leaving just structures in the interior lung lobe.

## 2.3 Nodule Separation Scheme

We address the issue of separating the nodules from surrounding tissue by applying mean curvature minimization using the method of Merriman-Bence-Osher (MBO) [31] to the output of the segmentation scheme. The effect of this scheme is to ‘simplify’ the underlying structures of the nodule candidates (or vessels); essentially, through the diffusion process, a spiculated mass will become

smoother/more spherical, while structures connected to each other by (relatively) thin connections will be separated.

Let  $\phi^0$  denote the binary function indicating the segmentation result after removal of the lung wall. The MBO algorithm applied to  $\phi^0$  is a discrete time approximation of mean curvature motion and can be described as follows:

For  $k = 1, 2, \dots, K$

$$\psi = G_\sigma * \phi^k \quad (6)$$

$$\phi^{k+1}(x) = \begin{cases} 1, & \text{if } \psi(x) \geq 0.5 \\ 0, & \text{if } \psi(x) < 0.5 \end{cases} \cdot \quad (7)$$

Step (6) above is time step of the heat equation, which is equivalent to convolution with the Gaussian kernel  $G_\sigma$ , and can be solved efficiently by the fast Fourier transform (FFT). After each MBO iteration, a rule based classifier is applied to each connected component of the result, to check if the component is a nodule candidate. The rule based classifier is described in the next section. The number of iterations  $K$  is set in advance to prevent too much smoothing.

## 2.4 Rule Based Classifier

The effect of the MBO step is to make a spiculated mass smoother and more spherical in shape, which allows structures to be identified as potential nodules using simple geometric features. The final step in determining nodule candidacy is the application of a simple rule-based classifier, in which candidacy is determined by the following features: area, volume, circularity, elongation.

The following definitions are used for each feature: assuming that the nodules are spherical, the area and volume of the nodule candidates can be computed using the standard formulae: Area =  $\pi r^2$ ; Volume =  $3/4\pi r^3$ .

Elongation is defined simply as the ratio of the largest dimension in the x,y or z direction over the minimum dimension in any direction i.e.

$$\text{Elongation} = \frac{\max([x\text{Length}, y\text{Length}, z\text{Length}])}{\min([x\text{Length}, y\text{Length}, z\text{Length}])}$$

Circularity is defined as:

$$\text{Circularity} = \frac{4\pi\text{Area}}{\text{Perimeter}^2}$$

In the above equation, ‘Area’ and ‘Perimeter’ are calculated using the median slice of the connected component. The respective maximum and minimum thresholds for each feature are listed in section 3. This step closely follows the method proposed by Choi et al. in [32] and further details can be found therein.

## 2.5 Summary of the Complete Algorithm

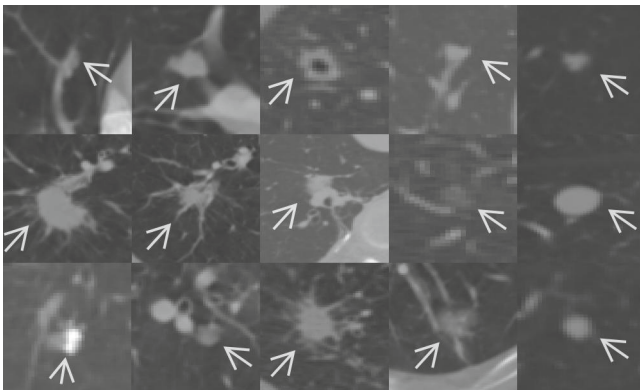
In summary, the proposed nodule candidacy detection scheme comprises the following steps; first a global 2 phase segmentation is performed, which segments

the volume into 2 classes: tissue and air. The next step is to further segment the tissue into lung wall and interior lobe data; this is done using morphological techniques. The main part of the proposed scheme is the use of mean curvature smoothing to isolate vascular connected nodules. The detection step is carried out by applying the rule-based classifier once before the MBO smoothing and subsequently on each connected component after each MBO iteration. The final lung nodule candidacy mask is obtained by logically OR-ing all of the intermediate detection results. The entire algorithm is outlined below.

Input: 3D CT lung image

1. Obtain two region segmentation by global minimization of (2).
2. Remove lung wall from the segmentation result as described in section 2.2 and let  $\phi^0$  denote indicator function of the remaining region.
3. Apply rule based classifier to each remaining connected component as in section 2.4
  - Store positive connected components as potential nodule candidates
4. For iterations  $k=1, \dots, K$ :
  - Apply one step of MBO scheme (6), (7) to obtain  $\phi^k$ .
  - Apply rule based classifier on each component of  $\phi^k$ . If positive: store connected component as potential nodule candidate and for all points  $x$  inside connected component and set indicator function  $\phi^k(x) = 0$ .

Output: Set of nodule candidates, represented by a binary function.

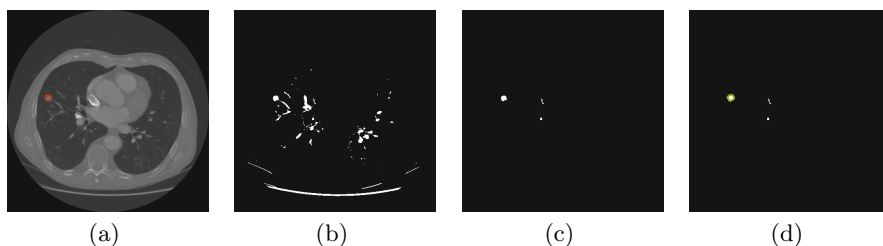


**Fig. 1.** Examples of nodules included in the test subset. The majority of the nodules in the test set exhibit some degree of attachment to surrounding vascular tissue; isolated nodules were also included (as can be observed in the rightmost column).

### 3 Experiments

A test set of 16 datasets were selected from the lung image consortium (LIDC) database [33,34] which consisted of both nodules exhibiting vascular attachment as well as isolated nodules. Our standard of reference were the expert annotations provided with this database. The test set includes a total of 27 nodules.

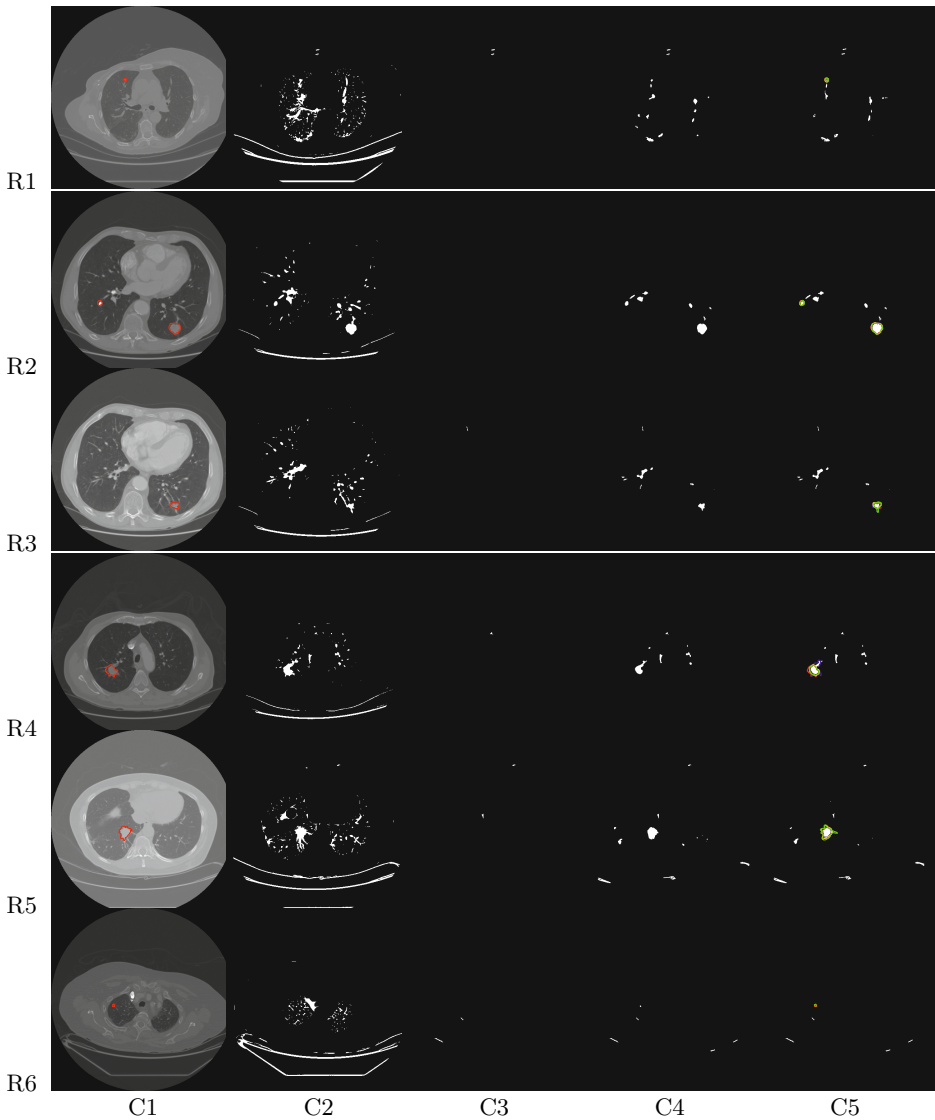
Figure 1 shows examples of nodules in the test subset. Figure 2 highlights the normal operation of the rule based classifier. It can be observed in figure 2(b) that the nodule in question is isolated from other structures in the lung after application of the first segmentation step and the filter easily selects the nodule. The results of the algorithm are demonstrated in figure 3, which displays 6 sample slices from the test set. Compared with the data set in figure 2(a), each nodule in figure 3 exhibits connectivity with surrounding tissue. The first segmentation step is unable to sufficiently separate the nodules from surrounding tissue in these cases due to their similar intensity profiles. Application of the MBO scheme has the effect of either removing fine structures attached to the nodule, such as fine blood vessels, or splitting the regions into two or more geometrically simpler components, one of which encompasses the nodule region.



**Fig. 2.** Example of normal operation of rule-based classifier: (a) Annotated Data indicating nodule (b) Initial Segmentation Results + lung wall removal (c) Corresponding Detection Results (d) Corresponding Detection Results with annotation

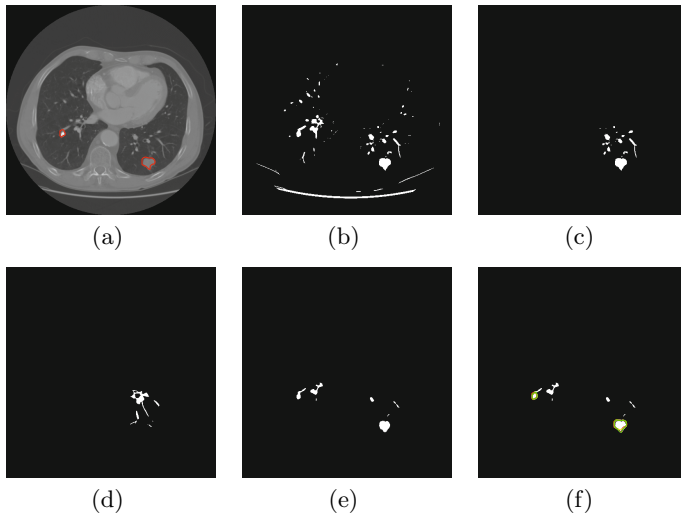
In figures 4 and 5, two examples of nodules are shown which in 4(a) and 5(a) appear to be isolated. Using only steps 1-3 of the algorithm results in the nodules remaining undetected because, as adjacent slices reveal, there is some degree of vascular attachment. The nodules are detected (Figures 4(e) and 5(e)) when the MBO scheme (step 4) is used as part of the detection scheme.

Figure 6 shows a case where the detection scheme fails to detect a nodule. In this case, the degree of connectivity between the nodule and surrounding structures was too extensive for the proposed method to work. Figure 6(c) shows the connected structure of which the nodule forms a part. Ongoing work is focused on a more complex algorithm for removing the lung wall taking into account prior geometrical knowledge about the shape, such that potential nodules attached to the lung wall gets disconnected.

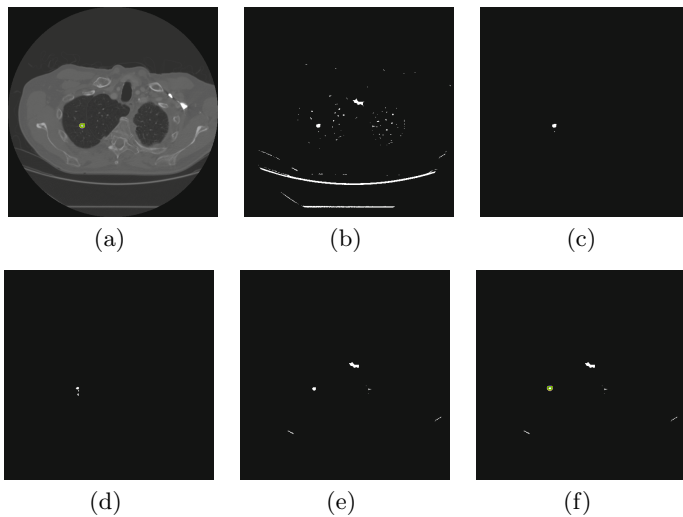


**Fig. 3.** Results achieved by the method on 6 sample frames (Rows R1 R6). Each row shows an example of a nodule with vascular attachment. C1: The original data with expert annotation. C2: 2 phase global segmentation result C3: Results of the rule-based detection method C4: Detection results post MBO processing C5: Detection results with superimposed expert annotation.

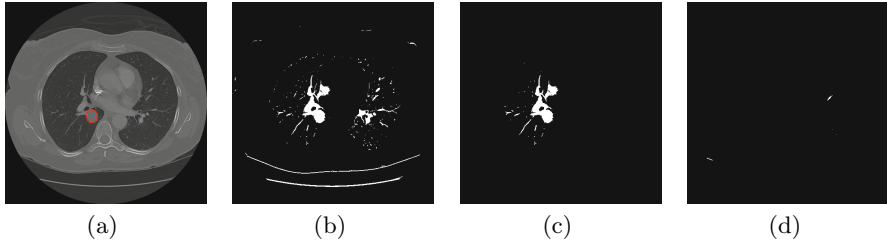




**Fig. 4.** Example of a nodule which is processed using only the rule-based classifier (i.e. using only steps 1-3 of the algorithm) (a) Original Annotated dataset (b) slice 71 of dataset: nodule looks well delineated from surrounding structures but is actually part of a large connected component (c) the connected component which contains the nodule (d) an adjacent slice (e) Step 4: Detection with MBO (f) Annotation overlaid on result



**Fig. 5.** Second example of a nodule which is processed using only the rule-based classifier (i.e. using only steps 1-3 of the algorithm) (a) Original Annotated dataset (b) nodule looks well isolated from surrounding structures but is actually part of a large connected component (c) connected component which includes the nodule (d) an adjacent slice (e) Step 4: Detection with MBO (f) Annotation overlaid on result



**Fig. 6.** Example of a case where MBO-detection scheme fails (a) Original Annotated dataset (b) 2-phase segmentation (c) Connected component (d) Detection with MBO - failure

The following empirically derived parameter values were used in the experiments: the length parameter,  $\nu$  was set to  $1e-12$ , for the 3-D global segmentation. The initial estimates for the mean value of both regions,  $c_1$  and  $c_2$ , were set to 0.3 and 0.6 respectively. The morphological opening operation used to remove the lung wall was carried out using a spherical kernel of radius 13. With respect to the MBO scheme, with the exception of 1 dataset,  $\sigma$  for the Gaussian kernel was set to 1, while the maximum number of iterations used was 20. For one dataset, sigma was reduced to 0.7, with the number iterations remaining at 20. The thresholds used for the rule based classifier were the following: maximum diameter,  $T_{max}^d$ , was set to 30, minimum diameter  $T_{min}^d$  was set to 3, correspondingly the maximum area threshold was  $(T_{max}^d/2)^2\pi$ , minimum area threshold was  $(T_{min}^d/2)^2\pi$ , maximum volume threshold was set to  $3(T_{max}^d/2)^3\pi/4$ , minimum threshold was set to  $3(T_{min}^d/2)^3\pi/4$ , maximum elongation was set to 4, while minimum circularity was set to  $\frac{1}{6}$ .

Minimum and maximum diameter thresholds for the rule-based classifier were chosen based on data in the LIDC dataset, where 97% of the nodules recorded have a diameter in the 3-30mm range (with 86% in the 3-12mm range and a further 11% in the 12-30mm range). With respect to the test set used in these experiments, 80% of the nodules were in the 3-12mm diameter range, with the remaining 20% in the 12-30mm range. The greater representation of larger nodules in our test set was a design decision taken in response to the previously discussed findings in [24], in which it was reported that larger nodules tended to be characterized by irregular shapes and thus were harder to detect.

As described, the MBO parameters were changed for one dataset. This dataset presented a nodule with a diameter of 6mm that also exhibited a connection with neighboring blood vessels. For this nodule, a sigma value of 1 represented an over-smoothing and consequent detection failure, while a sigma value of 0.7 resulted in accurate detection. To make the system more robust with respect to parameter choice a smaller sigma value can be used and the number of iterations increased.

The experiments show that the algorithm successfully detected all but one of the nodules present (see figure 6). This resulted in an average detection rate of 96%, while an average of 16 false positives were detected per scan. Without the MBO smoothing step the detection rate was 44%.

## 4 Conclusion

Several studies [24,14,13] have highlighted the difficulty in detecting larger nodules in lung images, which tend to be characterized by greater shape diversity. In the current paper an algorithm was described for handling this task using variational and PDE based methods. The algorithm was tested on 16 datasets containing 27 nodules with various degrees of attachment to surrounding tissue. A 96% detection rate was obtained. These initial results show that the proposed method has the potential to be an effective module in an automated detection pipeline.

Future work is focused on a more complex algorithm for separating the most difficult nodules from the lung wall to improve the detection rate further, and a finer segmentation step applied in the end, using the detected nodule candidates as location prior information.

## References

1. American Cancer Society: Cancer Facts and Figures 2014 (2014)
2. Goo, J.M.: A computer-aided diagnosis for evaluating lung nodules on chest CT: the current status and perspective. *Korean Journal of Radiology* 12(2), 145–155 (2011)
3. Weir, H., Thun, M., Hankey, B., Ries, L., Howe, H., Wingo, P., Jemal, A., Ward, E., Anderson, R., Edwards, B.: Annual report to the nation on the status of cancer, 1975-2000. *J. National Cancer Institute* 95, 1276–1299 (2003)
4. Henschke, C., McCauley, D., Yankelevitz, D., Naidich, D., McGuinness, G., Miettinen, O., Libby, D., Pasmantier, M., Koizumi, J., Altorki, N., Smith, J.: Early lung cancer action project: overall design and findings from baseline screening. *Lancet* 354, 99–105 (1999)
5. Moltz, J.H., Bornemann, L., Kuhnigk, J.M., Dicken, V., Peitgen, E., Meier, S., Bolte, H., Fabel, M., Bauknecht, H.C., Hittinger, M., Kießling, A., Pusken, M., Peitgen, H.O.: Advanced segmentation techniques for lung nodules, liver metastases, and enlarged lymph nodes in CT scans. *IEEE Journal of Selected Topics in Signal Processing* 3, 122–134 (2009)
6. Bornemann, L., Dicken, V., Kuhnigk, J.M., Wormanns, D., Shin, H.O., Bauknecht, H.C., Diehl, V., Fabel, M., Meier, S., Kress, O., Krass, S., Peitgen, H.O.: Oncotreat: a software assistant for cancer therapy monitoring. *International Journal of Computer Assisted Radiology and Surgery* 1, 231–242 (2007)
7. Sahiner, B., Chan, H.P., Hadjiiski, L.M., Cascade, P.N., Kazerooni, E.A., Chughtai, A.R., Poopat, C., Song, T., Frank, L., Stojanovska, J., Attili, A.: Effect of CAD on radiologists' detection of lung nodules on thoracic CT scans: analysis of an observer performance study by nodule size. *Acad. Radiol.* 16, 1518–1530 (2009)

8. Park, E.A., Goo, J.M., Lee, J.W., Kang, C.H., Lee, H.J., Lee, C.H., Park, C.M., Lee, H.Y., Im, J.G.: Efficacy of computer-aided detection system and thin-slab maximum intensity projection technique in the detection of pulmonary nodules in patients with resected metastases. *Invest. Radiol.* 44, 105–113 (2009)
9. Hirose, T., Nitta, N., Shiraishi, J., Nagatani, Y., Takahashi, M., Murata, K.: Evaluation of computer-aided diagnosis (CAD) software for the detection of lung nodules on multidetector row computed tomography (MDCT): JAFROC study for the improvement in radiologists' diagnostic accuracy. *Acad. Radiol.* 15, 1505–1512 (2008)
10. Goo, J.M., Kim, H.Y., Lee, J.W., Lee, H.J., Lee, C.H., Lee, K.W., Kim, T.J., Lim, K.Y., Park, S.H., Bae, K.T.: Is the computer-aided detection scheme for lung nodule also useful in detecting lung cancer? *Journal of Computer Assisted Tomography* 32, 570–575 (2008)
11. Beigelman-Aubry, C., Raffy, P., Yang, W., Castellino, R.A., Grenier, P.A.: Computer-aided detection of solid lung nodules on follow-up MDCT screening: evaluation of detection, tracking, and reading time. *AJR Am. J. Roentgenol.* 189, 948–955 (2007)
12. Awai, K., Muraio, K., Ozawa, A., Komi, M., Hayakawa, H., Hori, S., Nishimura, Y.: Pulmonary nodules at chest CT: effect of computer-aided diagnosis on radiologists detection performance 1. *Radiology* 230, 347–352 (2004)
13. Marten, K., Engelke, C., Seyfarth, T., Grillhösl, A., Obenauer, S., Rummeny, E.: Computer-aided detection of pulmonary nodules: influence of nodule characteristics on detection performance. *Clinical Radiology* 60, 196–206 (2005)
14. Lee, J.W., Goo, J.M., Lee, H.J., Kim, J.H., Kim, S., Kim, Y.T.: The potential contribution of a computer-aided detection system for lung nodule detection in multidetector row computed tomography. *Invest. Radiol.* 39, 649–655 (2004) PMID: 15486524
15. Armato III, S.G., Giger, M.L., MacMahon, H.: Automated detection of lung nodules in CT scans: preliminary results. *Medical Physics* 28, 1552–1561 (2001)
16. Tan, M., Deklerck, R., Jansen, B., Bister, M., Cornelis, J.: A novel computer-aided lung nodule detection system for CT images. *Medical Physics* 38, 5630–5645 (2011)
17. Li, Q., Sone, S., Doi, K.: Selective enhancement filters for nodules, vessels, and airway walls in two- and three-dimensional CT scans. *Medical Physics* 30, 2040–2051 (2003)
18. Li, Q., Arimura, H., Doi, K.: Selective enhancement filters for lung nodules, intracranial aneurysms, and breast microcalcifications. *International Congress Series*, vol. 1268, pp. 929–934 (2004)
19. Messay, T., Hardie, R., Rogers, S.K.: A new computationally efficient CAD system for pulmonary nodule detection in CT imagery. *Medical Image Analysis* 14, 390–406 (2010)
20. Opfer, R., Wiemker, R.: Performance analysis for computer-aided lung nodule detection on lidc data. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6515 (2007)
21. Guo, W., Li, Q.: High performance lung nodule detection schemes in CT using local and global information. *Medical Physics* 39, 5157–5168 (2012)
22. Li, Q., Li, F., Doi, K.: Computerized detection of lung nodules in thin-section CT images by use of selective enhancement filters and an automated rule-based classifier. *Academic Radiology* 15, 165–175 (2008)
23. Li, Q., Doi, K.: Analysis and minimization of overtraining effect in rule-based classifiers for computer-aided diagnosis. *Medical Physics* 33, 320–328 (2006)

24. Murphy, K., van Ginneken, B., Schilham, A., de Hoop, B., Gietema, H., Prokop, M.: A large-scale evaluation of automatic pulmonary nodule detection in chest CT using local image features and k-nearest-neighbour classification. *Medical Image Analysis* 13, 757–770 (2009)
25. Kuhnigk, J.M., Dicken, V., Bornemann, L., Bakai, A., Wormanns, D., Krass, S., Peitgen, H.O.: Morphological segmentation and partial volume analysis for volumetry of solid pulmonary lesions in thoracic CT scans. *IEEE Transactions on Medical Imaging* 25, 417–434 (2006)
26. Chan, T., Vese, L.: Active contours without edges. *IEEE Transactions on Image Processing* 10, 266–277 (2001)
27. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics* 66, 1632–1648 (2006)
28. Yuan, J., Bae, E., Tai, X.C.: A study on continuous max-flow and min-cut approaches. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2217–2224 (2010)
29. 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, 559–587 (2014)
30. Giardina, C.R., Dougherty, E.R.: *Morphological Methods in Image and Signal Processing*. Prentice-Hall, Inc., Upper Saddle River (1988)
31. Merriman, B., Bence, J., Osher, S.: Diffusion generated motion by mean curvature. In: *Crystal Grower’s Workshop. MS Selected Letters*, pp. 73–83 (1993)
32. Choi, W.J., Choi, T.S.: Genetic programming-based feature transform and classification for the automatic detection of pulmonary nodules on computed tomography images. *Information Sciences* 212, 57–78 (2012)
33. Armato, S., McLennan, G., McNitt-Gray, M., Meyer, C., Yankelevitz, D., Aberle, D., Henschke, C., Hoffman, E., Kazerooni, E., MacMahon, H., Reeves, A., Croft, B., Clarke, L.: L.I.D.C.R. group, lung image database consortium: developing a resource for the medical imaging research community. *Radiology* 232, 739–748 (2004)
34. McNitt-Gray, M.F., Armato III, S.G., Meyer, C.R., Reeves, A.P., McLennan, G., Pais, R.C., Freymann, J., Brown, M.S., Engelmann, R.M., Bland, P.H., Laderach, G.E., Piker, C., Guo, J., Towfic, Z., Qing, D.P.Y., Yankelevitz, D.F., Aberle, D.R., van Beek, E.J., MacMahon, H., Kazerooni, E.A., Croft, B.Y., Clarke, L.P.: The lung image database consortium (LIDC) data collection process for nodule detection and annotation. *Academic Radiology* 14, 1464–1474 (2007)

# Hierarchical Planar Correlation Clustering for Cell Segmentation

Julian Yarkony<sup>1</sup>, Chong Zhang<sup>2</sup>, and Charless C. Fowlkes<sup>3</sup>

<sup>1</sup> Experian Data Lab, San Diego, CA  
julian.e.yarkony@gmail.com

<sup>2</sup> CellNetworks, University of Hiedelberg, Germany  
chong.zhang@iwr.uni-heidelberg.de

<sup>3</sup> Department of Computer Science  
University of California, Irvine  
fowlkes@ics.uci.edu

**Abstract.** We introduce a novel algorithm for hierarchical clustering on planar graphs we call “Hierarchical Greedy Planar Correlation Clustering” (HGPPC). We formulate hierarchical image segmentation as an ultrametric rounding problem on a superpixel graph where there are edges between superpixels that are adjacent in the image. We apply coordinate descent optimization where updates are based on planar correlation clustering. Planar correlation clustering is NP hard but the efficient PlanarCC solver allows for efficient and accurate approximate inference. We demonstrate HGPPC on problems in segmenting images of cells.

## 1 Introduction

We approach the problem of image segmentation in the framework of hierarchical segmentation where the goal is to group the pixels into a hierarchical structure where contiguous groups of pixels are divided and further subdivided. At the coarsest level of the hierarchy, all pixels are in the same region. At the finest level of the hierarchy each pixel is its own region. Each boundary that is present at a given level of the hierarchy is present in each finer level of the hierarchy.

Hierarchical segmentation can be understood as assigning confidence to various boundaries where boundaries present in coarser levels of the hierarchy are estimated to be more reliable. Hierarchical segmentation has been done primarily using agglomerative clustering, with the Ultrametric Contour Maps Algorithm being the state of the art (Arbelaez et al., 2011). Here we frame hierarchical segmentation as an ultrametric rounding problem (Ailon and Charikar, 2005; Yarkony, 2012).

Model the data to be clustered as the nodes of a graph where each pair of nodes is connected with an edge  $e$  that is associated with a real valued weight  $X_e$ . For any real value  $\alpha$  let  $Y_e^\alpha := [X_e \geq \alpha]$  where  $[ ]$  is the indicator function. Now consider the unweighted graph  $G^\alpha$  with edges connecting nodes only if  $Y_e^\alpha = 0$ . If  $X$  is an ultrametric then for all  $\alpha$  and  $e$ ,  $Y_e^\alpha = 0$  if and only if the pair of

nodes connected by  $e$  are in the same component of  $G^\alpha$ . Ultrametrics define a natural model of hierarchical grouping where the threshold  $\alpha$  specifies the level of the hierarchy. If  $\alpha$  is large then  $G^\alpha$  has few regions while if  $\alpha$  is small it has many regions. Edges present in  $G^\alpha$  are present in  $G^{\alpha+v}$  for all  $v > 0$ .

Given an initial graph and set of (sparse) edges where each edge  $e$  is associated with a real valued target  $T_e$ , the objective of ultrametric rounding is to assign a new set of values  $\{X_e\}$  to the edges which satisfies the property of being an ultrametric and is minimally distorted from the targets (either in an  $L^1$  or  $L^2$  sense). In our application nodes correspond to superpixels and edges indicate adjacency. Superpixels (Ren and Malik, 2003) are small compact groups of pixels which can be produced by various approaches. Superpixels are the most elementary unit in our hierarchical segmentation approach. We connect neighboring superpixels with an edge and associated score  $T_e$  that defines how strong the image boundary is locally between the two superpixels. Large  $T_e$  are associated with stronger visual indications of an edge between the superpixels connected by edge  $e$ . The goal of finding the closest ultrametric  $X$  to  $T$  can thus be interpreted as finding a hierarchical segmentation which is consistent with the local evidence encoded in  $T$ . Edges only connect nodes whose corresponding superpixels are immediately adjacent in the image. Thus our graph is planar which allows for many computational advantages which is the focus of this paper.

We focus on the application of segmenting cells in biological images. Cell segmentation is one of the prerequisite tasks in answering many biological questions related to both basic understanding of cell function and interpretation of pathological states. Recent emerging research efforts in diverse cell lines and microscopic imaging techniques require robust and automatic algorithms for performing segmentation, particularly in high-throughput experiments. While cell imaging with fluorescent labels or other chemical staining can provide contrast on objects of interest for easy segmentation, it is not ideal for studying cells under natural conditions. Without such dyes, cells are much harder to segment. Cells in brightfield or phase contrast images are only distinguishable by their outer membrane. Other major challenges of segmenting cells from these images are: touching cells, weak or broken boundaries, large variations on boundary pattern, and false boundaries due to artifacts or other sub-cellular structures.

## 2 Related Work

### 2.1 Related Work on Clustering

Hierarchical clustering has been considered since early machine learning. Agglomerative clustering is the primary way in which this has been approached in the domain of computer vision. The seminal ultrametric contour maps algorithm (UCM) (Arbelaez et al., 2011) is the clearest application of this approach in image segmentation. UCM associates with each pair of superpixels  $i, j$  a distance metric  $D_{ij}$ .  $D_{ij}$  is initially a function of image features. UCM initializes each superpixel as an independent region. UCM proceeds by merging the pair of adjacent regions whose average distance metric between superpixels across the

boundary is minimal. Usually this is average weighted by the length  $l_{ij}$  of the boundary between the superpixels. Let  $B(Q_1, Q_2)$  be the set of edges between the superpixels making up region  $Q_1$  and  $Q_2$ . The weighted average distance is computed as:

$$\bar{D}(Q_1, Q_2) = \frac{\sum_{[i,j] \in B(Q_1, Q_2)} l_{ij} D_{ij}}{\sum_{[ij] \in B(Q_1, Q_2)} l_{ij}} \quad (1)$$

(2)

In the UCM algorithm, when two regions  $Q_1$  and  $Q_2$  are merged, each edge between the superpixels spanning across the two regions is set to the average value  $\bar{D}(Q_1, Q_2)$ . This assure that the resulting set of distances forms an ultrametric. UCM continues grouping the pair of regions whose average distance is minimal until all superpixels are in the same region. UCM is a fast greedy method which is quite successful but does not claim to minimize the ultrametric distortion.

Ultrametric rounding for image segmentation has been explored in a regime in which each  $X_e$  may only take on a set of fixed discrete values (Yarkony, 2012) using the formulation of (Ailon and Charikar, 2005). Our work significantly departs from this line as it does not restrict  $X$  to take on a set of discrete values.

## 2.2 Related Work on Cell Segmentation

Many efforts have been devoted very recently in cell segmentation based on boundaries. In (Liu et al., 2014) cell segments are selected from a UCM-based hierarchical segmentation region candidates through an integer linear programming (ILP) formulation. Each region candidate has a score predicted from SVM classifier, that takes part of its input from a cell contour shape model. This technique tries to find the best segmented cells from multiple hierarchical layers. However, the dependency on a common cell shape may not likely to apply this technique on cells evolve or deform such as the fibroblast cells in (Wu et al., 2012). However, the fact that in (Wu et al., 2012) the segmentation is formulated as a partial matching problem between cell boundaries obtained from consecutive frames in time-lapse images limits its applicability to static images. An interactive cell segmentation approach to correct erroneous segmentation is proposed very recently (Su et al., 2014). It uses an augmented affinity graph to efficiently incorporate and propagate corrected labels for an updated partitioning of the superpixels. But this method explicitly uses phase retardation features (Su et al., 2013) to generate superpixels so as to enable efficient corrections on superpixel level. Yet another method in (Zhang et al., 2014a) combines detection of cell centers and clustering cell boundary points in an ILP fashion. But this method is primarily designed for cells with convex shapes with similar sizes.

## 3 Ultrametric Rounding

We start by formulating ultrametric rounding as an optimization problem. Consider a graph  $G$  with edges indexed by  $e$ .  $G$  is often a sparse graph meaning



that most pairs of nodes are not connected. We denote the desired ultrametric as  $X$  which is indexed by  $e$ . Here we assume  $X_e$  is a real valued in the range  $[0, 1]$ . For  $X$  to be an ultrametric it must be the case that if we remove the set of edges for which  $X_e$  is greater than any given value  $\alpha$  we do not remove any edges within a connected component of the resulting graph. This can be enforced by the constraint that for any cycle  $C$  in our graph containing an edge  $e$  between adjacent superpixels separated by a boundary (a pair where  $X_e \geq \alpha$ ), at least one other boundary is present along every cycle  $C$  connecting them. We write this as:

$$\sum_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall C \in \text{Cycles} : \hat{e} \in C \tag{3}$$

where  $[ \ ]$  to denotes the indicator function whose value is 1 if the condition is true and otherwise outputs a zero. An equivalent definition is that for an edge in a cycle there must be at least one other edge in the cycle whose value is as large or larger.

$$\max_{e \in C - \hat{e}} X_e \geq X_{\hat{e}} \quad \forall C \in \text{Cycles} : \hat{e} \in C \tag{4}$$

We call the above inequalities ‘‘ultrametric inequalities’’. Each edge  $e$  is associated with a target value  $T_e \in [0, 1]$ . Finding the ultrametric  $X$  closest to  $T$  in an  $L^p$  sense ( $p$  is 1 or 2 depending on the desired norm) is the objective of ultrametric rounding. We write the optimization problem below.

$$\min_X \sum_e |X_e - T_e|^p \tag{5}$$

$$\text{s.t. } \max_{e \in C - \hat{e}} X_e \geq X_{\hat{e}} \quad \forall \{C \in \text{Cycles} : \hat{e} \in C\} \tag{6}$$

### 3.1 Correlation Clustering

When constructing our solver for minimizing ultrametric distortion we rely heavily on repeated calls to a solver for correlation clustering on a planar graph. Thus we now briefly discuss correlation clustering (Bansal et al., 2002; Kim et al., 2011; Yarkony et al., 2012; Bagon and Galun, 2011; Andres et al., 2012, 2013, 2011). Correlation clustering is a powerful clustering criteria in which each pair of nodes (in our case adjacent superpixels) is associated with a real valued term  $\theta_e$  where  $e$  indexes the edge between the two nodes. Correlation clustering groups the nodes into regions so as to minimize the sum of the  $\theta_e$  terms of edges spanning the boundary. We define the presence of a boundary using binary indicator vector  $Y$  which is indexed by  $e$ . Here  $Y_e = 1$  if and only if there is a boundary on edge  $e$ . Notice that if  $\theta_e > 0$  then it is desirable to set  $Y_e = 0$  and if  $\theta_e < 0$  it is desirable to set  $Y_e = 1$ . However  $Y$  has to be set so that a clustering is

produced. This means that no  $Y_e$  can be set to 1 in the middle of a region. These constraints are called cycle inequalities and they are the discrete binary analog of the ultrametric inequalities in Eq 3, 4. The form of cycle inequalities are written below.

$$\sum_{e \in C - \hat{e}} Y_e \geq Y_{\hat{e}} \quad \forall \{C \in \text{Cycles}, \hat{e} \in C\} \quad (7)$$

Correlation clustering is a natural clustering criteria because the number of regions is not a user defined hyper-parameter that must be hand tuned for each problem. Instead it is a function of the potentials  $\theta$  themselves. Notice that if  $\theta$  is exclusively positive then all superpixels are in the same region in the optimal solution. Also notice that if all  $\theta$  terms are negative then each superpixel is in its own region in the optimal solution.

Solving the correlation clustering problem is NP hard even for planar graphs (Bachrach et al., 2011). However for many problems in computer vision the PlanarCC algorithm (Yarkony et al., 2012) can solve them exactly usually in seconds or fractions of seconds. PlanarCC is a dual column generation algorithm operating only on planar graphs. PlanarCC provides upper and lower bounds on the optimal value of the objective. The upper bound is associated with a partition  $Y$  that achieves this value. In practice the upper and lower bounds are identical or nearly identical for problems in the domain of image segmentation (Yarkony et al., 2012) meaning that the solution is verified to be the global optima. PlanarCC provides fast performance for image segmentation problems in computer vision notably on the benchmark Berkeley Segmentation Data Set (BSDS)(Martin et al., 2001).

## 4 The Hierarchical Greedy Planar Correlation Clustering Algorithm (HGPPC)

We now consider the problem of minimizing ultrametric distortion as in Eq 5. We employ a coordinate descent approach in which at each step we identify optimal setting of  $X$  in a particular space that includes that current solution. We alternate between three unique coordinate descent steps which are described below. When we apply an update we denote the current setting of our solution as  $X^0$ , and the output as  $X^1$ . We initialize  $X^0$  to be the zero vector. At all times during our algorithm our solution describes an ultrametric. Two out of the three coordinate updates use the PlanarCC algorithm which requires planarity of the graph in order to work. To satisfy planarity in our application we have edges between each adjacent pair of superpixels and no other edges.

### 4.1 Update One: Shifting the Values in the Ultrametric While Preserving Their Order

Consider optimizing over  $X$  subject to the constraint that the ordering of  $X$  does not change. We frame this as an optimization problem which is potentially a linear or quadratic program depending on the norm applied on the ultrametric.

$$\begin{aligned} \min_X \sum_e |T_e - X_e|^p & \tag{8} \\ \text{s.t. } X_e \geq X_{\hat{e}} \quad \forall e, \hat{e} : X_e^0 \geq X_{\hat{e}}^0 & \end{aligned}$$

Let  $A_b$  be the set of edges that take on the  $b$ 'th smallest unique value specified by  $X^0$ . Our goal is to find new values  $\lambda_b$  to assign to each set of edges  $A_b$ . Let  $|\lambda|$  denote the the number of unique values in  $X^0$  and  $|A_b|$  the cardinality of  $A_b$ .

$$\begin{aligned} \min_\lambda \sum_b \sum_{e \in A_b} |T_e - \lambda_b|^p &= \min_\lambda \sum_b |A_b| |\lambda_b - T_e|^p & \tag{9} \\ \text{s.t. } \lambda_b \leq \lambda_{b+1} & \end{aligned}$$

In addition to solving the optimization above as a linear/quadratic program we can approach it as a dynamic program on a chain structured Markov random field. For each variable  $\lambda_b$  we create a node that has cost to take on each possible value  $\alpha_b$  of  $Z_b(\alpha_b)$  which is defined below.

$$Z_b(\alpha_b) = \sum_{e \in A_b} |T_e - \alpha_b|^p = |A_b| |\alpha_b - T_e|^p \tag{10}$$

We also have a pairwise potential over each pair of adjacent  $b$  values  $Z_{b,b+1}(\alpha_b, \alpha_{b+1})$  which is defined below.

$$Z_{b,b+1}(\alpha_b, \alpha_{b+1}) = \infty [\alpha_b > \alpha_{b+1}] \tag{11}$$

This pairwise potential simply enforces that the ordering of the values of  $b$  remains constant. We discretize the space of possible values for  $\lambda$  terms making sure to include all unique values in  $X^0$ . For example we can include 1000 uniformly distributed points between  $\min(T)$  and  $\max(T)$  in addition to all unique values of  $X^0$ . We denote the set of all such values as  $\Omega$ .

Computing the optimal  $\lambda$  in the above graphical model can be done using dynamic programming in time  $O(|\Omega||\lambda|)$ . Once we solve for  $\lambda$  we simply set each index of  $X$  to its associated value in  $\lambda$ . Thus  $X_e^1$  is set to  $\lambda_b \forall b, \forall e \in A_b$ .

### 4.2 Update Two: Raising the Values of $X$ to $\alpha$ in Large Groups

We now introduce a coordinate update that raises the values in  $X$  in large groups over long ranges of value while preserving the ultrametric property of  $X$ . This is a coordinate update parameterized by a randomly chosen value  $\alpha$  on the range of  $[\min(T), \max(T)]$ . Here  $\alpha$  is different every time this update is done.

During this update we optimize  $X$  over the space of ultrametrics subject to the constraint that  $X_e \in \{X_e^0, \max(X_e^0, \alpha)\}$  for all  $e$ . We denote this space as  $\hat{S}(X^0, \alpha)$  and the super-space that does not enforce the ultrametric property as  $S(X^0, \alpha)$ . We now write the objective of this update formally.

$$X^1 = \arg \min_{X \in \hat{S}(X^0, \alpha)} \sum_e |T_e - X_e|^p \tag{12}$$

Notice that in the space  $S(X^0, \alpha)$  the only possible violations to the ultrametric property come in the form of ultrametric inequalities of pairs of cycle  $C$ , and edge  $\hat{e}$  such that:  $\forall e \in C, X_e^0 < \alpha$  and  $X_{\hat{e}} = \alpha$ .

Using this we write a version of the ultrametric inequalities needed to ensure that any  $X \in S(X^0, \alpha)$  also lies in  $\hat{S}(X^0, \alpha)$ .

$$\max_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall \{C \in \text{Cycles}, \hat{e} \in C\} \tag{13}$$

Notice that we can replace the max in the above equation with a  $\sum$ . This is because each of the inequalities can be only violated if all terms under the sum/max are zero.

$$\sum_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall \{C \in \text{Cycles}, \hat{e} \in C\} \tag{14}$$

We write our coordinate update as an instance of correlation clustering. We use binary indicator  $Y_e$  as an indicator for  $[X_e \geq \alpha]$  and edge potentials  $\theta$  given by:

$$\theta_e = \begin{cases} |\alpha - T_e|^p - |X_e^0 - T_e|^p & \forall e \text{ s.t. } X_e^0 < \alpha \\ -\infty & o.w. \end{cases}$$

where edges with potential  $-\infty$  are required to be active in the final solution.

The resulting correlation clustering problem is then

$$\min_Y \sum_e \theta_e Y_e \tag{15}$$

$$\text{s.t. } \sum_{e \in C - \hat{e}} Y_e \geq Y_{\hat{e}} \quad \forall (C \in \text{Cycles}, \hat{e} \in C) \tag{16}$$

After computing  $Y$  we simply set  $X_e^1 \leftarrow \alpha$  iff  $(Y_e = 1 \text{ and } X_e^0 < \alpha)$ ; otherwise set  $X_e^1 \leftarrow X_e^0$ .

**Implementation Detail.** Since we already established that no edge  $e$  s.t.  $X_e^0 \geq \alpha$  is involved in any necessary ultrametric inequality in  $S(X^0, \alpha)$  and their  $\theta$  terms are negative then we can simply remove (ignore) those edges from the graph and set their values in  $X^1$  to  $X^0$ . This saves us from having  $-\infty$  as the value of an edge potential.

Another way of ignoring edges such that  $X_e^0 \geq \alpha$  is as follows. For each such edge set  $\theta_e = 0$ . Next then solve for  $Y$ . Finally set  $X_e^1 \leftarrow X_e^0$  for all such edges. We use this approach as it avoids instantiating multiple graph structures.

### 4.3 Update Three: Lowering the Values of $X$ for a Subset of $X$

We now discuss a coordinate update that lowers the values in  $X$  for all values that take on a unique value in  $X$  so as to reduce the ultrametric distortion of  $X$ . This update parameterized by a randomly chosen value  $\alpha$  on the range of  $[\min(T), \max(X)]$ . Here  $\alpha$  is different every time we perform this update. Let

the set of all unique values in  $X^0$  be denoted  $\lambda^0$ . Here  $\lambda^0$  is sorted with  $\lambda_0^0$  being the smallest and  $\lambda_{|\lambda^0|}^0$  being the greatest. Let  $\mu$  be the smallest value in  $\lambda^0$  greater than  $\alpha$ .

We optimize over the space of solutions in which each  $X_e$  such that  $X_e^0 = \mu$  may take on either  $\alpha$  or  $\mu$  and all  $X_e$  such that  $X_e^0 \neq \mu$  must continue to take on their current value. We denote the space of solutions that meet these properties as  $V(X^0, \alpha)$  and the subset of that space corresponding to ultrametrics as  $\hat{V}(X^0, \alpha)$ . We now formally write the optimization over the space  $\hat{V}(X^0, \alpha)$ .

$$X^1 = \arg \min_{X \in \hat{V}(X^0, \alpha)} \sum_e |T_e - X_e|^p \tag{17}$$

The ultrametric inequalities needed to enforce that an  $X \in V(X^0, \alpha)$  is also in  $\hat{V}(X^0, \alpha)$  are written below.

$$\max_{e \in C - \hat{e}} [X_e > \alpha] \geq [X_{\hat{e}} > \alpha] \quad \forall \{C \in \text{Cycles } \hat{e} \in C\} \tag{18}$$

As in the previous subsection (see the transition from Eq 13, to Eq 14) we can replace the max with a  $\sum$  allowing us to write our coordinate update as an instance of correlation clustering with  $Y_e$  as an indicator for  $[X_e > \alpha]$ . The correlation clustering objective is described by the potentials

$$\theta_e = \begin{cases} -|\alpha - T_e|^p + |X_e^0 - T_e|^p. & \forall e \text{ s.t. } X_e^0 = \mu \\ -\infty & \forall e \text{ s.t. } X_e^0 > \mu \\ \infty & \forall e \text{ s.t. } X_e^0 \leq \alpha \end{cases}$$

where the edges with negative and positive infinite weights are required to be cut or not cut respectively.

After solving the optimization above we simply set  $X_e^1 \leftarrow \alpha$  iff ( $Y_e = 0$  and  $X_e^0 = \mu$ ); otherwise set  $X_e^1 \leftarrow X_e^0$ . Note that this operation can be performed in parallel with a unique value  $\alpha$  chosen between each pair of adjacent  $\mu$ . As in the previous section we can ignore the edges that must be boundaries in the solution meaning ( $X_e^0 > \alpha$ ) as they are not involved in any violated cycle inequalities and furthermore must be set to 1. Ignoring them is done by setting their  $\theta$  value to zero. Similarly we can merge any superpixels that are connected by an  $\infty$  valued potential. Merging superpixels was not done in our experiments but can conceivably make inference faster.

#### 4.4 Final Procedure

Updates can be performed in any order. Furthermore one can complete multiple updates of one type in a row. For our experiments we consider one iteration to be completing updates 1,2,1,3. We repeat this iteration many times in our experiments.

## 4.5 Optimality in PlanarCC

For our experiments we used the PlanarCC code provided by the authors of (Yarkony et al., 2012). We operated this code unchanged. PlanarCC attacks an NP hard problem so it is conceivable that its lower and upper bounds are not tight at convergence or when a user would want an anytime solution. Thus when we terminate PlanarCC which we run for no more than a minute we take the best anytime solution generated (including the solution corresponding to the initial solution). We never saw this time limit reached.

## 5 Experiments

In order to evaluate the generality and robustness of our approach, we test it on datasets that differ in sample preparation and imaging equipment and conditions.

Data set one: These are bright field Diploid yeast cell images from (Zhang et al., 2014a), in which both out-of-focus and in-focus cells exist and are cluttered together. And the cells of interest are only the in-focused ones, i.e. those with least contrast on cell boundaries. Apart from this, cell boundaries can be partially missing and with diverse appearances, even in the same cell.

Data set two: These are phase-contrast HeLa cell images from (Arteta et al., 2012). It presents a high variability in cell shapes and sizes, as opposed to the ellipse like cells in data set one. These images have relatively lower resolution, where cell boundaries are disturbed by the bright halo owing to this specific imaging technique.

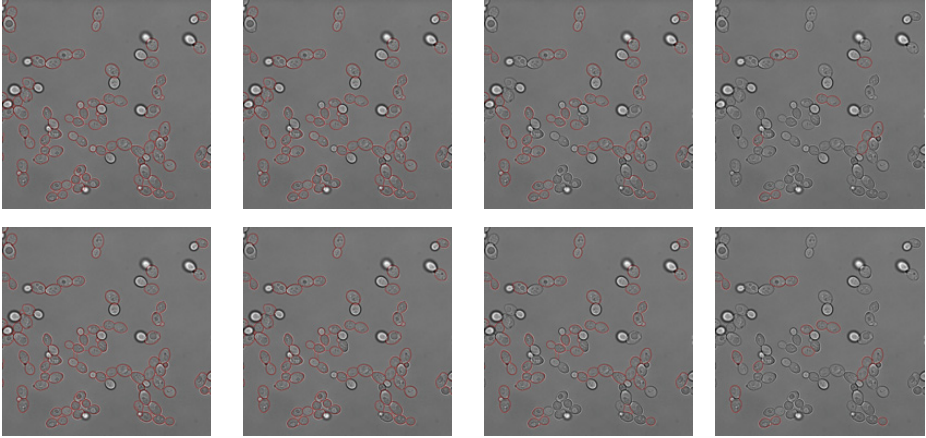
### 5.1 Producing Problem Instances

The edge probability map is predicted from a trained classifier using ilastik (Sommer et al., 2011), an open-source toolkit that relies on a family of generic nonlinear image features and random forests, to estimate the probability of belonging to a cell boundary edge for each individual pixel. We use a small labeled training data set.

To compute superpixels we use a watershed transformation then smooth the result using a gaussian filter. Finally we compute the average boundary probability along each superpixel boundary thus providing a value  $Pb_e$  for every edge  $e$ . UCM operates on this raw probability. We take the log odds ratio to convert that to an energy which is then used as the targets for HGPCC. The equation for the targets is written below.

$$T_e = -\log\left(\frac{1 - Pb_e}{Pb_e}\right) \quad (19)$$

For HGPCC we experimented with  $L1$  and  $L2$  norms in the log odds ratio space. In Fig 1 we display the results of UCM and of HGPCC for an image in



**Fig. 1.** Top Row: UCM segmentation thresholding  $X$  at values various thresholds. Bottom Row: HGPCC segmentation thresholding  $X$  at values at the same thresholds as UCM. We indicate boundaries in red.

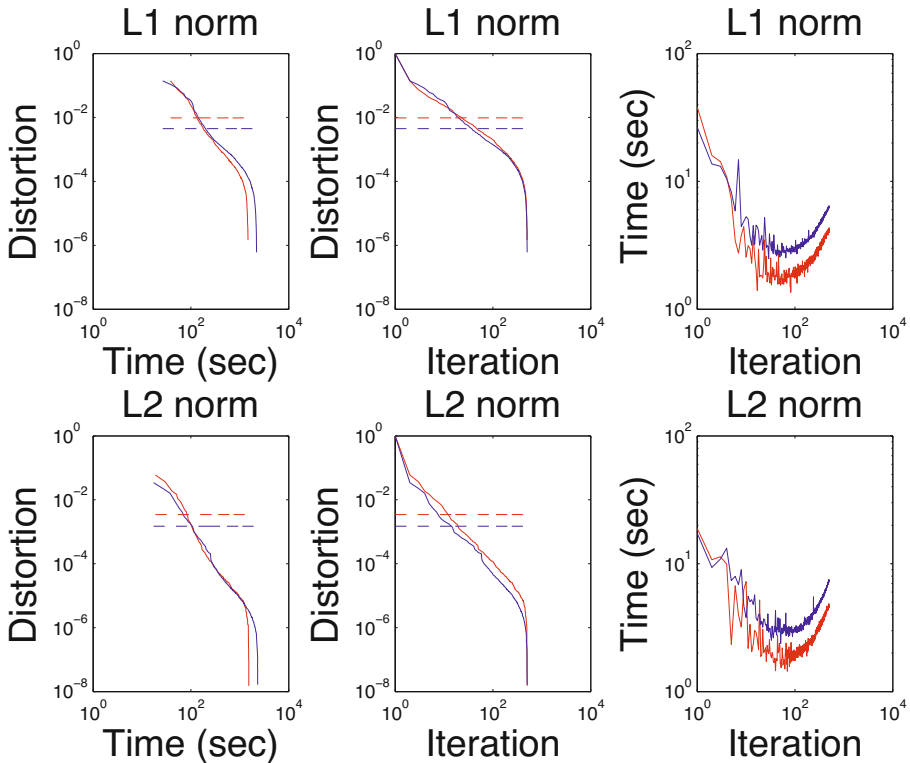
the data set one. Once the nearest ultrametric to  $T$  is solved for in log odds space we convert  $X$  to a probability by a sigmoid operation.

With regards to the quality of the segmentations we found no significant qualitative difference between UCM and HGPCC. That being said HGPCC has multiple advantages over UCM. First it is an energy minimization formulation which allows for structured learning and principled mathematical extensions to be used. Second HGPCC is robust to indications of no boundaries being placed on actual boundaries, which may result in the merging of these boundaries at finer positions in the hierarchy for UCM than desirable.

### 5.2 Experimental Comparisons: Distortion and Timing

For problems in data set one and data set two we completed 500 iterations of HGPCC. For each iteration we completed updates 1,2,1,3 in that order. We found that HGPCC converged very rapidly. Furthermore the time to complete an iteration of HGPCC decreases at first then after convergence begins to increase again. We compared against the UCM algorithm which since it is not an iterative algorithm was not timed. It is very fast compared to our approach. We found that HGPCC produces lower distortion ultrametrics than UCM very early during optimization.

When plotting the distortion we applied the following normalization scheme. All distortions including the output of UCM are normalized by subtracting off the lowest value of HGPCC for a given instance and dividing by the gap between the lowest and highest distortions of HGPCC for a given instance. All results are averaged across the data sets. All results are plotted in Fig 2.



**Fig. 2.** We show the convergence of HGPPC as a function of time and iteration and compare it to the final result of UCM (which is not timed). We use the data set one and data set two and color their results red and blue respectively. Dotted lines correspond to UCM and solid lines to HGPPC. Left Column) Distortion as a function of time. Center Column) Distortion as a function of iteration. Right Column) Time for an iteration of HGPPC as a function of iteration.

## 6 Conclusion

We present a novel fast algorithm for finding low distortion ultrametrics on planar graphs. Our method exploits the fact that correlation clustering can often be done efficiently on planar graphs with very high degrees of accuracy. Our method is an analog of alpha expansion/alpha beta swap (Boykov et al., 2001) as both make large efficient moves in the space of values for their variables. This work extends the family of PlanarCC (Yarkony et al., 2012; Andres et al., 2013; Zhang et al., 2014b) methods so as to include efficient hierarchical clustering.

**Acknowledgement.** We thank F. Huber and M. Knop from ZMBH University of Heidelberg, Germany for sharing the bright field images.



## References

- Tan, M., Deklerck, R., Jansen, B., Bister, M., Cornelis, J.: A novel computer-aided lung nodule detection system for CT images. *Medical Physics* 38, 5630–5645 (2011)
- Ailon, N., Charikar, M.: Fitting tree metrics: Hierarchical clustering and phylogeny. In: *Proceedings of the Symposium on Foundations of Computer Science*, pp. 73–82 (2005)
- Yarkony, J.: *MAP Inference in Planar Markov Random Fields with Applications to Computer Vision*. PhD thesis, University of California Irvine (2012)
- Ren, X., Malik, J.: Learning a classification model for segmentation. In: *Ninth IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, vol. 1, pp. 10–17 (October 2003)
- Liu, F., Xing, F., Yang, L.: Robust muscle cell segmentation using region selection with dynamic programming. In: *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pp. 1381–1384 (2014)
- Wu, Z., Gurari, D., Wong, J.Y., Betke, M.: Hierarchical Partial Matching and Segmentation of Interacting Cells. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) *MICCAI 2012, Part I. LNCS*, vol. 7510, pp. 389–396. Springer, Heidelberg (2012)
- Su, H., Yin, Z., Kanade, T., Hun, S.: Interactive cell segmentation based on correction propagation. In: *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pp. 1267–1270 (2014)
- Su, H., Yin, Z., Hun, S., Kanade, T.: Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical Image Analysis* 17, 746–765 (2013)
- Zhang, C., Huber, F., Knop, M., Hamprecht, F.A.: Yeast Cell Detection and Segmentation in Bright Field Microscopy. In: *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pp. 1267–1270 (2014a)
- Bansal, N., Blum, A., Chawla, S.: Correlation clustering. *Journal of Machine Learning*, 238–247 (2002)
- Kim, S., Nowozin, S., Kohli, P., Yoo, C.D.: Higher-order correlation clustering for image segmentation. *Advances in Neural Information Processing Systems* 25, 1530–1538 (2011)
- Yarkony, J., Ihler, A., Fowlkes, C.C.: Fast Planar Correlation Clustering for Image Segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part VI. LNCS*, vol. 7577, pp. 568–581. Springer, Heidelberg (2012)
- Bagon, S., Galun, M.: Large scale correlation clustering 816 optimization. CoRR, abs/1112.2903 (2011)
- Andres, B., Kroeger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Koethe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part III. LNCS*, vol. 7574, pp. 778–791. Springer, Heidelberg (2012)
- Andres, B., Yarkony, J., Manjunath, B.S., Kirchhoff, S., Turetken, E., Fowlkes, C.C., Pfister, H.: Segmenting planar superpixel adjacency graphs w.r.t. Non-planar superpixel affinity graphs. In: Heyden, A., Kahl, F., Olsson, C., Oskarsson, M., Tai, X.-C. (eds.) *EMMCVPR 2013. LNCS*, vol. 8081, pp. 266–279. Springer, Heidelberg (2013)
- Andres, B., Kappes, J.H., Beier, T., Kothe, U., Hamprecht, F.A.: Probabilistic image segmentation with closedness constraints. In: *Proceedings of the Fifth International Conference on Computer Vision (ICCV 2011)*, pp. 2611–2618 (2011)

- Bachrach, Y., Kohli, P., Kolmogorov, V., Zadimoghaddam, M.: Optimal coalition structures in graph games. CoRR, abs/1108.5248 (2011)
- 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 the Eighth International Conference on Computer Vision (ICCV-2001), pp. 416–423 (2001)
- Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Learning to Detect Cells Using Non-overlapping Extremal Regions. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 348–356. Springer, Heidelberg (2012)
- Sommer, C., Straehle, C., Kothe, U., Hamprecht, F.A.: ilastik: Interactive learning and segmentation toolkit. In: Eighth IEEE International Symposium on Biomedical Imaging, ISBI 2011 (2011)
- Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001)
- Zhang, C., Yarkony, J., Hamprecht, F.A.: Cell Detection and Segmentation Using Correlation Clustering. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014, Part I. LNCS, vol. 8673, pp. 9–16. Springer, Heidelberg (2014)

# Author Index

- Ackermann, Hanno 450  
Åström, Freddie 307
- Bae, Egil 15, 278, 478  
Bansal, Sumukh 223  
Baravdish, George 307  
Baxter, John S.H. 278  
Bekkers, Erik 464  
Bergmann, Ronny 155  
Bertozzi, Andrea L. 209  
Bodnariuc, Ecaterina 378  
Bui, Alex 478
- Chang, Huibin 335  
Chin, Tat-Jun 450  
Cremers, Daniel 126, 141, 183
- Diethelm, Remo 112  
Doğan, Günay 292  
Dong, Xingping 237  
Dragomiretskiy, Konstantin 197  
Duggan, Nóirín 478  
Duits, Remco 464  
Duran, Joan 141
- Elad, Michael 99
- Favaro, Paolo 112, 350  
Felsberg, Michael 307  
Fenster, Aaron 278  
Fisher III, John W. 43, 71  
Fowlkes, Charless C. 492
- Glavin, Martin 478  
Gurung, Arati 378
- Hoeltgen, Laurent 85  
Hoffmann, Sebastian 169  
Hosseini Kamal, Mahdad 350  
Hsu, William 478  
Hu, Huiyi 209
- Jeong, Seong-Gyun 436  
Jones, Edward 478
- Kennedy, Ryan 364  
Kohli, Pushmeet 406  
Krivobokova, Tatyana 263
- Larsson, Viktor 1  
Li, Housen 263  
Li, Kunqian 249  
Li, Peiran 392  
Li, Zhi 321  
Liao, Xiangli 249  
Liu, Liman 249, 392  
Loog, Marco 464
- Mobahi, Hossein 43, 71  
Moeller, Michael 126, 141  
Möllenhoff, Thomas 126  
Munk, Axel 263
- Olsson, Carl 1  
Osokin, Anton 406
- Pavlovskaja, Maira 421  
Perrone, Daniele 112  
Peter, Pascal 263  
Peters, Terry M. 278  
Petra, Stefania 378  
Plonka, Gerlind 169  
Průša, Daniel 57
- Rajchl, Martin 278  
Rosenhahn, Bodo 450
- Sbert, Catalina 141  
Scheuermann, Björn 450  
Schnörr, Christoph 378  
Shapovalov, Roman 406  
Shen, Jianbing 237  
Shen, Shiwen 478  
Strekalovskiy, Evgeny 126  
Stühmer, Jan 183  
Sulam, Jeremias 99  
Sun, Kun 392  
Sunu, Justin 209
- Tai, Xue-Cheng 15, 278, 335  
Tao, Wenbing 249, 392  
Tarabalka, Yuliya 436  
Tatu, Aditya 223

- Taylor, Camillo J. 364  
Tu, Kewei 421  
Vandergheynst, Pierre 350  
Van Gool, Luc 237  
Vese, Luminita 478  
Vetrov, Dmitry 406  
Wang, Junyan 29  
Weickert, Joachim 85, 169, 263  
Weinmann, Andreas 155  
Werner, Tomáš 57  
Yang, Danping 335  
Yarkony, Julian 492  
Yeung, Sai-Kit 29  
Yuan, Jing 15, 278  
Zeng, Tiejong 321  
Zerubia, Josiane 436  
Zhang, Chong 492  
Zhu, Song-Chun 421  
Zosso, Dominique 197